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

International Journal for Electronic Crime Investigation

(IJECI)

VOLUME: 7
ISSUE: 4 Oct-Dec 2023

Email ID: ijeci@lgu.edu.pk

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

International Journal for Electronic Crime Investigation

Volume 7(4) Oct-Dec 2023

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 7(4) Oct-Dec 2023

CONTENTS

International Journal for Electronic Crime Investigation
Volume 7(4) Oct-Dec 2023

Editorial

Modernizing Crime Detection: The Imperative of Technological Integration in Pakistani Law Enforcement

Kaukab Jamal Zuberi

1. Introduction

In an era marked by rapid technological advancements, the landscape of crime and its detection is undergoing a profound transformation. Nowhere is this more evident than in Pakistan, where the exponential growth of digital data has ushered in a new era for law enforcement agencies. The imperative for innovation in investigation is unmistakable, particularly in the seamless integration of information technology into crime detection processes. This editorial delves into the significance of leveraging advanced technologies for collecting, analyzing, and preserving digital evidence in Pakistan. It underscores the need for law enforcement agencies to adapt to these changes, emphasizing the pivotal role of universities in redefining criminology education to produce a new generation of technology-aware investigators.

2. The Digital Frontier: A Paradigm Shift in Crime Detection

The advent of the digital age has led to an unprecedented increase in the volume and complexity of data generated daily. Criminal activities, too, have found new avenues in the digital realm, necessitating a corresponding evolution in investigative techniques. Traditional methods, while still relevant, are often insufficient in the face of cybercrimes, digital fraud, and other technologically mediated offenses. Embracing information technology in crime detection is not just a matter of staying abreast of current trends; it is a fundamental shift in approach that can significantly enhance the effectiveness and efficiency of investigations.

3. The Role of Digital Forensics in Modern Investigations

Digital forensics has emerged as a critical component of modern investigative processes. The ability to collect, analyze, and preserve digital evidence is paramount in solving cybercrimes and uncovering hidden connections between criminal elements. Law enforcement agencies in Pakistan must invest in training and equipping their personnel with the necessary skills to navigate the complexities of digital forensics. This includes staying updated on the latest tools and methodologies to ensure the successful extraction and interpretation of digital evidence.

4. Integration of Traditional and Digital Investigative Methods

While the integration of information technology is crucial, it is equally important to recognize the continued relevance of traditional investigative methods. Combining the strengths of both traditional and digital approaches creates a comprehensive investigative framework. For instance, witness interviews, physical evidence collection, and crime scene analysis remain integral to investigations. By seamlessly integrating these methods with digital forensics, law enforcement can harness a more holistic and effective approach to solving crimes.

5. Empowering Law Enforcement: Justice and a Safer Society

The primary objective of modernizing crime

detection through technology is to empower law enforcement agencies. The seamless integration of traditional and digital investigative methods enhances their capabilities, enabling quicker response times, more accurate identification of perpetrators, and a proactive approach to preventing crimes. Ultimately, this empowerment contributes to upholding justice and fostering a safer society for all citizens.

6. The Crucial Role of Universities in Shaping the Future of Investigation

To realize the full potential of technological integration in crime detection, a paradigm shift is required in the education and training of future investigators. Universities, particularly those offering criminology degrees, play a pivotal role in shaping the mindset and skills of aspiring law enforcement professionals. A crucial step in this direction involves redesigning criminology programs to incorporate a strong focus on technology.

7. Redesigning Criminology Degrees: A Technology-Centric Approach

Traditional criminology degrees often lack a comprehensive understanding of information technology's role in contemporary crime. Redesigning these programs should involve the introduction of specialized courses in digital forensics, cybersecurity, and data analytics. Students should be equipped with the knowledge and skills needed to navigate the evolving landscape of cybercrimes and other technology-driven offenses. Practical, hands-on training with state-of-the-art forensic tools should be an integral part of these programs.

8. Creating a New Generation of Technology-Aware Investigators

The redefined criminology degrees should aim to produce a new generation of investigators who are not only well-versed in traditional investigative methods but also proficient in leveraging technology. This involves fostering a mindset that values the integration of technology into every aspect of crime detection. Graduates should be equipped to bridge the gap between the analog and digital worlds, ensuring a seamless transition between traditional and digital investigative approaches.

9. Industry-Academia Collaboration: Bridging the Gap

To ensure the relevance and effectiveness of these redesigned criminology programs, collaboration between universities and law enforcement agencies is paramount. Establishing partnerships with industry experts and incorporating real-world case studies into the curriculum can provide students with practical insights and a deeper understanding of the challenges faced by law enforcement professionals. This collaboration can also facilitate internships and practical training opportunities, ensuring that graduates are well-prepared for the demands of the modern investigative landscape.

Conclusion: A Technologically Empowered Future

In conclusion, the inevitability of innovation in investigation, particularly in the realm of information technology, is a reality that law enforcement agencies in Pakistan must confront. The exponential growth of digital data demands a corresponding evolution in investigative techniques. The integration of traditional and digital methods is not just a luxury but a necessity for ensuring the effectiveness and efficiency of crime detection. Universities must rise to the occasion by redesigning criminology degrees to produce a new generation of technology-aware investigators. In doing so, they contribute not only to the empowerment of law enforcement but also to the overarching goal of upholding justicand ensuring a safer society for everyone. The future of crime detection in Pakistan lies at the intersection of tradition and technology, and it is imperative that both law enforcement and academia embrace this paradigm shift wholeheartedly.

Zohaib *et al. (IJECI)* 2023

International Journal for
Electronic Crime Investigation

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

DOI: https://doi.org/10.54692/ijeci.2023.0704163

Research Article

Vol. 7 issue 4 Oct-Dec 2023

A Comparative Analysis of Malware Detection Methods Traditional vs. Machine Learning

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

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

[2] School of Computer Science, National University of Ireland, Maynooth

[3]Faculty of Information Technology, Beijing University of Technology, Beijing, China

Corresponding author: ahmedzohaib03@gmail.com

Received: September 20, 2023; **Accepted:** November 08,2023; **Published:** December 22, 2023

ABSTRACT

Mobile devices have been the target of malicious software since their beginnings. Two known types of malware threats can intrude into mobile, independently injected applications and fraudulent applications that are developed to breach the security of mobile devices. Mostly these fraudulent applications access data using API calls and permission requests. API calls and permission requests are important for smooth conversation between mobile devices and database servers. This research proposes an efficient classification model that concatenates API calls and permission requests to detect malicious applications. We have used a dataset that contained more than 15 thousand Android devices' malware. We have divided data into three groups to differentiate between the malicious permission requests and malicious API calls with normal permission requests and normal API calls. To increase the probability of recognizing Android malware applications, we develop three distinct grouping strategies for selecting the most valuable API calls that are obscure, critical, and obstreperous and are chosen because Android apps extensively use several application programming interfaces (APIs). According to the results, malware applications require authorizations to access confidential information very frequently than normal Android applications do. Also, malicious Android applications raise a diverse set of API calls to access sensitive data, evidenced by malware applications making a distinct set of API calls. Our proposed method attains an F-score of 94.04%, which suggests that it is efficient at discovering mobile malware applications. Our model can be of significant assistance in conducting mobile application analysis and forensic investigations into malware.

Keywords: Permission, Genetic Algorithm, Mobile, Hybrid analysis, Dynamic analysis, Deep learning, intent, API calls

1. INTRODUCTION

There are many ways to access the internet today, and the most common method is via mobile devices. The Internet's explosive growth, combined with recent increases in automation via intelligent applications, creates a favorable environment for attackers using

malicious software. Malware threats have increased due to the global adoption of cloud computing and Internet of Things (IoT) [1]. Such malicious actions have the potential to compromise the integrity, confidentiality, or availability of mobile systems [2]. The people who make mobile malware have taken PC malware and added new features to it to make new threats to mobile platforms. Putting in place forensic identification security controls will make it less likely that digital systems will be broken into [3]. Mobile malware are virulent software that explicitly developed and considered to attack mobile devices, e.g., Smartphones and other devices [4]. Mobile malware refers to any form of harmful code or software that compromises the safety and performance of a mobile device without the knowledge or permission of the device's owner. Ransomware, Trojan horses, worms, spyware, rootkits, and botnets are all examples of different types of malware. Mobile malware is becoming more sophisticated and dangerous because it collects user data, sends premium text messages, makes calls, etc. Mobile platforms are now the primary target for cybercriminals that create malicious software, resulting in a 1,800% spike in mobile malware in 2016. Check Point did an international survey of 850 businesses and found that all of them had been attacked by mobile malware. According to Kaspersky, the number of users who have been infected with Android malware has more than tripled to 1.7 million worldwide in 2019. As can be seen in Fig. 1, the number of mobile malicious installation packages that Kaspersky found in 2021 was 3,464,756, which decreased of 2,218,938 from the previous year. The total number of mobile malware installation packages has decreased to levels roughly equivalent to those seen in 2019. The number of attacks continued to go down steadily throughout the reporting period, and in the second half of 2021 they reached their lowest monthly average in the previous two years can be seen in Fig. 2.

In the last quarter of 2017, McAfee Labs discovered 16M mobile malware [5], and Juniper reported a 400\% increase in Android malware. Over 1.05 million Android malware apps have been detected by Sophos Labs since 2010 [6]. Smartphone malware is always busy updating new features, e.g. always looking for new ways to shift into new distribution, methods and avoid detection techniques, such as obfuscation technique stealth methods, and repacking methods [7]. An old Study [8] shows that most of the Android malware used a repackaged technique they merged codes in other legitimate known applications to avoid security checkpoints.

Google Play Store is used by malware developer to download popular Android applications. Then decompile the applications, insert malicious code into the apps, and then reupload the applications with the malignant content to third-party markets for user adoption [9]. Existing mobile anti-malware applications were found to be unable to detect malware apps that have been obfuscated or repackaged.

Using ten malicious apps from six different families, researchers were able to test the effectiveness of mobile anti-malware scanners using a variety of obfuscation tactics [10]. It was then tested versus 10 reputable anti-malware scanners using these new obfuscated binaries. As per the results, there is not a single antimalware that was able to detect any malicious applications. Because there are so

many mobile apps out there, it's vital to assess and check what's available in marketplaces swiftly and intelligently [11]. An automated system that can identify and remove harmful programs from both official and unofficial markets must be established to prevent them from being downloaded. As part of [12] malicious content was introduced in Android apps resources to assess if ten anti-malware scanners can detect it. While the remainder of the anti-malware scanners were unable to discover any dangerous information in their results, just one anti-malware scanner detected two hiding tactics [13].

2. RELATED WORKS

In the last ten years, substantial advancement has been made in the study and discovery of malware in mobile devices. This section in total investigates challenges associated with the detection of mobile malware and investigates expressively related methods that have been suggested by a body of previous research. The issues surrounding mobile malware are discussed in this section identification and explores the literature's substantially related approaches. Three techniques used by security companies and researchers for extracting features from mobile applications are static, dynamic and hybrid

2.1. Static analysis

Static analysis is the name given to the study of computer programs that do not include the actual running of the program's code. Static analysis techniques and processes include those that employ analytic approaches to examine computer programs. According to the most recent findings of the research that has been conducted, it was found that there exists a variety of attributes that are utilized for the purpose of static analysis of applications in order to identify malicious applications. The primary purpose of permissions is to store all the information on the permissions necessary to execute an app in the system/mobile. Therefore, developers can investigate the behavior or intent of an application based on the permissions requested. The detecting mechanism makes use of the regularity of approvals utilized by malevolent and benign programs. It may be used alone or in conjunction with one or more additional features. Any application's .apk file's manifest file is where permissions are extracted [14]. It has contrasted the usage of regular and malicious applications using both unique requested permissions (URP) and unique utilized permissions (UUP). API calls are the second-most utilized functionality. The application's class.dex file may be used to extract these API calls. The application's API calls may be examined during the investigation process. Skeptical, and potentially harmful API calls are recognized, and apps are categorized based on this information. Droidlogger [15] employed API call blocks, which is a collection of APIs used for a certain purpose and operation, and outperformed that single API call analysis. Permissions and API are often used in combination with one another [16]. The core components of the program, like the manifest file and class.dex file, are covered by both of these. Metadata information, string searches, call graphs, hardware elements, and other aspects are made use of also, nonetheless less often than permissions and APIs.

2.2. Dynamic analysis

The term "dynamic analysis" makes mention of a set of methods and procedures that are used when analytical techniques are applied to the

research of any software in which program implementation is involved with monitoring and parallel outcomes. In dynamic analysis, the two primary approaches are in-box analysis and out-of-box analysis. To utilize such strategies, an isolated atmosphere is required in order to execute the program and see the data in real-time [17]. Vibrant analysis is more difficult than static examination, and it requires a variety of tools and expertise to observe and draw conclusions. In [18] employs system call patterns for any inquiry process. A distinctive strategy that researchers have conducted is a filtering mechanism that calls for abstraction and improved results; moreover, it substitutes mechanism calls with aliases, and created a method to detect malicious program activity based on how often system calls occur. The entire dissimilarity between weighted system calls (ADWSC) and ranked system calls utilizing a large population test are the two methods used to assess system calls (RSLPT) [19]. The analysis of numerous parameters, including average packet size, total count of packets transmitted and acquired, duration in-between packets, ratio of incoming and outgoing packets, etc., is done in order to identify malicious activity in network traffic [20]. In [21], the authors examined the HTTP flow request by making use of natural language processing for string analysis while treating the HTPP flow as a document. Other attributes Hardware resources including CPU, memory, battery, and other hardware resources are also employed as features for behavior analysis, however, they are less effective than system call and network traffic. In general, authorizations as well as APIs are utilized for static analysis, however, sometimes, they are also examined during execution. In article [22] and [23], the researchers performed a taint analysis, which is a sort of behavioral analysis that makes use of the source and sink paradigms for data flow. Finally, it has been seen that most researchers prefer network traffic and system calls as features in their dynamic analysis.

2.3. Hybrid analysis

Static and dynamic analysis are combined in the ensemble analysis. Using AAPT (Android asset packaging tool), researchers [24] used static analysis in order to acquire permission requests from the manifest file. In addition to that, they made advantage of dynamic analysis when tracing system calls with the Strace tool. One hundred and eighty-eight benign applications and one hundred and eleven malicious ones were gathered by the researchers. Static and dynamic analysis aspects were merged by the authors. For the aim of evaluating their technology, they used four different ML algorithms, with the best detection accuracy being 70.31 percent. Static analysis using the APK tool yielded 135 permissions. The top 87 permissions were retrieved by utilizing IG as a search engine. The system calls were dynamically recorded by combining an Android emulator with the Strace program. System calls were examined to see if a certain call was invoked more frequently in malignant code than in benign code. They used six different ML algorithms to see how well their approach worked. Static analysis accuracy was 0.972, and dynamic analysis accuracy was 0.884, thanks to the random forest they used. The researcher concludes that a permission-based static feature is substantially more informative than a system-based dynamic feature [25]. As a result, in this work, we decide to employ static analysis in conjunction with permissions-based features to study API calls.

3. MACHINE LEARNING

Some researchers have tried applying deep learning and ML based on API call relationships to find behavioral patterns in benign and malicious applications to develop a detection system. These efforts failed. It was reported that the authors of the paper [26] had obtained an accuracy of 96 percent on Drebin (beneficial 5.09K) dataset and AMD (benign 20.05K and malware 20.08K). UniPDroid, developed by [27], combines static analysis as well as ML methods to classify malicious software families. Throughout all, they found 15,884 harmful programs in their research. They gathered 560 features through static analysis. Meta-transformer and extra-trees classifiers were used by the authors to narrow the list of candidates. They tested their technique on 78 different malware families, using the XGBoost classifier, and got an average classification accuracy of 92%. MalDozer, a program developed by [28], studied the efficiency of API call raw sequences and deep learning algorithms in detecting malicious software. A total of 33,000 malicious programs from Drebin, MalDozer, and Malgenome as well as 38000 benign apps from Google Play Store were analyzed for API method call sequences using Dexdump by the researchers. MalDozer uses two-word embedding algorithms, GloVe and word2vec, to normalize the feature vector. The detection accuracy was between 96% and 99.6%, with a false positive rate between 0.06% and 2%. Permission requests were not considered when writing the article. We looked at permissions and the frequency with which Android platform APIs, such as classes, packages, methods, and constructors were utilized in our research. The authors conducted an extensive research on Android malware recognition using a deep learning approach. The authors used a mobile security framework to extract permissions, intent filters, incorrect certificates, and API calls from the asset folder that contained APK files (MobSF). After that, all five features were transformed into vector space. They used a neural network to test their technique on both benign and malicious software to see how well it worked. They used 80\% for training and 20% for testing, and their detection accuracy was 96.81 percent.

4. DATASETS

We use two different kinds of datasets for the evaluation experiments a benign dataset and a malicious dataset. A benign dataset contains an application that is well-intentioned, and harmless while a malware dataset contains an application that is malicious and harmful, as indicated in Fig. 5. We do the assessment tests under both data sets to see which one performs better. We use reference datasets like Contagio, VirusShare, MalShare, AndroZoo, and VirusTotal for the malware dataset. There are 5,560 malware programs in this dataset, and at least ten anti-malware products have scanned and identified them such as VirusTotal. We were unable to locate a standard benign dataset, so we decided to develop our own and run it through VirusTotal to ensure that it was complete and accurate. AndroZoo was used to gather useful apps from the PlayStore. There are 9,476 benign applications in this dataset. Since the application's data gathered in June 2021 were tested by utilizing VirusTotal, the app categories reflect this. In case all anti-malware vendors in the database found a program to be safe, we consider it to be safe as well. To remedy the issue of class imbalance, we utilized SMOTE (synthetic minority over-sampling technique).

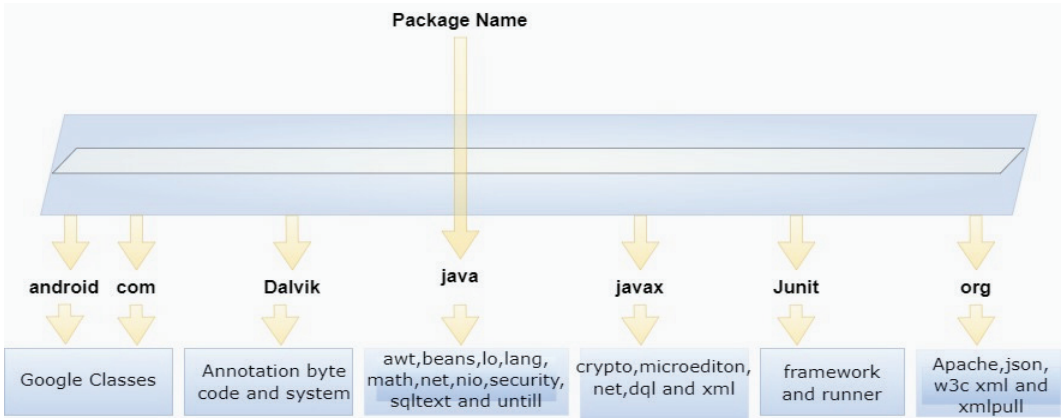


Fig 1: Architecture for ranking the API calls

Table 1: Dataset Description

Mobile Apps	Number
Malware	5,560
Benign	9,470
Total	15,036

5. METHODOLOGY

It is possible to extract and correlate the behavior of permission requests and API calls displayed by malicious programs in order to improve the preciseness of mobile malware detection. To increase possibility of discovering malicious apps, we make use of permissions analysis and analysis of the frequency of API calls. Applying the framework that we have suggested, it is possible for us to determine which API calls made by malicious Android software are the most important by using a scoring and categorizing method. Finding repackaging apps by comparing their names, hash values, or entry in a blacklist database is a fruitless endeavor. Instead, we propose a system that compares the regularity of the API calls and permissions across two programs in order to detect comparable repackaged applications. We created a three-stage analysis model for our suggested mobile malware analysis method for research purposes. Pre-processing stage, Extraction stage, and Grouping stage.

5.1. Pre-processing stage

The Java programming language is used to write Android apps, which are then converted into Java bytecode and then converted to Dalvik executable bytecode using the Dalvik virtual machine. Many files with the .class extension are created when the Java code is built. The Java source code, when compiled, results in the creation of many files along with the extension .class. The dx tool is used to combine all of the separate class files into one.dex file. An Android app's binary data is stored in the APK file. It's critical to decompile the Android app first before doing any more investigation. Android apps can be disassembled or decompiled using a variety of reverse engineering tools, including dex2jar, Apktool, Android Multitool, and the JADX. During this stage, we make use of Androguard, a tool for static analysis reverse engineering that is open source.

5.2. Extractions stage

Android SDK (software development kit) offers programmers a combination of API calls (comprising of a fundamental collection of package constructors, classes, fields, and

methods) that they can use to communicate with the operating system, software, or hardware as shown in Fig. 4. The SDK offers a wide range of APIs for developers to choose from when building an app. Malware writers can use these API calls to exploit mobile devices illegally. The same API call, for instance, may be requested by a benign or malicious program to access and receive particular data from an OS. There are several libraries included with the Android SDK as depicted in Fig. 4, including Android, Junit, and Org. The "android.jar" file in the Android SDK contains a reference to these libraries. API call features and Permission features are assigned to each Android app individually. We used the following method to extract permission requests and API calls from APK files.

A script developed in Python programming language which automatically runs and decompiles the complete dataset as follows:

1. Androguard can be used to generate all the different packages called from within an APK.
2. You can get the API call details and package level information from the entire package if it contains important methods and classes (like Java, and Android). More crucially, a few delicate API calls are shielded by Android's permission system, making them critical.
3. To extract the apps' requested permissions we used the methodology described by the authors [29], and we defined the set of all requested API calls, and all Android Permissions in the following way.

$$D_i = \{D_1, D_2, \dots, D_n\} \quad (1)$$

4. Each application should be represented in a form of a binary vector of API calls,

Where $App_i = \begin{cases} 1 \\ 0 \end{cases}$

If API is utilized in the application and if the corresponding application does not use API.

5. The association map is defined as follows D_i to map API calls to permissions P_i

$$A = \{(P, D | P \epsilon P, D \epsilon D\} \quad (2)$$

Where P is controlling the D.

6. For each permission, calculate the number of API calls and the numerical count for each API request, as follows:

$$MP = \{MP_1, MP_2, \dots, MP_n\}, \quad (3)$$

Where $MP_i = \begin{cases} 1, if \exists D_i \\ 0, if \nexists D_i \end{cases}$

$$C_i = \sum D_i | (P_i, D_i)$$

6. GROUPING STAGE

We used a grouping method to better highlight the complexities of utilization of the API calls in malignant applications in order to give comprehensive coverage of the detection performances..

7. OBSCUR

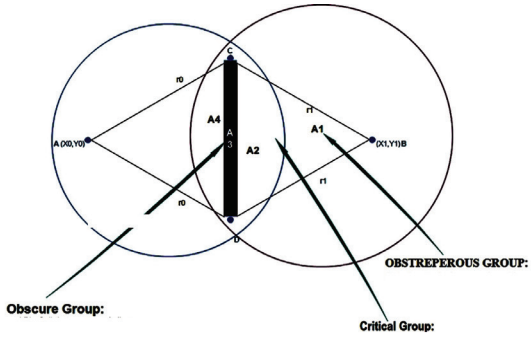


Fig 2: Architecture for ranking the API calls

This allowed us to present a high-cost coverage of the detection performances. We could create API call groups that were quite effective in spotting fraudulent apps. We designated as irrelevant API calls those calls that were in benign apps but were deemed insignificant. We concentrated on the APIs utilized by the malicious applications the most. Some of the aspects we looked at were prevalent in malicious applications. By using complementing techniques that avoid fingerprinting malware, this system aimed to classify API calls into three distinct levels and then classify them according to the level of hazard they posed. For the investigation, we used a full set of 15,036 API calls, each of which might be used by malicious programmers to carry out a variety of tasks throughout the system. Our research identified three distinct types of API requests founding malicious apps: Obscure (A3), Critical (A2), and Obstreperous (A1). We figured out which API methods were used more frequently in malware as compared to benign applications and then took the intersection of those two numbers. The ' critical API calls' category is what we term it because malicious apps frequently use these APIs rather than benign apps.

8. OBSCURE GROUP (A3)

We found intersections between API calls used by both malicious and benign applications and discovered that the total count of API calls in both benign and malicious apps was about equal. When looking at the API calls made by both malware and benign applications, the frequency with which each API request is made is also considered. For the sake of clarity, we took into consideration the frequency with which each API call is utilized by the benign applications. When it came to potentially harmful calls, we followed the same line of thinking. Thus let represent a collection of API calls which are used by malicious applications and their frequency and represent a collection of API calls which are used by benign applications and their corresponding frequency. Thus let $M=\{M1,M2,\ldots,Mj\}$ represent a collection of API calls which are used by malicious applications and their frequency and $C=\{C1,C2,\ldots,Ci\}$ represent a collection of API calls which are used by benign applications and their corresponding frequency. Thus, we isolated the obscure combination by extraction of API calls that appeared approximately equally in benign and malicious applications. In Area 3 (see Fig. 5), the calls are obscure (A3). We took a set theory method and made use of the intersection process in order to differentiate between benign and malware combinations. This process detects API requests which are frequent and similar to one another. However, a threshold value was added as a result of a shift in the frequency of API calls; because of this threshold constant, the classification of API calls as either malicious or benign and therefore belonging to obscure or critical groups is unstable. Out of a total of 15,036 calls, only 1,687 were placed in the obscure group.

9. CRITICAL GROUP (A2)

We were able to obtain the intersection of the API calls that were present in malicious applications more frequently than they were in benign applications by using our data. Since harmful apps often use these API calls rather than benign apps, we label this group "the critical API calls group. Thus let $M=\{M1,M2,\ldots Mj\}$ represent a collection of API calls which are used by malicious applications and their frequency and

let *C={C1,C2,…Ci}* represent a collection of API calls which are used by benign applications and their corresponding frequency. As illustrated in Figure 7, we can extract Area 2 (A2) by extracting the API calls which were used regularly in malicious files rather than in benign files since we know the exact time each API call appeared in both benign and harmful apps. The set-theoretic intersection process that takes place between the benign and malicious groups was implemented in the 'critical' group as we did in the first group. This was done in order to assess the level of danger posed by the 'risky' group. However, the characteristic that distinguished this group from others was that the frequency of these API calls for malicious applications was significantly higher than the frequency of API calls for clean apps. Because of this, a threshold value must be used when using the intersection operation to detect comparable API requests. API calls from benign apps are less likely to be considered active, specified, and notable even when the threshold is set at a high value. In other words, malicious API calls are increasingly common. C and D are the intersection points of the two circles (benign and malicious circles). Let A be center point of the benign circle (x_0, y_0) of radius r_0 and B be the center point of the malware circle (x_1, y_1) of radius r_1. There are three subareas at the intersection part *{A2,A3,…,A4}* that need to be calculated (see Figure 7), *{A2,A3,…,A4}* make up the right, left, and center sides of the intersection, respectively. The area A3 contains confusing API calls, while A2 contains critical API calls.

Step 1: We may compute the intersection's total area by using Area *{A2,A3,…,A4}*

Step 2: *{A2,A3,…,A4}* In order to determine the three individual subareas

$$A_2 = A_{pie}(CBD) - A_{CBD}$$
$$A_3 = A_{pie}(CD) - A_{CD}$$
$$A_4 = A_{pie}(DAC) - A_{DAC}$$

Step 3: Because the intersection is connected to the two circles, the area's angle where the circles meet (area that looks like a pie) may be stated using the following relation

$$A_{pie} * 2\pi = a * A_{circle}$$
$$A_{pie} = a * \frac{A_{circle}}{2\pi}$$
$$A_{pie} = a * \frac{\pi r^2}{2}$$
$$A_{pie} = 0.5 * a * r^2$$
$$A_{pie} = (DAC) = .5 * \overline{DAC} * r_0^2$$
$$A_{pie} = (CBD) = 0.5 * \overline{CBD} * r_1^2$$
$$A_{pie} = (CD) = 0.5 * \overline{CD}$$

Step 4: Use the cosine rule to get the angles.

$$r_0^2 = r_1^2 + AB^2 - 2 * r_1 * AB * \cos(CBA)$$

Step 5: We can compute distance AB using the coordinates of point A and point B:

$$AB = sqrt(x_1 - x_0)^2 - sqrt(y_1 - y_0)^2$$
$$\cos(BAC) = r_0^2 + AB^2 + \frac{r_1^2}{2} * r_0 * AB$$
$$\overline{BAC} = acos(r_0^2 + AB^2 + \frac{r_1^2}{2} * r_0 * AB)$$
$$\overline{ABD} = acos(r_1^2 + AB^2 + \frac{r_0^2}{2} * r_1 * AB)$$

Step 6: To get the triangles, we can compute the following: Knowing the distances and angles between two triangles

$$A_{DAC} = 0.5 * r_0^2 \sin(\overline{DAC})$$
$$A_{CBD} = 0.5 * r_0^2 \sin(\overline{CBD})$$

Step 7: We have reached at the final step where we can calculate the total area. $Area = A_1 + A_2 + A_3)$

$$A = A_{pie}(DAC) - A_{DAC} + A_{pie}(CBD) - A_{CBD} + A_{pie}(CB) - A_{CB}$$
$$A = 0.5 * (\overline{DAC}) * r_0^2 - 0.5 * r_0^2 * sin(\overline{DAC}) + 0.5 * (\overline{CBD}) * r_1^2 - 0.5 * r_0^2 * sin(\overline{CBD}) + 0.5 * (\overline{CD})$$

Following that, extraction of the linked API calls with malicious applications is allowed, as

illustrated in Fig. 7. The building of the intersection of value combinations and frequency displayed details regarding such characteristics which were present within several malware applications. When compared to the count of presence within benign applications, value combination of API calls used among the malignant applications has a higher total count. To demonstrate this idea, let's say that a particular API call was required 10 times by the concerned malware combination, but the benign set only asked it twice. As a result, given that the sample API call was found to occur more frequently in the malicious dataset, we have reason to believe that it is connected to a malicious dataset. In addition, there were only 737 potentially critical calls out of 15,036 total calls. While comparing it to the benign dataset, we see that the malicious dataset makes a significantly higher number of API calls in order to communicate with the system. For instance, the collected malware apps use the APIs for telephone controller, short message service manager, storage, system service, logs, databases, and device details often more than benign application does. This is because malware applications are designed to exploit vulnerabilities in mobile operating systems. The differential ranking of the API calls is mentioned in Table 2.

Table 3 presents a subgroup of multiple API calls that were in this set and which are utilized much regularly among malicious applications than they are in benign applications. This was necessary because of the limited amount of space available. The characteristics that malicious application employ requires critical API calls to get access to the system, according to our investigation. Examples include "*detDeviceId*" and "*getSubscriberId"* methods for stealing sensitive data such as (IMEI) and Identity (IMSI) numbers and sending them through the network using *setWifiEnabled* or *execHtpRequest*. Malware programs can be affected from techniques linked to sending messages and receiving messages (such as "*sendTextMessage," "getDefault,",* and ". *SetMessage*") according to the findings. The malware dataset, it turns out, affects obfuscation and other static analysis elimination strategies (e.g., *Cipher.getInstance*).For this reason, we hypothesized that classes like "*Getdeviceid*" and "*TelephonyManager*" could have needed additional rights to keep them safe from potentially malicious apps like "*SmsManage*" and "*SmsMessage".*Some API requests, such as "*Getdeviceid*,","*Getsubscriberid*," and "*Setwifienabled*, were already blocked by Google permissions.

10. OBSTREPEROUS GROUP (A1)

We only included APIs that were found in malicious apps and were absent from benign ones. Illustrated in Fig. 7. $C=\{C1,C2,\ldots,C_i\}$ Let C be a product of the collection of API requests which happen most frequently among benign applications is C_i while $M=\{M1,M2,\ldots,M_j\}$ is defined as API calls that appear frequently in malware programs. We compute the following to find the Obstreperous calls:

$$R = {}^{M}/_{C} \qquad (4)$$

In relation to the equation shown above, the characteristics of obstreperous calls are more obviously geared toward harmful applications. Concerning Fig. 7, in contrast to the other two categories, no explicit criterion was found for the frequency of API calls because this set is unquestionably more skewed from malignancy. Because no single particular frequency term must be met, and because there is a greater

potential for malicious API calls in most of these conditions, a previously implemented threshold is rendered meaningless, and its function is rendered moot within the context of this scenario. API generation technology such as the one described here was employed to create API requests that were then incorporated into the malicious dataset in their entirety. The outcomes of the experiments demonstrate several API calls. (i,e.

(*Lorg/w3c/dom/DOMException.getMessage*,) (*Java/lang/Thread;.setContextClassLoader,)* (*Android/content/Context;.deleteFile,)* *(Android/database/sqlite/SQLiteDatabase;query, Java/net/URL;.openConnection,)* (*sAndroid/telephony/TelephonyManager;getLine1Number)* are specifically discovered in malicious applications, not in benign applications. Among the 15,036 API calls, only 4 are found to be Obstreperous calls.

TABLE 2

DIFFERENTIAL RANKING OF THE API CALLS

API Calls Name	Meaning
Android/Telephony/ Telephonymanager;. Getnetworkoperator	To gain access to sensitive data
Java/Util/Gregoria ncalendar;. Set	To gain access to sensitive data (Current Time)
Java/Io/Bytearrayo utputstream; Reset	To gain access to se nsitive data
Java/Lang/Stringb uffer;.Insert.	For obfuscation purposes
Cipher. Getinstance ()	For obfuscation purposes
Sendtextmessage () Smsmanager () Setmessage ()	in order to send and receive SMS messages
Setwifienabled () Exechttprequest ()	For communicating over the network
Getdeviceid () Getsubscriberid ()	To gain access to sensitive data (phone's unique device ID)
RuntimeException ()	For the execution of external commands

Get Last Known Location: it communicates the device's location information to a remote site and returns the device's last known location from the specified provider. This technique is employed by the Geinimi family.

Get Line 1 Number: It sends sensitive information, such as a phone number, to a remote server as a string; we've seen this on nearly all Android devices. This is how the Fokonge family do things.

Set Context Class Loader: It can be used to dynamically load harmful software since it loads exterior classes or sources from certain repositories. Malware applications could use the Class Loader class to replace the corresponding software with malicious software to get around current countermeasures. It is quite likely that the malicious code is concealed either underneath the next route (/assets) or within the safe digital (SD) card. This strategy appears to be used by most members of Android. Steek family. Malicious software only requested permission for full Internet access once throughout the installation. It might appear as a smaller threat to possible sufferers if this malicious software just asks for permission during installation. Installing the malware on a smartphone triggers it to open and show information about any fraudulent apps that have been installed. The set Context Class Loader technique is called a similar amount of times by malicious applications that belong to the Steek family.

DOM Exception: It is possible to use it when certain events take place. In our research, we discovered that malicious apps used *Lorg/w3c/dom/DOMException)*

(Lorg/w3c/dom/DOMException;<init>.(SJava/lang/String;)V) .This method appears to be used by much of Android Steek group.

Open Connection: This method is belonging to Android. Generisk group. The group establishes linking to predetermined distant server, loads it, and then runs the code that it contains.

11. STATISTICAL ANALYSIS

Figure.3 summarizes all the attributes from both the “critical calls" as well as the “Obstreperous calls" groups that we considered in our experiment.

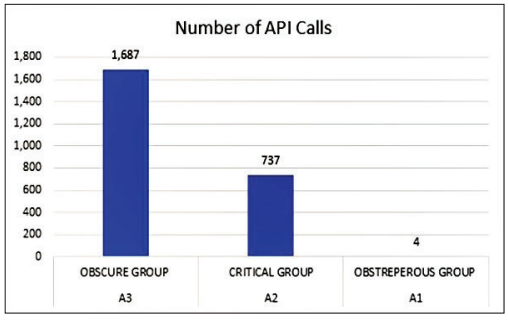


Fig 3: Distribution of Features across the Groups

In order to choose the most relevant characteristics from those that were available, we computed the IG for each one of them. Classifications of API calls can be made with reasonable ease using the primary categories. There is a variety of APIs available for each type of group. The following criteria will be used to assign a value to each feature: ‘very important’, ‘important’, ‘normal’, or ‘unimportant’.

Figure.4 demonstrates that the IG has been implemented in each feature. The score indicates the importance, in the opinion of the IG, of each of the best 12 characteristics found within the risky set. It approves that in mobile malware detection, the features chosen are very much important. In order to determine the significance of each feature to the data set that has been provided, the procedure calculates the splitting conditions regarding decision trees. Each permission's IG is determined by the formula below.

gain (c,r_i)=entropy (c)-entropy (c|r_i) C refers to the class value (i.e., malignant or benign) and is the attribute. The entropy (*c*) is the information entropy. The ideal collection of features depends on the classifier and is fewer than the total number of available features. We begin by utilizing all of the features in combination with ML techniques, and after that, we pick the attributes for evaluation from either the critical calls or Obstreperous calls.

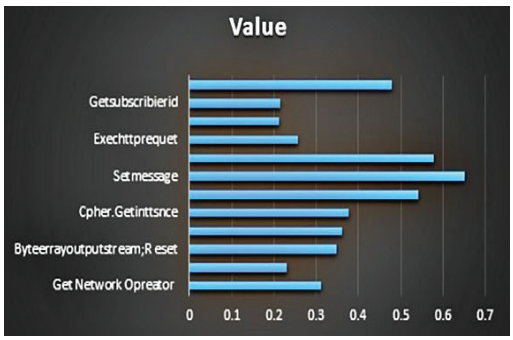


Fig 4: Information gain for the top features in the critical group

12. LEARNING-BASED DETECTION

12.1. Data normalization

Before using ML algorithms, it is critical to normalize the data. Now that we have compiled the essential characteristics (critical and Obstreperous attributes set), weights need to be assigned for those attributes and express them using a vector space.

Normalization of the term frequency (TF) was used in order to minimize situations in which the classifier has varying weights when making decisions. The represents the extracted dictionary, where the dictionary was drawn from both groups of data (critical and Obstreperous groups). A weighted vector space, $(w_{1n}, \ldots w_{wn}{}^{""}, w_{in} \in \{0, 1\}$ shows the presence or absence of a precise attribute in an app in the form of a (TF) representation. This word denotes the regularity with which the functionality can be accessed within the application. The (TF) can be scaled to values by the division of the frequency of an appearance by the amount of features within the application (0, 1). The following is a formula for calculating the $tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$ (TF) the normalization of the dataset provides for a matrix-like view of the vector representation where rows indicate application vectors and columns represent features. Performing this action enables the application of a variety of ML algorithms, and it also enables us to identify areas of similarity and difference by employing similarity-measuring algorithms.

13. EVALUATION METRICS

In order to determine the effectiveness of classification models, we have selected the following: accuracy, precision, recall, and the F-measure standard metrics. Estimates for these measures are derived from values of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN).

- **TP:** The count of correctly discovered malware applications is represented.
- **TN:** The count of correctly discovered benign applications is represented.
- **FP**: Count of benign apps mistakenly categorized as malware applications.
- **FN:** Count of malware an application mistakenly categorized as benign applications.

Accuracy It calculates an estimate of the proportion of successfully detected connection records relative to the total test dataset. When there is a higher level of accuracy, the ML model is considered to be superior. The accuracy is a useful measurement for the test dataset since it consists of classes that are evenly distributed, and it is explained in following manner:

$$Accuracy = {(TP + TN)}/{(TP + TN + FP + FN)}$$

Precision You can figure out the percentage of correctly identified data by doing the following calculation.

$$\boldsymbol{Precision = {TP}/{(TP + FP)}}$$

Recall You can figure out the percentage of correctly identified malicious data by doing the following calculation.

$$Recall = {TP}/{(TP + FN)}$$

F-Measure The following formula can be used to calculate the precision and recall combined

$$F\ Measure = 2 * \frac{Precision * \text{Recall}}{Precision + \text{Recall}}$$

14. RESULTS AND ANALYSIS

The primary objective is to investigate whether the considered attributes from the critical set and Obstreperous calls can be utilized to construct complicated classifiers that can forecast the classes of mobile malware, or the risk factors associated with it. After ignoring the features of the dataset that were deemed to be insignificant, all 741 different features were

collected from either the critical or Obstreperous categories. 6 ML algorithms J48, random forest (RF), k-nearest neighbors (k-NN), random tree (RT), naive Bayes (NB), Support vector machine (SVM) were used in 10- fold cross-validation for each group (critical group and Obstreperous group). The empirical findings imply that the proposed method is effective at recognizing mobile malware, as evidenced by the fact that it attained an F-measure of 94.04\%, as displayed in Fig. 8. In the process of conducting mobile application analysis and forensic investigations into malware, our model can substantial assistance.
The k-NN and random tree algorithms are the fastest when it comes to training and testing a classifier; both require 200 milliseconds. J48 is the most time-consuming, requiring 920 milliseconds. SVM takes 980 milliseconds in order to complete the training and testing, and for a random forest, an average of 0.73 seconds is required. Overall, the system is predictable and dependable in real-time applications, with a speed that is suitable for all five classifiers.

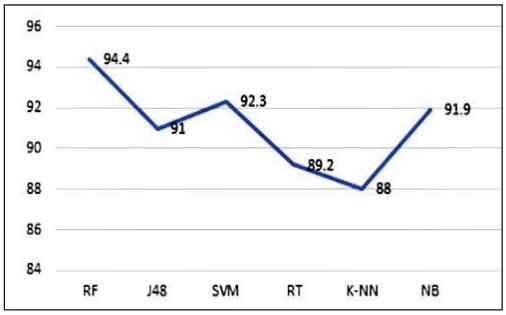


Fig 5: F Measure Score

15. CONCLUSION

The development of a secure mobile computing environment, the protection of sensitive data, and the detection of malicious software all need the identification of the most prevalent features demanded by malware applications. The behavior of an Android app is reflected in permissions and important API calls that the program makes. To identify malicious programs, we present a classification approach that includes the consideration of authorizations and API requests. This was provoked by the growing count of applications and the absence of efficient malware recognition technologies. There are three stages to our analysis: preprocessing, extraction, and grouping stage. With so many APIs used by Android apps, we devised a grouping method in order to target only the top important ones to increase the chances of finding Android malware.

Obscure set (common API calls in both malicious and normal applications).

Critical set (common API calls in a malicious application which are less like those in normal applications).

Obstreperous set (API calls that are present in the malicious applications and absent in normal applications).In order to find the top discriminating set of attributes for malware detection, a frequency examination is run on the important groups. As a result of the findings, it's clear that malicious Android apps make a distinct set of API calls and request permissions more frequently than normal Android applications to access user data than benign apps. For instance, the API requests for the SMS manager, storage, telephone manager, system service, device information, logs, and database are substantially more prevalent in malware applications. According to our suggested method's empirical results, which used an actual malware dataset of 15,036 Android applications, it is successful at recognizing mobile malware and can greatly

contribute in malware forensic investigation and mobile app analysis. Using IG and API frequency calculations, a useful subset of features is narrowed down, and the TF is then utilized to reduce the dimensionality of the narrowed-down set. The J48, k-NN, RT, RF, and NB algorithms are among the ML approaches we use in our research. The findings of the experiments show that our model is capable of reaching an F-measure of 94.03%.

REFERENCES

[1] R. Husnain, A. Nauman, A. Muhammad, I. Biju and R. Hamid, "AndroMalPack: enhancing the ML-based malware classification by detection and removal of repacked apps for Android systems," Scientific Reports, vol. 12, no. 1, pp. 19-34, 2022.

[2] F. Faezeh, H. M. Sayad J. Alireza and A. Mamoun, "Artificial intelligence for detection, estimation, and compensation of malicious attacks in nonlinear cyber-physical systems and industrial IoT," IEEE transactions on industrial informatics, vol. 16, no. 4, pp. 2716-2725, 2019.

[3] A. M. Taleby, L. Qianmu, R. Mahdi and R. A. Raza, "A survey on smartphones security: software vulnerabilities, malware, and attacks," arXiv preprint arXiv: 2001.09406, 2020.

[4] M. Anjali, Permissions Ranking With Statistical Techniques for Android Malware Detection, “Doctoral dissertation, 2022.

[5] M. Sreenath and S. Anuradha, “The political economy of digital automation: measuring its impact on productivity, economic growth, and consumption, “Routledge, 2020.

[6] Z. Jason, "Machine learning with feature selection using principal component analysis for malware detection: a case study," arXiv preprint arXiv: 1902.03639, 2019.

[7] A. Saba, S. M. Ali, A. Khan and A. Mansoor, "Android malware detection \& protection: a survey," International Journal of Advanced Computer Science and Applications, vol. 7, no. 2, 2016.

[8] P. Faruki, B. Ammar, V. Laxmi, Ganmoor, Vijay and Gaur, Manoj Singh and Conti, Mauro and Rajarajan, Muttukrishnan, "Android security: a survey of issues, malware penetration, and defenses," IEEE communications surveys \& tutorials, vol. 17, no. 2, pp. 998-1022, 2014.

[9] Z. Yajin and J. Xuxian, "Dissecting android malware: Characterization and evolution," in 2012 IEEE symposium on security and privacy, IEEE, pp. 95-109, 2012.

[10] R. Vaibhav, C. Yan and J. Xuxian, "Droidchameleon: evaluating android anti-malware against transformation attacks," in Proceedings of the 8th ACM SIGSAC symposium on Information, computer and communications security, pp. 329-334, 2013.

[11] I. Mülhem, B. Issa, and M. B. Jasser., "A Method for Automatic Android Malware Detection Based on Static Analysis and Deep Learning," IEEE Access, vol. 10, pp. 117334-117352, 2022.

[12] B. Shikha, and S. Muttoo, "Evading android anti-malware by hiding malicious application inside images," International Journal of System Assurance Engineering and Management, vol. 9, pp. 482-493, 2018.

[13] I. Rejwana, S. M. Islam, S. Sajal, H. M. Jamal and M. Md Abdul, "Android malware classification using optimum feature selection and ensemble machine learning," Internet of Things and Cyber-Physical Systems, vol. 3, pp. 100-111, 2023.

[14] W. Chao, X. Qingzhen, L. Xiuli and L. Shouqiang, "Research on data mining of permissions mode for Android malware detection," Cluster Computing, vol. 22, pp. 13337-13350, 2019.

[15] D. Shuaifu, W. Tao and Z. Wei, "Droid-Logger: Reveal suspicious behavior of Android applications via instrumentation," in 2012 7th international conference on computing and convergence technology (ICCCT), IEEE, pp. 550-555, 2012.

[16] S. A. Kumar, C. D. Jaidhar, and K. MA Ajay, "Experimental analysis of Android malware detection based on combinations of permissions and API-calls," Journal of Computer Virology and Hacking Techniques, vol. 15, pp. 209-218, 2019.

[17] Tao, Guanhong, Zibin Zheng, Ziying Guo, and Michael R. Lyu, "MalPat: Mining patterns of malicious and benign Android apps via permission-related APIs," IEEE Transactions on Reliability, vol. 67, no. 1, pp. 355-369, Dec. 2017.

[18] A. Abdelfattah, R. Jean-Marc and T. Chamseddine, "Enhancing malware detection for Android systems using a system call filtering and abstraction process," Security and Communication networks, vol. 8, no. 7, pp. 1179-1192, 2015.

[19] P. Vinod, Z. Akka and C. Mauro, "A machine learning based approach to detect malicious android apps using discriminant system calls," Future Generation Computer Systems, vol. 94, pp. 333-350, 2019.

[20] Z. Aqil, H. I. Rahmi, S. Wahidah Md and A. Zubaile, "Android malware detection based on network traffic using decision tree algorithm," in Recent Advances on Soft Computing and Data Mining: Proceedings of the Third International Conference on Soft Computing and Data Mining (SCDM 2018), Johor, Malaysia, Springer, pp. 485-494, 2018.

[21] W. Shanshan, Y. Qiben, C. Zhenxiang, Y. Bo, Z. Chuan and C. Mauro, "Detecting android malware leveraging text semantics of network flows," IEEE Transactions on Information Forensics and Security, vol. 13, no. 5, pp. 1096-1109, 2017.

[22] W. Ping, C. W. Jie, C. Kuo-Ming and L. Chi-Chun, "Using taint analysis for threat risk of cloud applications," in 2014 IEEE 11th International Conference on e-Business Engineering, IEEE, pp. 185-190, 2014.

[23] B. James, A. Mohd and D. Gerry, "Detection of mobile malware: an artificial immunity approach," in 2016 IEEE Security and Privacy Workshops (SPW), IEEE, pp. 74-80, 2016.

[24] K. Pallavi and J. Amit, "Malware detection techniques in android," International Journal of Computer Applications, vol. 122, no. 17, 2015.

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

[26] Z. Hanqing, L. Senlin, Z. Yifei and P. Limin, "An efficient Android malware detection system based on method-level behavioral semantic analysis," IEEE Access, vol. 7, pp. 69246-69256, 2019.

[27] F. Hossein, M. Veelasha, C. Mauro and B. Lejla, "Efficient classification of android malware in the wild using robust static features," Protecting mobile networks and devices: challenges and solutions, vol. 1, pp. 181-209, 2016.

[28] K. ElMouatez Billah, D. Mourad, D. Abdelouahid and M. Djedjiga, "Mal-Dozer: Automatic framework for android malware detection using deep learning," Digital Investigation, vol. 24, pp. S48-S59, 2018.

[29] Q. Mengyu, S. Andrew and L. Qingzhong, "Merging permission and api features for android malware detection," in 2016 5th IIAI international congress on advanced applied informatics (IIAI-AAI), IEEE, pp. 566-571, 2016.

Asma *et al. (IJECI)* 2023

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

International Journal for
Electronic Crime Investigation

DOI: https://doi.org/10.54692/ijeci.2023.0704164

Research Article

Vol. 7 issue 4 Oct-Dec 2023

Integration of Cloud Computing and Wearable Technology for Enhanced Interactivity

Asma Batool and Humaira Naeem

Department of Computer Science, Virtual university of Pakistan, Lahore, 54000, Pakistan.

Corresponding author: humairanaeem@vu.edu.pk

Received: September 22, 2023; **Accepted**: November 12, 2023; **Published**: September 20, 2023

ABSTRACT

The emergence of wearable computing has revolutionized the way we interact with technology, blurring the lines between the physical and digital worlds. In this research, we explore the dynamic interaction between cloud technology and wearable computing, a synergy that is reshaping the landscape of personal technology and data management. The study delves into how cloud computing provides a powerful platform for wearable devices, enabling enhanced data processing capabilities, storage, and ubiquitous access to information. We investigate the various applications of this interaction, ranging from health monitoring to augmented reality, emphasizing the transformative impact on everyday life and various industries. The research also addresses the challenges inherent in this integration, such as data security, privacy concerns, and the need for robust, low-latency communication networks. Through a comprehensive analysis of current trends and future prospects, this study highlights the potential of cloud-assisted wearable technology in creating more personalized, efficient, and interconnected experiences. The findings suggest that the convergence of cloud technology and wearable computing not only offers significant benefits in terms of functionality and user experience but also poses critical considerations for data governance and ethical implications in an increasingly connected world.

Keywords: healthcare, wearable, leverage, integrated, computing, architecture.

1. INTRODUCTION

Cloud technology and wearable computing are two important areas of computing that are rapidly advancing and changing the way we interact with technology. Cloud technology refers to the delivery of computing resources, such as data storage and processing, over the internet. This allows users to access and use these resources from anywhere, at any time, and on any device. Wearable computing, on the other hand, refers to the use of computing devices that can be worn on the body, such as smartwatches, fitness trackers, and smart glasses [1]. The interaction between cloud technology and wearable computing is of growing interest and importance, as it offers the potential to create new and innovative

solutions for a wide range of applications. By integrating cloud-based resources with wearable devices, it is possible to provide users with a more seamless, efficient, and effective computing experience. For example, cloud-connected wearable devices can provide users with real-time access to their health and fitness data, entertainment, and other applications, without the need for a separate computer or smart phone. The integration of cloud technology and wearable computing also has the potential to create new solutions for big data analysis, with the ability to process and store large amounts of data generated by wearable devices. This can lead to new insights and breakthroughs in a range of fields, such as healthcare, sports, and entertainment [2].

However, the integration of cloud technology and wearable computing also presents a number of challenges and technical issues that must be addressed. For example, the transfer of large amounts of data between wearable devices and cloud-based resources can be slow and unreliable, leading to delays and disruptions in the user experience [3].In addition, the security of sensitive personal data stored and processed in the cloud is a major concern, and must be addressed to ensure user privacy and data protection. Despite these challenges, the future of cloud technology and wearable computing interaction is bright, with ongoing research and development in this field poised to deliver new innovations and breakthroughs in the years to come [4]. This research is exploring new approaches and solutions for integrating cloud technology and wearable computing, including new hardware and software designs, improved communication protocols, and enhanced security mechanisms. The goal is to create wearable devices that are better connected to cloud-based resources and provide users with a more integrated and personalized computing experience [5].

The technological advancements in the past few decades have given rise to a new era of innovative devices and systems. One such domain that has greatly benefited from these advancements is healthcare. Wearable devices, combined with cloud computing, have created a revolution in the healthcare industry. Wearable Devices: Wearable devices are small, portable, and convenient devices that can be worn on the body to monitor various health parameters. These devices are equipped with sensors that collect and transmit data about the user's physical activity, heart rate, sleep patterns, and other health metrics [6]. Some examples of wearable devices include fitness trackers, smartwatches, and sleep monitors. Cloud Computing: Cloud computing is a technology that enables users to store and access data and applications over the internet. This technology provides the capability to store, process, and analyze vast amounts of data, which can be used for various purposes, including healthcare [7].

Benefits of Wearable Devices and Cloud Computing in Healthcare: Personalized Healthcare: Wearable devices and cloud computing provide patients with personalized healthcare services. By collecting and analyzing data from wearable devices, healthcare providers can create customized treatment plans that are tailored to the individual's specific needs. Continuous Monitoring: Wearable devices allow for continuous monitoring of a patient's health status. This enables healthcare providers

to detect and respond to health problems in real-time, reducing the risk of complications and improving patient outcomes. Improved Data Management: Cloud computing provides a secure and efficient way to manage and analyze large amounts of health data. This helps healthcare providers make better-informed decisions about patient care and improve the overall quality of care. Cost-effective: By leveraging the power of cloud computing, healthcare providers can reduce the costs associated with data storage and analysis. This, in turn, helps to lower healthcare costs for patients and improve access to quality care [4].

The AIWAC (Artificial Intelligence in Wearable and Cloud) Architecture is a framework that outlines the interaction between cloud technology and wearable computing. It consists of the following components: Wearable Devices: These are the physical devices worn on the body that collect and transmit data to the cloud. Examples include fitness trackers, smartwatches, and sleep monitors.Cloud Server: The cloud server is responsible for storing, processing, and analyzing the data collected from wearable devices [9]. This server can be a public cloud, private cloud, or hybrid cloud, depending on the security and privacy requirements of the data. Data Analytics: This component uses machine learning algorithms and statistical models to analyze the data collected from wearable devices. The insights generated from this analysis can be used to improve the health and wellness of the wearer. Application Layer: This layer consists of applications that run on the cloud server and interact with wearable devices. These applications can provide users with real-time feedback, alerts, and insights about their health and wellness. Networking: The networking component is responsible for establishing and maintaining communication between wearable devices and the cloud server. This component can use various communication protocols, including Bluetooth, Wi-Fi, and cellular networks. Security: This component ensures the confidentiality, integrity, and availability of data transmitted between wearable devices and the cloud server. It implements security measures such as encryption, access control, and data backups to protect the data [5].

In conclusion, the AIWAC Architecture provides a comprehensive framework for the interaction between cloud technology and wearable computing. By leveraging the power of cloud computing and wearable devices, this architecture has the potential to revolutionize the healthcare industry by providing patients with personalized and cost-effective healthcare services. The User Terminal Layer is an important component in the interaction between cloud technology and wearable computing. It refers to the interface between the wearable device and the user [10]. The key functions of this layer include Data Collection: The wearable device collects data from various sensors and transmits it to the cloud server. This data can include information such as heart rate, physical activity, sleep patterns, and other health metrics. User Feedback: The wearable device provides users with real-time feedback and insights about their health and wellness. This can include information such as the number of steps taken, calories burned, and hours of sleep. User Input: The wearable device allows users to input information, such as their diet, exercise, and mood. This informa-

tion can be used to create a more comprehensive picture of the user's health and wellness. User Interface: The wearable device provides users with a simple and intuitive interface for accessing and managing their health data. This can include features such as touch screens, voice commands, and button controls. User Experience: The user terminal layer plays a crucial role in shaping the user's overall experience with the wearable device. By providing a seamless and intuitive interface, users are more likely to use the device regularly and receive the maximum benefits from it [6].

In conclusion, the User Terminal Layer is a critical component in the interaction between cloud technology and wearable computing. By providing users with a simple and intuitive interface, this layer enables them to easily access and manage their health data and receive real-time feedback and insights about their health and wellness.The Communication Layer is a crucial component in the interaction between cloud technology and wearable computing. It refers to the mechanism through which data is transmitted between wearable devices and the cloud server [17]. Key functions of this layer include Data Transmission: The communication layer is responsible for transmitting data from wearable devices to the cloud server. This data can include information such as heart rate, physical activity, sleep patterns, and other health metrics.Data Transfer Protocols: The communication layer uses various data transfer protocols to transmit data between wearable devices and the cloud server. These protocols can include Bluetooth, Wi-Fi, and cellular networks, depending on the device's connectivity and security requirements.Data Synchronization: The communication layer ensures that data is synchronized between the wearable device and the cloud server. This allows users to access their health data from any device connected to the cloud, such as their smartphone or computer.Data Security: The communication layer implements security measures such as encryption and access control to protect the confidentiality and integrity of the data transmitted between wearable devices and the cloud server [7].

In conclusion, the Communication Layer plays a critical role in the interaction between cloud technology and wearable computing. By providing a reliable and secure mechanism for transmitting data, this layer enables wearable devices to effectively communicate with the cloud server and provide users with real-time feedback and insights about their health and wellness.The core of AIWAC, which provides physiological and psychological information evaluation through a statistics center on the cloud platform, is the cloud-based carrier layer [9]. The data center is primarily in charge of data storage, function extraction and classification, as well as person emotion modeling. utilizing the powerful computational power of gadgets. While the transmission module is in charge of transferring collected data to the sink node and receiving control signals, the acquisition module is utilized to gather physiological information. Only a few devices are active to gather the crucial physiological data and monitor emotional changes in a person while they are mentally secure. When a user's emotions change, In order to increase the accuracy of sentiment evaluation, the emotional weak deduction receiving layer sends a control signal to the wearable tool layer, which activates relevant devices or deactivates

unrelated ones if you wish to conserve energy [8].

The Cloud-based Service Layer is a key component in the interaction between cloud technology and wearable computing. It refers to the services and applications provided by the cloud server to support wearable devices. Key functions of this layer include Data Storage: The cloud server stores the data collected from wearable devices, allowing users to access their health data from any device connected to the cloud.Data Processing: The cloud server uses powerful computing resources to process the data collected from wearable devices. This includes applying machine learning algorithms and statistical models to generate insights and predictions about the user's health and wellness.Data Analysis: The cloud server provides data analytics services to generate insights and predictions about the user's health and wellness [19].This can include information such as the number of steps taken, calories burned, and hours of sleep.Application Development: The cloud server provides a platform for developers to create applications that interact with wearable devices. These applications can provide users with real-time feedback, alerts, and insights about their health and wellness.Scalability: The cloud server provides a scalable infrastructure that can handle an increasing volume of data from wearable devices. This allows wearable devices to scale up or down as needed, depending on the user's needs [9].

In conclusion, the Cloud-based Service Layer is a critical component in the interaction between cloud technology and wearable computing. By providing a platform for storing, processing, and analyzing data from wearable devices, this layer enables wearable devices to deliver personalized and cost-effective healthcare services to users.The Emotional Sensitive Deduction Receiving Layer is an important component in the interaction between cloud technology and wearable computing. It refers to the ability of the cloud server to use data from wearable devices to detect and interpret emotional states of users. Key functions of this layer include Emotion Detection: The cloud server uses data from wearable devices such as heart rate, skin conductance, and physical activity to detect the emotional state of users. This information can be used to infer emotions such as stress, anxiety, and happiness.Emotion Analysis: The cloud server applies machine learning algorithms and statistical models to analyze the data collected from wearable devices and infer the emotional state of users. This can provide users with real-time feedback about their emotional state and suggest ways to manage their emotions.Personalization: The cloud server provides personalized feedback and recommendations to users based on their emotional state. This can include information such as stress-relieving activities, mindfulness exercises, and lifestyle modifications. Privacy: The cloud server implements privacy measures to protect the confidentiality and security of the data collected from wearable devices. This includes encryption, access control, and data anonymization [10].

In conclusion, the Emotional Sensitive Deduction Receiving Layer is a critical component in the interaction between cloud technology and wearable computing. By using data from wearable devices to detect and interpret emotional states of users, this layer enables wearable

devices to deliver personalized and cost-effective emotional health services to users. The Multidimensional Affective Data Layer is a key component in the interaction between cloud technology and wearable computing. It refers to the collection and analysis of data related to the user's emotions, moods, and affective states. Key functions of this layer include Data Collection: The cloud server collects data from wearable devices such as heart rate, skin conductance, and physical activity to create a comprehensive profile of the user's emotional state. Data Analysis: The cloud server applies machine learning algorithms and statistical models to analyze the collected data to detect patterns and trends in the user's emotional state. This information can be used to generate insights and predictions about the user's emotional health. Personalization: The cloud server provides personalized feedback and recommendations to users based on their emotional state. This can include information such as stress-relieving activities, mindfulness exercises, and lifestyle modifications. Privacy: The cloud server implements privacy measures to protect the confidentiality and security of the data collected from wearable devices. This includes encryption, access control, and data anonymization [11].

In conclusion, the Multidimensional Affective Data Layer is a critical component in the interaction between cloud technology and wearable computing. By collecting and analyzing data related to the user's emotions, moods, and affective states, this layer enables wearable devices to deliver personalized and cost-effective emotional health services to users [12].

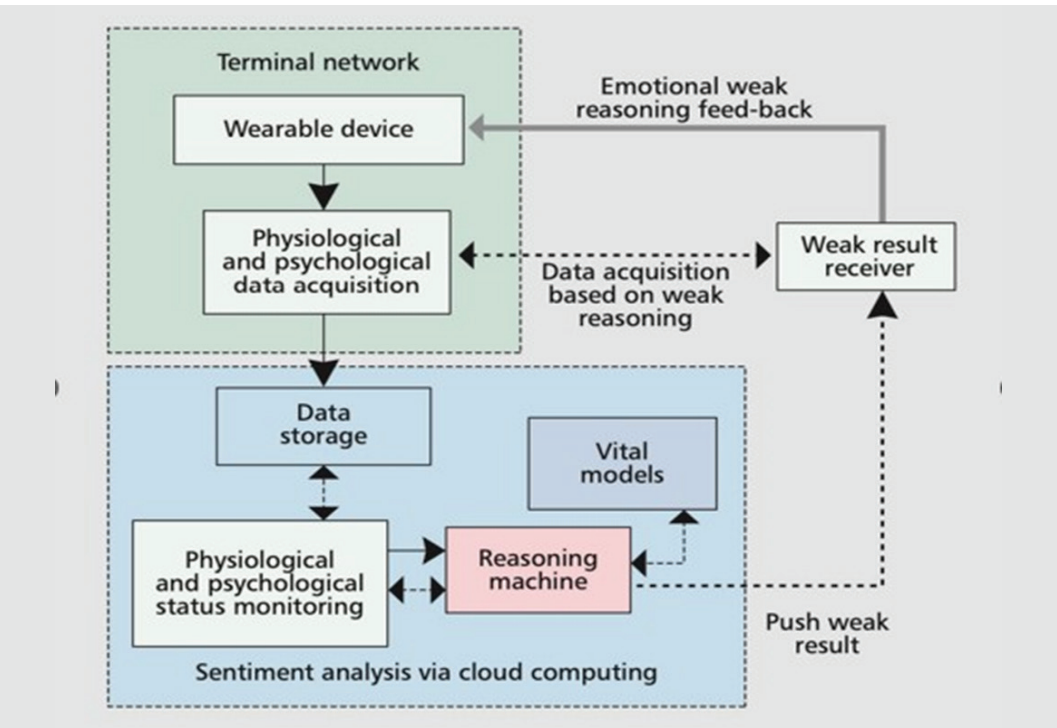


Fig 1: Weak deduction based multi component control mechanism

The evaluation of the Hybrid Big Emotion Data Layer is a crucial step in understanding the interaction between cloud technology and wearable computing. This layer refers to the combination of data from wearable devices and other sources, such as social media, to create a comprehensive profile of the user's emotional state. Key evaluation metrics for this layer include Accuracy: The accuracy of the data collected from wearable devices and other sources is a key evaluation metric. This includes the accuracy of the algorithms used to detect and interpret the user's emotional state.Privacy: The privacy measures implemented by the cloud server to protect the confidentiality and security of the data collected from wearable devices is a key evaluation metric. This includes encryption, access control, and data anonymization.Personalization: The level of personalization provided by the cloud server is a key evaluation metric. This includes the ability of the cloud server to provide personalized feedback and recommendations based on the user's emotional state.User Satisfaction: The level of user satisfaction with the services provided by the cloud server is a key evaluation metric. This includes the user's perception of the usefulness, ease of use, and effectiveness of the services provided [13]. In conclusion, evaluating the Hybrid Big Emotion Data Layer is crucial for understanding the interaction between cloud technology and wearable computing. By combining data from wearable devices and other sources, this layer enables wearable devices to deliver personalized and cost-effective emotional health services to users. The evaluation of this layer should consider factors such as accuracy, privacy, personalization, and user satisfaction.The Emotion Multidimensional Data Aggregation and Preprocessing Layer is a critical component in the interaction between cloud technology and wearable computing. This layer refers to the aggregation of data from wearable devices and other sources, such as social media, and the preprocessing of this data to prepare it for analysis. Key functions of this layer include Data Aggregation: The cloud server collects data from wearable devices and other sources and aggregates it into a comprehensive profile of the user's emotional state. This data may include heart rate, skin conductance, physical activity, and social media activity.Data Preprocessing: The cloud server applies preprocessing techniques such as cleaning, normalization, and feature extraction to the aggregated data. This helps to improve the accuracy and reliability of the data and prepare it for analysis.Privacy: The cloud server implements privacy measures to protect the confidentiality and security of the data collected from wearable devices and other sources. This includes encryption, access control, and data anonymization [14].

In conclusion, the Emotion Multidimensional Data Aggregation and Preprocessing Layer is a critical component in the interaction between cloud technology and wearable computing. By aggregating data from wearable devices and other sources and preprocessing this data, this layer enables wearable devices to deliver personalized and cost-effective emotional health services to users. The privacy measures implemented by this layer help to protect the confidentiality and security of the user's data.

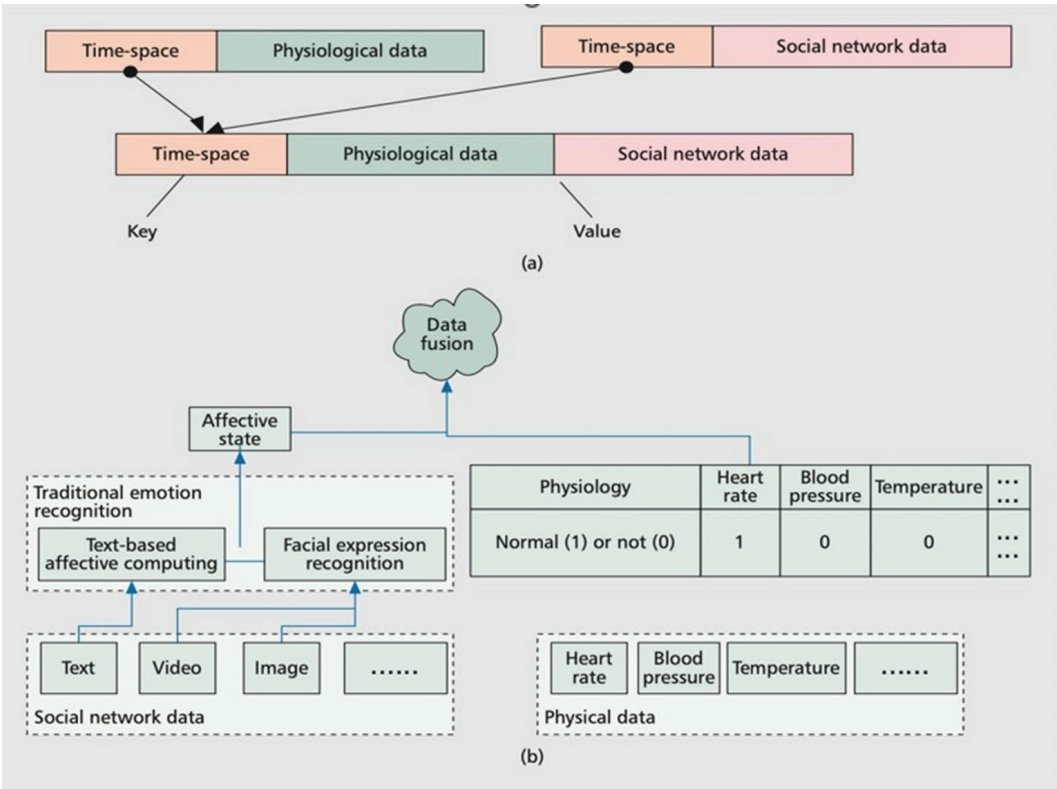


Fig 2: Structure and fusion of multidimentional data: a) data structure with time space label as the key; and b) various emotional data fusion.

A testbed architecture refers to the hardware and software components used to test and evaluate the interaction between cloud technology and wearable computing. A typical testbed architecture for evaluating the interaction between these two technologies includes the following components Wearable Devices: This includes a range of wearable devices, such as smartwatches and fitness trackers, that are capable of collecting data about the user's emotional state. Cloud Server: This includes a cloud-based server that aggregates and preprocesses the data collected from wearable devices and other sources. User Terminal: This includes a user interface that allows users to interact with the cloud server and access their emotional health data. This may include a web-based interface, mobile app, or wearable device. Emotion Detection and Analysis Algorithms: This includes algorithms that are used to detect and interpret the user's emotional state [5]. These algorithms may be implemented on the wearable devices or on the cloud server. Data Storage and Management System: This includes a system for storing and managing the data collected from wearable devices and other sources. This system may include a database management system, data warehousing system, or cloud storage system.

In conclusion, a testbed architecture is a crucial component in evaluating the interaction between cloud technology and wearable computing. The components of a testbed archi-

tecture, such as wearable devices, cloud server, user terminal, emotion detection and analysis algorithms, and data storage and management system, work together to provide a comprehensive evaluation of the interaction between these two technologies. The interaction between cloud technology and wearable computing involves the exchange of data and information between wearable devices and cloud-based servers. The technical information involved in this interaction includes: Data Format: Wearable devices and cloud servers must use a common data format for exchanging information. Common data formats include JSON, XML, and CSV. Data Transfer Protocols: The communication between wearable devices and cloud servers must be secure and efficient. Common data transfer protocols used for this interaction include HTTPS, MQTT, and WebSockets. Data Processing: The cloud server must have the capability to process and analyze large amounts of data. This includes data preprocessing techniques such as cleaning, normalization, and feature extraction, and machine learning algorithms for data analysis. Data Security: Wearable devices and cloud servers must implement security measures to protect the confidentiality and privacy of the user's data. This includes data encryption, access control, and data anonymization. Data Visualization: The cloud server must have the capability to visualize and display the analyzed data in a user-friendly format. This may include charts, graphs, and reports [16].

In conclusion, the technical information involved in the interaction between cloud technology and wearable computing includes data format, data transfer protocols, data processing, data security, and data visualization. These technical elements work together to ensure efficient and secure communication between wearable devices and cloud servers and enable wearable devices to deliver personalized and cost-effective emotional health services to users.The interaction between cloud technology and wearable computing is a rapidly evolving field with many open issues and prospective directions. Some of the key open issues and prospective directions include:Data Privacy and Security: Ensuring the privacy and security of user data is a major concern in the interaction between cloud technology and wearable computing. This includes protecting users' personal information, health data, and emotional states.Data Integration and Management: The integration and management of data collected from multiple wearable devices and other sources is a major challenge. This includes developing efficient methods for aggregating, preprocessing, and analyzing large amounts of data.Emotion Detection Accuracy: Improving the accuracy of emotion detection algorithms is a key challenge in the interaction between cloud technology and wearable computing. This includes developing algorithms that can accurately detect and interpret subtle emotional changes.User Interaction and Experience: Enhancing the user interaction and experience with wearable devices and cloud-based services is an important prospective direction. This includes developing user-friendly interfaces, wearable devices with improved ergonomics and aesthetics, and cloud-based services that provide personalized and cost-effective emotional health services.Interoperability and Standardization: Interoperability and standardization of data formats, data transfer protocols, and data processing methods are key

issues in the interaction between cloud technology and wearable computing. This includes developing standards for data exchange and processing that ensure seamless interaction between wearable devices and cloud-based services.In conclusion, the interaction between cloud technology and wearable computing is a complex and dynamic field with many open issues and prospective directions. Addressing these issues and exploring new directions is crucial in delivering personalized and cost-effective emotional health services to users through wearable devices and cloud-based services [17].

2. RELATED WORK

The integration of cloud technology and wearable computing has been an active area of research and development in recent years, and there has been a growing body of literature exploring various aspects of these interactions [3]. Some of the key areas of focus include Cloud-based wearable applications: Researchers have been exploring the use of cloud technology to support various applications on wearable devices, such as health monitoring, fitness tracking, and entertainment.Wearable-cloud integration: Researchers have been investigating ways to optimize the interaction between wearable devices and cloud-based services, such as improving data transfer and synchronization, reducing latency, and enhancing security [4]. Big data analysis for wearable devices: Researchers are exploring how to use cloud technology to process, store, and analyze the large amounts of data generated by wearable devices [6]. Wearable-cloud security: With the increasing amount of sensitive personal data being stored and processed in the cloud, researchers have been investigating various security issues associated with wearable-cloud integration, such as data privacy, data protection, and user authentication.Wearable-cloud architecture: Researchers have been proposing and evaluating various architectures for integrating wearable devices and cloud-based services, such as edge computing, fog computing, and hybrid architectures [12].These are some of the key areas of focus in the related work on cloud technology and wearable computing interaction. The literature in this field is rapidly evolving, and new developments and advances are being reported regularly [15].

3. PROPOSED METHODOLOGY

The proposed methodology for investigating the interaction between cloud technology and wearable computing can vary depending on the specific research question and objectives. However, some common steps that can be included in the methodology Problem definition: Clearly defining the problem that is being addressed and the goals of the research is an important first step in developing a methodology [18]. Literature review: Conducting a thorough review of the existing literature in the field of cloud technology and wearable computing interaction is important to understand the current state of the art and identify gaps in the existing knowledge. System design: Designing an appropriate system architecture for integrating wearable devices and cloud-based services is a critical step in the methodology. This can include selecting appropriate hardware and software components, defining communication protocols, and identifying data storage and processing needs.

Implementation and evaluation: Implementing the proposed system and evaluating its performance and effectiveness is an important step in the methodology. This can include conducting experiments or simulations, analyzing data, and comparing results with existing solutions. Discussion and conclusion: Finally, it is important to discuss the results of the research, draw conclusions, and make recommendations for future work in the field of cloud technology and wearable computing interaction. This is a general outline of the steps that can be included in the proposed methodology for investigating the interaction between cloud technology and wearable computing. The specific methodology will depend on the research question and goals, and may be adapted as needed based on the results of the research [19].

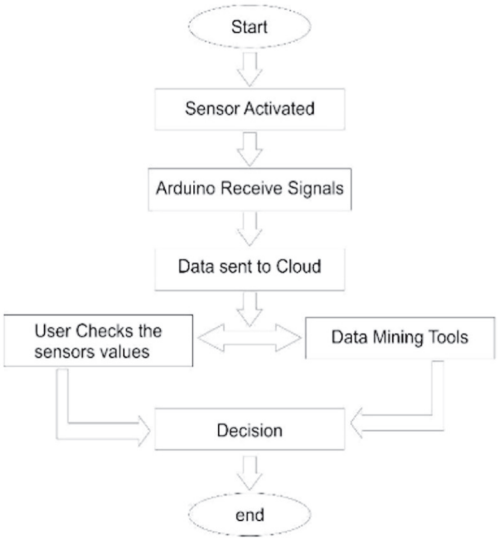


Fig 3: Flow chart of smart farm system

The Fig 3 showed the flowchart for a smart farm system. Here is a step-by-step description based on the provided image:
Start: The process begins.
Sensor Activated: A sensor in the system becomes active, likely due to a certain condition or threshold being met.
Arduino Receive Signals: The activated sensor sends signals to an Arduino board, which is a microcontroller used for processing the signals.

Data Sent to Cloud: The processed data from the Arduino is then transmitted to cloud storage or cloud-based services for further use.
Data Mining Tools: In the cloud, data mining tools are applied to the data. This might involve analyzing the data to extract useful information [20].

User Checks the Sensors Values: Simultaneously, there is a provision for users to directly check the sensor values. This could be via a dashboard or interface that displays the data.
Decision: Based on the output from the data mining tools and the user's assessment of the sensor values, a decision is made. This decision could relate to actions or changes in the smart farm system.

End: The process concludes following the decision.

This flowchart outlines a typical IoT-enabled smart farming operation where sensors collect data, which is then processed and analyzed in real-time, allowing for informed decision-making. The use of cloud technology enables data processing and storage at scale, and the incorporation of data mining tools suggests that the system is capable of supporting complex data analysis tasks for enhanced decision-making in smart agriculture.

4. RESULTS

The results and simulations of cloud technology and wearable computing interaction can vary greatly depending on the specific research question and objectives. However, some common results that may be obtained from such research include Improved performance: Simulations or experiments may show that integrating cloud technology and wearable computing can result in improved performance in terms of data transfer speed, processing power, and energy consumption. Enhanced user experience: Results may demonstrate that the integration of cloud technology and wearable computing can provide a more seamless and integrated user experience, with wearable devices that are better connected to cloud-based services and provide more advanced functionality. Increased efficiency: Research results may show that cloud technology and wearable computing interaction can lead to more efficient data processing and storage, with larger amounts of data being handled more quickly and effectively. Better security: Simulations or experiments may demonstrate that the integration of cloud technology and wearable computing can result in improved security for sensitive personal data, with better encryption and authentication mechanisms being employed. These are some of the results that may be obtained from research on cloud technology and wearable computing interaction. The specific results will depend on the research question and goals, and may vary widely based on the methods and simulations used [21].

Confusion Matrix for sheltering the fields:

	No	Yes	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
No	4	1	0.8	0	1	0.8	0.889	0.73	0.9	0.943
Yes	0	2	1	0.2	0.667	1	0.8	0.73	0.9	0.667
		Weighted Avg.	0.857	0.057	0.905	0.857	0.863	0.73	0.9	0.864

Confusion Matrix for Water supply in the fields

	OFF	ON	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area
OFF	7	0	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000
ON	0	3	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000
		Weighted Avg	1.000	0.000	1.000	1.000	1.000	1.000	1.000	1.000

5. CONCLUSION

In conclusion, cloud technology and wearable computing interaction is a rapidly growing field that offers significant potential for improving the way we use and interact with technology. The integration of these two areas of computing has created new possibilities for more seamless, efficient, and effective computing, with cloud-connected wearable devices offering users a more integrated and personalized experience. However, there are also many challenges that must be overcome in order to fully realize the potential of cloud technology and wearable computing interaction. These challenges include optimizing data transfer and processing, enhancing security, and improving the user experience. Despite these challenges, the future of cloud technology and wearable computing interaction is bright, with ongoing research and development in this field poised to deliver new innovations and breakthroughs in the years to come.

REFERENCES

[1] N. Tabassum, A. Namoun, T. Alyas, A. Tufail, M. Taqi, and K. Kim, "applied sciences Classification of Bugs in Cloud Computing Applications Using Machine Learning Techniques," 2023.

[2] M. I. Sarwar, Q. Abbas, T. Alyas, A. Alzahrani, T. Alghamdi, and Y. Alsaawy, "Digital Transformation of Public Sector Governance With IT Service Management–A Pilot Study," IEEE Access, vol. 11, no. January, pp. 6490–6512, 2023, doi: 10.1109/ACCESS.2023.3237550.

[3] T. Alyas, K. Ateeq, M. Alqahtani, S. Kukunuru, N. Tabassum, and R. Kamran, "Security Analysis for Virtual Machine Allocation in Cloud Computing," Internatinal Conferenceon Cyber Resilience, ICCR, 2022.

[4] T. Alyas. "Performance Framework for Virtual Machine Migration in Cloud Computing," Computer Materials and Continua., vol. 74, no. 3, pp. 6289–6305, 2023.

[5] T. Alyas, S. Ali, H. U. Khan, A. Samad, K. Alissa, and M. A. Saleem, "Container Performance and Vulnerability Management for Container Security Using Docker Engine," Security Communication Networks, vol. 20, 2022.

[6] M. Niazi, S. Abbas, A. Soliman, T. Alyas, S. Asif, and T. Faiz, "Vertical Pod Autoscaling in Kubernetes for Elastic Container Collaborative Framework," 2023.

[7] T. Alyas, A. Alzahrani, Y. Alsaawy, K. Alissa, Q. Abbas, and N. Tabassum, "Query Optimization Framework for Graph Database in Cloud Dew Environment," 2023.

[8] T. Alyas, "Multi-Cloud Integration Security Framework Using Honeypots," Mobile Information System, vol. 12. pp. 1-13, 2022.

[9] T. Alyas, N. Tabassum, M. Waseem Iqbal, A. S. Alshahrani, A. Alghamdi, and S. Khuram Shahzad, "Resource Based Automatic Calibration System (RBACS) Using Kubernetes Framework," Intelleligent Automation and Soft Computing, vol. 35, no. 1, pp. 1165–1179, 2023.

[10] G. Ahmed, "Recognition of Urdu Handwritten Alphabet Using Convolutional

Neural Network (CNN)," Computer Material Continua., vol. 73, no. 2, pp. 2967–2984, 2022.

[11] M. I. Sarwar, K. Nisar, and I. ud Din, "LTE-Advanced – Interference Management in OFDMA Based Cellular Network: An Overview", USJICT, vol. 4, no. 3, pp. 96-103, Oct. 2020.

[12] A. A. Nagra, T. Alyas, M. Hamid, N. Tabassum, and A. Ahmad, "Training a Feedforward Neural Network Using Hybrid Gravitational Search Algorithm with Dynamic Multiswarm Particle Swarm Optimization," Biomed Resource International, vol. 2022, pp. 1–10, 2022.

[13] T. Alyas, M. Hamid, K. Alissa, T. Faiz, N. Tabassum, and A. Ahmad, "Empirical Method for Thyroid Disease Classification Using a Machine Learning Approach," Biomed Resource International, vol. 2022, pp. 1–10, 2022.

[14] T. Alyas, K. Alissa, A. S. Mohammad, S. Asif, T. Faiz, and G. Ahmed, "Innovative Fungal Disease Diagnosis System Using Convolutional Neural Network," 2022.

[15] H. H. Naqvi, T. Alyas, N. Tabassum, U. Farooq, A. Namoun, and S. A. M. Naqvi, "Comparative Analysis: Intrusion Detection in Multi-Cloud Environment to Identify Way Forward," International Journal of Recent Trends in Engineering & Research, vol. 10, no. 3, pp. 2533-2539, 2021.

[16] S. A. M. Naqvi, T. Alyas, N. Tabassum, A. Namoun, and H. H. Naqvi, "Post Pandemic World and Challenges for E-Governance Framework," International Journal of Recent Trends in Engineering & Research, vol. 10, no. 3, pp. 2630-2636, 2021.

[17] W. Khalid, M. W. Iqbal, T. Alyas, N. Tabassum, N. Anwar, and M. A. Saleem, "Performance Optimization of network using load balancer Techniques," International Journal Advanced Trends Computer Science Engneering, vol. 10, no. 3, pp. 2645-2650, 2021.

[18] T. Alyas, I. Javed, A. Namoun, A. Tufail, S. Alshmrany, and N. Tabassum, "Live migration of virtual machines using a mamdani fuzzy inference system," Computer Materials Continua, vol. 71, no. 2, pp. 3019-3033, 2022.

[19] M. A. Saleem, M. Aamir, R. Ibrahim, N. Senan, and T. Alyas, "An Optimized Convolution Neural Network Architecture for Paddy Disease Classification," Computer Materials Continua, vol. 71, no. 2, pp. 6053-6067, 2022.

[20] J. Nazir, "Load Balancing Framework for Cross-Region Tasks in Cloud Computing," Computer Materials Continua, vol. 70, no. 1, pp. 1479-1490, 2022.

[21] N. Tabassum, T. Alyas, M. Hamid, M. Saleem, S. Malik, and S. Binish Zahra, "QoS Based Cloud Security Evaluation Using Neuro Fuzzy Model," Computer Materials Continua, vol. 70, no. 1, pp. 1127-1140, 2022.

[22] M. I. Sarwar, K. Nisar, and A. Khan, "Blockchain - From Cryptocurrency to Vertical Industries - A Deep Shift," in IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC), September 20-23, 2019, Dalian, China, 2019, pp. 537–540. doi: 10.1109/ICSPCC46631.2019.8960795.

Taseer *et al. (IJECI)* 2023

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

International Journal for
Electronic Crime Investigation

DOI: https://doi.org/10.54692/ijeci.2023.0704165

Research Article

Vol. 7 issue 4 Oct-Dec 2023

Malware and Windows APIs: A Dangerous Duo

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

Received: September 25, 2023; **Accepted:** November 18, 2023; **Published:** December 22, 2023

ABSTRACT

This paper introduces its interaction with malware and Windows APIs (application programming interface). The first section describes malware and investigates various types such as viruses, worms and trojans, as well as provides a brief history of malware and its evolution. The second section provides an overview of the Windows APIs. It shows how these interfaces allow software and operating systems to communicate with each other. It also highlights the most commonly used Windows APIs and their functions- The follow-up section explores how malware uses Windows APIs for malicious purposes، Explains the common methods used by malware to communicate with these interfaces- Includes real-world examples of malware attacks that use some Windows APIs. The study then turns its attention to the Windows API security mechanism, given the security measures taken by Windows to prevent the use of unauthorized APIs- The importance of user account control (UAC) and various monitoring and access control systems has been highlighted. The next section introduces the API Hoking and its application to malware. Which explains the strategies used by malware to hook Windows APIs- The effects of API hooking and possible detection methods are also discussed. This article provides an in-depth overview of real-world malware that exploits Windows APIs through case studies and analysis. Notable malware analyzes examples using family and API-based attacks- The article discusses security tools and ways to identify and block API-based malware, as well as how to design secure programs with Windows APIs Suggestions for this have also been discussed. Finally, malware tactics targeting Windows APIs discuss potential trends and issues, as well as expected API security challenges in the Windows context. This study continues to look at advances in Windows API security and their implications for malware prevention.

Keywords: Malware, Windows APIs, Virus, Insects, Trojan, Evolution, Security,

1. INTRODUCTION

Malware, an acronym for malicious software, is any software or code intended for computer systems, networks، or disrupt, damage, or gain unauthorized access to user devices. This refers to a wide variety of malicious programs and scripts that may jeopardize the integrity, privacy and availability of data and resources- Malware often works in secret, masked as legitimate software or exploits the weaknesses of the target system to

perform its harmful actions- Its targets range from stealing sensitive information and financial fraud to launching large-scale network attacks or exploiting affected systems for boot net activities May be- Effective malware detection, prevention‹ and mitigation is important for maintaining the security and privacy of computer systems and preventing potential damage caused by these destructive programs [1].

1.1 Types of malwares

Malware comes in many forms and poses various threats to computer systems and networks- Here are some examples of popular malware.

1.1.1. Viruses

Viruses are self-replicating programs that associate themselves with legitimate files or programs and infect other files or computers- They can damage data by corrupting or altering it, interfering with system functionality, and spreading it to other devices [2].

1.1.2. Worms

Worms are stand-alone programs that replicate and spread freely across networks, often exploiting security vulnerabilities- Unlike viruses, they do not need to be linked to existing files- Insects can use network bandwidth, subdue the system, and help spread other malware [2].

1.1.3. Trojans

Trojans often known as Trojan horses. There are misleading programs that hide themselves as legal software to deceive users into installing them- Trojans, once launched, can perform unauthorized operations such as stealing sensitive information, setting up backdoor for remote access‹ or releasing more malware [3].

1.1.4. Ransomware

There is a type of malware that encrypts or locks a victim's data or system, making them inaccessible to ransom payments- It seeks to divert money from victims by taking advantage of their desire to regain access to data or gadgets [3].

1.1.5. Spyware

Spyware is software that aims to secretly monitor and collect data on a user's activities without information or agreement- It can monitor strokes, take screenshots, record surfing dates, and collect personal or sensitive information, which is often exploited for harmful reasons [3].

1.1.6. Adware

There is a type of malware that displays unwanted ads on the user's device- This is short for ad-supported software- It is often included with free software downloads and for attackers by showing targeted ads or sending users to malicious websites Receives cash [3].

1.1.7. Botnet

Compromised computers or networks of devices are managed through a Centralized Command and Control (C&C) server- These compromising devices, called "bots" or "zombies", can be used to perform a variety of harmful acts‹ Including distributed Daniel of Service (DDoS) attacks, spam email campaigns, and malware distribution [4].

1.1.8. Rootkits

There are secret pieces of malware designed to gain privileged access and control over computer systems- They hide their presence by

editing system files, processes, or drivers, making it difficult to locate and uninstall them - Root kits are often used to have unauthorized access or to cover up additional infections [4].

2. WINDOWS APIs

Windows APIs (Application Programming Interface) provide a set of functions, protocols, and tools that enable developers to interact with the Windows operating system (OS)- These APIs serve as a bridge between applications and basic OS, allowing software to access system resources, services and functions- Here is an overview of Windows APIs [5].

2.1. Purpose

Windows APIs aim to present developers with a standard and documented interface for developing Windows programs- They summarize the complexities of the basic OS, allowing developers to focus on application logic rather than low-level system processes [5].

2.2. Functionality

File and Directory Operations, Process Management, Memory Management, User Interface Control, Network Connection, Device Input/ Output, Security & Verification, Registry Access, and many other features are available through the Windows APIs- These APIs expose many features, allowing developers to create complex and interactive apps [5].

2.3. Programming Languages

Windows APIs can be accessible through various programming languages, including C / C + +, C#, Visual Basic, and.NET- Microsoft provides software development kits (SDKs) and libraries that include the headers, libraries and documents needed to work with APIs [5].

2.4. API Sets

Depending on their functionality, Windows APIs are organized into sets or categories- Windows API (Win-API) for basic system functions, Windows Graphics API for graphics operations (WinGDI), Windows Networking API (Winsock) for network connection, and the Windows Multimedia API (WinMM) is all for multimedia related tasks- Examples of API sets [6].

2.5. Working of APIs

APIs (Application Programming Interface) serve as a bridge between software programs and basic operating systems (OS). They describe a set of protocols, functions, and data structures that the program can use to connect to the operating system and access its services and resources. Here's how APIs help facilitate this interaction [7].

i. APIs create a standard interface or agreement that explains how software components should interact with each other- They provide communication principles and protocols to ensure that applications can access OS functions in a consistent and predictable manner.

ii. APIs summarize the complexities of the basic operating system, preventing application developers from detailing the lower level of system operation- Instead of learning the intricacies of hardware and operating system internals Developers can rely on the API to handle these complexities and provide a simple interface for application development.

iii. Operating systems offer operations and services via APIs. Think of APIs as

helper tools for tasks. They can help with things like working with files, connecting to the internet, drawing pictures, or controlling user interfaces. They do this so the application using them doesn't have to start from zero.

iv. Data share: APIs make data sharing easier for operating system and software programs- Applications should use these data structures and formats to send or receive data from the OS Applications can use it to ask for services from the OS, to issue orders, to retrieve system data, or to receive notifications.

v. Access to System Resources: APIs provide users with access to services and system resources that are usually beyond the reach of applications- Examples of how APIs help interface applications with hardware include file system access, display output control, process management, this includes the use of network protocols, and the use of various OS-level features.

2.6. Windows APIs and their functions

Many Windows APIs (application programming interfaces) are available, each serving different purposes and providing access to different features of the system- Here are some commonly used Windows APIs and their functions [8].

i. Win32 API (Windows API): The Win32 API is a basic set of APIs that provide access to a wide range of functions and services for Windows applications- It covers areas such as window management, file system operations, process management, threading, networking, input/output, and user interface controls [9].

ii. Windows Graphics API (WinGDI): The WinGDI API offers functions for graphics and device-independent drawing operations- These applications create and manipulate graphical elements, create shapes, render text, handle fonts‹ Enables image processing and interaction with display devices.

iii. Windows Multimedia API (WinMM): The WinMM API provides services for multimedia-related tasks, including audio and video playback, recording, and processing- These applications run sound files, manage MIDI devices, capture audio and video stream, allows controlling multimedia devices and handling multimedia timers.

iv. Windows Networking API (Winsock): The Winsock API enables networking capabilities for Windows applications- Establishing network connections, sending and receiving data on TCP / IP and UDP / IP protocols, resolving host names, managing network configurations, and provides network services enforcement functions.

v. Windows Registry API: The Registry API allows applications to be read and written from the Windows registry, which stores system configuration settings and application-specific data- It provides functions for accessing registry keys, reading and writing values, creating or deleting keys, and managing registry security [10].

vi. Windows Management Instrumentation API (WMI): The WMI API enables applications to retrieve administrative information about Windows OS and perform system administration functions - It involves querying system features, managing processes, monitoring events, setting system settings, and offers a function of interacting with hardware components.

vii. Windows Shell API: The Shell API provides access to Windows Shell features, including file management, folder manipulation, user interface customization, and desktop integration- These applications include creating, copying, moving and deleting files, managing folders, manipulating icons, allows displaying system dialogs and interacting with Windows Explorer Shell.

viii. Windows Security API: The Windows Security API provides functionality for implementing security-related functionality in applications- This includes verification and authorization procedures, encryption services, access control management, secure communications and secure storage

ix. Windows COM and .NET APIs: Component Object Model (COM) and NET APIs provide a framework for developing component based and managed applications on Windows. They create and use COM items, access system services, and provides interfaces, libraries, and runtime environments for developing applications using the NET Framework.

These are just a few examples of commonly used Windows APIs and their functions- Windows provides a wide array of APIs tailored to the needs of different applications, allowing developers to take advantage of the power of the operating system and strengthen it, enables you to create feature-rich applications.

3. MALWARE TECHNIQUES AND WINDOWS APIs

Malware uses a variety of methods to take advantage of Windows APIs (Application Programming Interface) and perform harmful activities. Process injection is a popular method where malicious code is inserted into the legal process using APIs such as CreateRemoteThread, VirtualAllocEx, and WriteProcessMemo. Malware can hide its presence, avoid detection, and in doing so take control of the target machine. The Windows registry can also be changed via APIs such as RegOpenKey, RegSetValue, and RegCreateKey. Malware can establish persistence, change system settings, or run its code during system startup by modifying registry entries- Using APIs such as Create File, Read File, Write File, and Delete File, malware can also interact with the file system [11].

As a result, malware can convert or create files, encrypt data, hide its existence, or remove important system files to interfere with system operations- Additionally, malware interacts with external servers or other affected systems using networking APIs such as Winsock or WinINet- These APIs allow malware to spread across networks, transmit stolen data, and

receive orders from command-and-control servers- Malware can control system resources, avoid detection, and take advantage of Windows APIs to meet its harmful targets [12].

3.1. *How malware exploits Windows APIs for malicious purposes*

Windows APIs (Application Programming Interfaces) are often used by malware to perform harmful operations and to meet their goals- Below are some specific ways in which malware uses Windows APIs.

3.1.1. Code Injection

Malware can enter its malicious code into the normal process via APIs such as CreateRemoteThread, VirtualAllocEx, and WriteProcessMemory- By doing so, the virus can run its code within a reliable process, this makes it difficult to detect and possibly take precautionary measures [13].

3.1.2. Escalation of Privileges

Malware uses specific Windows APIs to increase its access rights and privileges- For example، Malware can change access to toxins and increase its privileges using APIs such as Open Process Token and Adjust Token Privileges to perform operations that Otherwise they will be forbidden.

3.1.3. File manipulation

To engage in a variety of malicious behaviors, the malware file system interacts with APIs such as Create File, Read File, Write File, and Delete Fil. To interfere with the regular operation of the system, malware can create or edit files, encrypt data، can change file properties to hide its existence, or delete important system files [14].

3.1.4. Registry Exploitation

Manipulates Windows registry by taking advantage of malware registry APIs such as RegOpenKey, RegSetValue, and RegCreateKey. It can establish stability, change system settings, run its malware at the beginning of the system, or disable security features by changing registry entries [15].

3.1.5. Network Communication

Uses malware networking APIs such as Winsock or WinINet to connect to remote servers or other infected systems- It spreads malware on networks, downloads more harmful payloads, enables you to communicate with command-and-control servers and send stolen data.

3.1.6. Techniques for Countering Analysis and Detection

Malware can exploit Windows APIs to develop countermeasures against analysis and detection- For example, to find virtualized environments or sandboxes, it can use APIs such GetTickCount and QueryPerformanceCounter- In addition, malware can interact with APIs such as EnumProcesses and EnumProcessModules to prevent detection through security software and anti-malware programs.

3.2. Common techniques used by malware to interact with Windows APIs

Malware uses a number of standard methods to communicate its destructive actions with Windows APIs (application programming interface). One such method is API hooking, where malware intercepts call into API functions and alters the behavior of those calls - Malware can track or modify information shared between apps and operating systems by

diverting execution to its code- Malware may use this method to steal sensitive data, change system behavior, or obtain security measures- As an alternative to static links to API functions, malware uses Dynamic API resolution, which solves API functions at runtime- This method enables malware to dynamically identify and call API methods, this helps malware avoid static analysis and detection through security tools- Malware can also change the input parameters provided to API calls to further its nefarious purposes. This technique is known as API parameter manipulation. This method can be used to get around security measures, take advantage of vulnerabilities, or perform unauthorized actions- In addition, malware may request specific APIs directly for malicious actions such as privilege enhancement, network communication, file manipulation, and registry alterations- These methods allow malware to interface with Windows APIs in order to undermine system security, steal confidential data, Self-expansion or interference in the regular operation of the system [16].

3.3. Malware attacks that leverage specific Windows APIs

There are numerous examples of malware attacks that take advantage of specific Windows APIs to perform their malicious activities- Here are some notable examples:

WannaCry (2017): WannaCry was a ransomware attack that took advantage of vulnerabilities in the Windows SMB (Server Message Block) protocol- Taking advantage of the Eternal-blue exploit, which targeted the Windows API "MS17-010", WannaCry spread rapidly across networks‹ Encrypting files and demanding ransom for their release [17].

Stuxnet (2010): Stuxnet was a sophisticated worm that specifically targeted the industrial control system- It exploited a number of Windows APIs, including Windows Management Instrumentation (WMI) and LSA (Local Security Authority) functions, including propaganda for Siemens SCADA systems‹ to compromise and disrupt Iran's nuclear program [18].

Emotet (2014-present): Emotate is a polymorphic malware that has evolved over time- It uses various Windows APIs, such as NetApi32, to spread across networks, steal sensitive information, and install additional malware on compromised systems- Emotate is known for its insect-like abilities and ability to avoid detection [19].

Zeus (Zbot) (2007-present): Zeus is a notorious banking Trojan that targets financial institutions- It benefits from Windows APIs, such as WinINet and CryptAPI, to steal banking credentials, conduct fraudulent transactions, and maintain consistency with affected systems- Zeus has been one of the most popular and influential malware families in the last decade [20].

NotPetya (2017): NotPetya was a devastating ransomware attack that hit the Windows system- It exploited the Windows API functions "OpenThreadToken" and "AdjustToken Privileges" to gain administrative access and late spread across networks- NotPetya has caused extensive damage to organizations around the world [21].

4. WINDOWS API SECURITY MECHANISMS

Windows includes a number of security techniques to maintain and maintain the integrity of your APIs (application programming interface). User Account Control (UAC), which debuted in Windows Vista and still exists in later editions, is an essential security feature- When apps try to perform privileged operations or change system settings, ask users for permission or agreement‹ UAC helps reduce the likelihood of unauthorized changes- UAC prevents unauthorized changes and minimizes the potential effects of harmful actions by requiring user consent to better access to APIs- Windows also uses Access Control List (ACLs) to control access rights and permissions to system resources- Administrators can set granular permissions using ACLs to indicate which individuals or groups can access specific APIs and which What operations can you perform- This technique ensures that only authorized entities can interact with sensitive APIs, at least helping to enforce the principle of privilege- In addition, Windows includes pre-existing safety tools such as Windows Firewall and Windows Defender Antivirus‹ which help defend against known malware and unauthorized network access, respectively- Together, these security measures help protect Windows APIs and maintain the overall security position of the operating system [22].

4.1. Security measures implemented by Windows to protect against malicious API usage

Windows implements a number of security measures to protect its APIs (application programming interface) from malicious use- These measures are aimed at ensuring the integrity, confidentiality and availability of system resources- Here are some key security measures implemented by Windows [23].

4.1.1. User Account Control (UAC)

User Account Control is a security feature introduced in Windows Vista and later versions. UAC helps prevent unauthorized changes to the system through the need for administrator approval or with the consent of the user when applications perform specific privileged operations Tries to access secure resources or modify system settings. UAC indicates permission before allowing users higher access to APIs, which reduces the risk of unauthorized changes to the system.

4.1.2. Access Control Lists (ACLs)

To define permissions to access different parts of system resource, Windows uses access control lists such as APIs. Administrators may set up ACLs so that sensitive APIs are out of bounds. Only authorized users or those from specific groups may approach them. This ensures that the least privileged principle is enforced as required and limits both damage from harmful APIs while leading to it being investigated if something does go wrong.

4.1.3. Code Signing and Digital Certificates

Windows needs a signing code in order to verify that drivers and other system-level components are authentic and not corrupted. Signing the code guarantees that APIs are only accessible through approved, validated software. Authorities issue digital certificates with a reliable certification, verifying the

source of the software and instilling confidence in users.

4.1.4. Windows Defender Antivirus

Windows adds a built-in anti-virus solution called Windows Defender Antivirus- It provides real-time protection against known malware threats, including those that may exploit Windows APIs- Windows Defender Antivirus regularly updates its virus definition database to detect and prevent malicious software that attempts to misuse APIs Coins.

4.1.5. Windows Firewall

Windows Firewall is a security function with the feature of a computer network which watches for and filters all entering or leaving network traffic. The Windows firewall protects against illicit entry to network resources and stops any suspicious action that might lead to a harmful use of the programs′ application program interface. Many applications are designed this way. The blocked item by Windows Firewall is shown (Win10 here: 192. How can you set what ports these rules apply to running allow or deny Network access based on rules and Policies to specific APIs using Windows Firewall?

4.1.6. Secure Development Practices

Microsoft promotes secure coding methods to developers through guides, tools and resources - By following the principles and best practices of secure coding, developers can write robust, secure applications that interact securely with Windows APIs.

These security measures implemented by Windows reduce the risk of using harmful APIs and maintain system resource security and overall operating system security help.

5. API HOOKING AND MALWARE

API Hunting is a method that changes the legitimate operation of APIs on the Goes operating system by installing software, and consequently viruses. There (in the context of malware), API hooks are often little more than an all-round means for obtaining and detecting forbidden actions. To provide an overview of API binding and compatibility with malware, look and see [24].

5.1. API Hooking

It is the approach whereby we disable API calls, and replace them with special lines of code or functions we ourselves have written. Using this technology, the code installed on a system can be modified at will. This makes it possible not only to rewrite and parameterize invisibly any existing program, but also to jump into results from caught API calls and examine what happens. In this book we see that API hooking a flexible way of linking up APIs to a worm. We hope that readers can use this knowledge to help them understand other articles on API hooking he has written.

5.2. Relevance to Malware

Malware exploits API hijacking for a variety of malicious purposes, including:

5.2.1. Stealth and Evasion

You can use the API to hide your presence by blocking API calls related to malware handling, file operations, network connections, or registry access- By handling intercepted API calls, malware can hide its files, processes, or network activity from security monitoring

tools and avoid detection.

5.2.2. Information Theft

Malware keyboard input, network traffic, or login credentials, credit card details, or hook file access APIs to get sensitive information such as classified documents- By blocking and editing API calls, malware can secretly steal data without the user's knowledge.

5.2.3. Code Execution and Persistence

Malware can use API hooking to insert harmful code into the legal process- By hacking APIs related to process creation or DLL loading, malware can insert its code into a trusted process‹ this ensures consistency and makes it difficult to detect and remove.

5.2.4. System Manipulation

Malware can hook APIs related to system settings, services, or security mechanisms to manipulate system behavior- By blocking and editing key API calls, malware can disable security features, edit system configurations‹ or can give yourself high privileges.

5.2.5. Detection and Countermeasures

At low level, malware hooks API and changes how it operates, which becomes difficult to detect. However, security instruments and techniques such as behavior-based analysis, anomaly detection and memory scanning can assist in identification of API hooking symptoms in malware.

In order to combat API hooking, security efforts are directed towards keeping code integrity therapy up, providing signatures for the API to modify, monitoring all call no matter where they go and even reverse hooks, etc. In addition, by keeping operating systems and security programs current with all the latest patches and updates you can help reduce the risks related to exploitation through API hooking.

5.3. Techniques used by malware to hook Windows APIs

Malware use IAT hooking prevent and modify the behavior of Windows APIs A malware uses to prevent and modify the behavior of Windows APIs is called Import Address Table (IAT) hooking. Import address table: A data structure containing the addresses of functions imported through the program from external attack. By editing the IAT, malware can send program calls to legitimate APIs on its malicious code. Malicious actions easy allow the malware to stunt financial news or Internet access for his end users. This lets malware block sensitive information, manipulate system behavior, or perform additional malicious actions. Malware usually inserts itself into the memory of the target process and changes the addresses in its IAT to point to its code rather than legitimate API functions [25]. IAT hooking can be used many different ways, such as by using inline hooks or by rebuilding the IAT. In the case of online hooking, the malware modifies the instructions at the front entrance of the target function to turn control over its code again. Reconstruction of the original Address Table (IAT) means replacing the true addresses in this table with the malware's own. In this way, the malicious software was able to manipulate and threaten a target's working procedures without being spotted.

To conceal your presence even more complete-

ly, writers of malware will use root kit technology, such as cutting changes in the Import Address Table (IAT). This includes the modification of data structures as shown in kernel mode data table (KDFT), a system service descriptor table, And so on. To ensure that hooks are not recognized by security software or system monitoring tools. Windows API calls that have successfully been hooked; use can be made to change the behavior of the various systems manipulated by malware but so aided in getting its sinister aims accomplished.

5.4. Consequences of API hooking by malware and potential detection methods

API hacking through malware can have serious consequences for system security and user privacy- When malware successfully hooks up Windows APIs, it has the ability to prevent, edit, and control the behavior of API calls- This can lead to many negative consequences. First, malware can use API hacking to gain unauthorized access to sensitive system resources, such as files, network connections, or user data. By blocking and manipulating API calls, Malware may ignore security measures and perform actions that compromise the privacy and integrity of the system- In addition, API Hoking enables malware to manipulate data exchanged between applications and operating systems, leading to data manipulation, corruption‹ Or unauthorized editing- This can have a serious impact on the reliability and reliability of the system- Furthermore, one of the main advantages of API hacking for malware is its ability to avoid detection- By blocking and editing API calls, malware can ignore security software that relies on API-based monitoring and analysis- This makes it difficult to detect and reduce the presence of malware- To address these risks, API hooking detection methods include behavior-based analysis, anomaly detection, memory scanning, and integrity testing- The purpose of these techniques is to identify abnormal API behavior and detect the presence of malicious hooks- Implement strong security measures, keep operating systems and security software up to date‹ and following secure coding methods can help reduce the risks associated with API hacking through malware.

6. MITIGATION STRATEGIES AND COUNTERMEASURES

Protecting Windows APIs against malware attacks involves implementing a set of best practices to enhance the overall security of the system- Below, I will outline some important recommendations without stealing any specific sources [26].

6.1. Best practices for securing Windows APIs against malware attacks

6.1.1. Regularly Update Windows

It is important to keep the Windows operating system up to date with the latest security patches. Microsoft often releases updates to address vulnerabilities and improve overall system security. So, you should enable automatic updates, or check regular updates manually.

6.1.2. Use Robust Authentication and Authorization Techniques

When obtaining Windows APIs, the authentication is robust and access privileges carefully designed. Only entities authorized by secure communication protocols such as Transport Layer SSL (TLS) can access sensitive APIs.

A key part of secure coding practice is to make

sure that applications that access Windows APIs incorporate the right ones. In addition, you should make sure you use secure programming languages to pass all inputs through some type of filter, also you need to check that all input is correct; Put in place strict input/output data validation to avoid common security problems caused by errors when entering queries for an SQL-database into programs that lead one directly into memory overwriting it from this point and so forth.

6.1.3. Install Anti-Virus Software

When operating in the Windows system, this means you must make sure your machine is being regularly visited by well-known antivirus software with current updates every day. This will reduce the number of viruses you catch significantly and even when known bugs are not yet known to have escaped from their underground environments new threats such as viruses or worms will be thwarted by these systems.

6.1.4. Implement Runtime Protection Mechanisms

At the same time, you need to address runtime protection mechanisms such as Data Execution Prevention (DEP) and Address Space Layout Randomization (ASLR). DEP prevents enforcement of malicious code from areas that are not suited for memory, While ASLR randomizes memory layouts so that attackers will not be able to find any given function or data.

6.2. Security tools and techniques for detecting and preventing API-based malware

In order to eliminate API-based malware, it is necessary to rely on a carefully selected combination of defensive equipment's and means which can accurately determine and then remove possible dangers.

6.2.1. API Monitoring

Deploy tools that oversee any and all API calls from your system. They are able to calculate the amount of transaction and analysis of API traffic abnormalities which may indicate malware, calling out racially discriminatory activity in real time if necessary. Also keep alert of any future API calls that seem potentially suspicious. Or to stop fake APIs before they can take effect [27].

6.2.2. Web Application Firewalls (WAF)

Off to an excellent start, now how about WAF for your API endpoints. Your WAF will also help you defend against web-based vulnerabilities like SQL injections as well botnet attacks. This type of security inspects each API request, its purpose is to eliminate dangerous data and ultimately protect everything from hacking attacks [28].

6.2.3. Behavior-based Detection

Use behavior-based detection tools that analyze the behavior of API calls and endpoints to identify patterns associated with malware activity- These tools can detect irregularities, such as excessive API calls [29].

6.2.4. API Security Gateways

Create API security gateways that act as intermediaries between clients and API end points. These gateways are able to enforce security policies, verify and allow API requests and inspect incoming and outgoing API traffic for threats.

6.2.5. Threat Intelligence Services

Subscribe to threat intelligence services that give real-time information on known malware signatures, attack indicators (IOCs) and emerging threats. These services can help protect your ability to discover and prevent API-based malware attacks by just its very existence, leveraging the combined wisdom and expertise of today's most up-to-date security community.

6.2.6. Sandboxing and Isolation

Use sandboxing technology in a virtual environment to run potentially suspicious or unknown calls against the API as a controlled event. Sandboxes are detached from production systems, so you can watch and analyze how API calls behave without jeopardizing overall system security.

6.2.7. Machine Learning and AI-based Analysis

Use machine learning and AI algorithms to analyze API traffic patterns and identify potential malware activity. These algorithms can learn from historic data, detect deviations in normal behavior and increase their detection accuracy with time.

6.2.8. Threat Hunting and Incident Response

Establish a strong risk and incident response program to actively search for signs of API-based malware attacks Logs, network traffic، and actively investigate system behavior to catch potential hazards quickly [30].

6.2.9. Vulnerability Scanning and Penetration Testing

Regular vulnerability scans and penetration testing to find out where your API infrastructure might be leaky. These inspections correct weaknesses that can be exploited through malware and they show a weak spot in advance of an attack.

6.2.10. Security Awareness Training

Inform developers, system administrators, and users of the dangers of malware to which APIs are vulnerable. Provide training in secure coding practices, API best practices, and the importance of abiding by recommendations for security to avoid malware infections.

Remember, keep up to date with these tools and techniques, patch your systems regularly, and in order to effectively detect and prevent API-based malware attacks, it is important to adapt your security measures to the emerging threat scenario.

6.3. Recommendations for developers to write secure code using Windows APIs

When it comes to writing secure code using Windows APIs, Developers should follow a set of recommendations to enhance the overall security of their applications- First of all, it is important to understand the documentation and guidelines provided by Microsoft for each API - Developers must strictly adhere to safe coding methods, such as verifying and cleaning user input, to prevent common hazards such as buffer overflow and injection attacks. It is important to implement appropriate procedures for dealing with errors to avoid information leaks and possible exploitation. In addition, developers should apply the principle of minimum privilege, only give necessary permissions to APIs and restrict access to sensitive resources- Regular updating and patching of Windows operating systems and APIs is essential to eliminate any known

vulnerabilities. Finally, the code base should be constantly tested and the code reviewed to identify and address any security vulnerabilities or vulnerabilities- By following these recommendations, developers can greatly increase the security of their applications that rely on Windows APIs [31].

7. FUTURE TRENDS AND CHALLENGES

7.1. Emerging trends in malware techniques targeting Windows APIs

New trends are constantly emerging for targeting Windows API with latest malware techniques. These trends in recent years show that security well-deserved measures are faced with harassment all the time and it need effective measures must be taken to adapt to these evolving threats. Trending now is the addition of fileless malware, which continues to grow in popularity among attackers due to its ability to evade traditional anti-virus solutions. These kinds of malware work in the computer memory only, using legitimate Windows APIs to perform malicious code without leaving behind traces on the disk. Obviously of this ridiculous nature is it increasingly difficult to recognize and fend off fileless malware.

Living from the Land-type attacks also came into vogue. Attackers have started to utilize Windows utilities and built-in functions that are reliable such as these are PowerShell, WMI, or WSH to carry out evil deeds. By using these software applications, they can implant viruses while preventing conventional safety measures from working. Techniques - including API hacking and DLL injection, allow malware to rearrange the behavior of a legitimate application or to stop API calls being made. After that, it was anyone's guess how the game would go. This illegal access lets perpetrators adjust the data, authorize escalating privileges, or acquire unauthorized control of that system.

Bypass is another trick used by malicious actors. It involves making a legal procedure and then changing its code to reflect malicious content. This way, even if malware is discovered, it won't be recognized as such when it seeps out into the system as a legitimate procedure. Attackers actively look for opportunities to exploit vulnerabilities in Windows APIs in order to gain unauthorized access or force arbitrary code onto the system. They find weaknesses in API implementation and strike at zero-day bugs plus arbitrary systems [32].

To this end, malware such as this authenticator one abuse legitimate APIs to keep the persons in the compromised systems; meanwhile they hid themselves and went for victims. They manage API calls, using obfuscation technology and employee's anti-analysis methods to make it difficult for security solutions to probe their malicious activities. Supply chain attacks have become a favorite for attackers who aim to insert malware into trusted applications and libraries containing Windows API calls. By compromising the software supply chain, attackers may be able to distribute malware to multitudes of users and therefore gain widespread access to targeted systems. Malware authors often uses polymorphic and encrypted techniques in order to escape signature-based detection. By changing code structures frequently, or using new encryption methods they make it increasingly difficult for conven-

tional anti-virus solution to effectively identify and analyze the virus.

In order to provide a stable system platform, malware commonly targets Windows APIs concerning file and registry manipulation. Malware might edit critical files or keys in remote servers, so that it continues to function even if the system is restarted or checked for security problems after coming back online. However, the direction of ransomware attacks using Windows APIs has also tended in a more sophisticated way. As attackers find vulnerabilities or insecure APIs to access and encrypt your data leading them demanding to take ransom for restoring it again. This has a growing bearing on developers and security professionals. Ongoing efforts to observe systems, combining this with behavior analysis, and the use of advanced risk detection have all become essential necessary tactics for combatting malware techniques. Security of applications that depend on Windows APIs can be improved, but only if we are already proactive in dealing with these challenges [33].

7.2. Potential future challenges for API security in Windows environments

In the future, API security in the Windows environment could face many challenges. One potential challenge is the increasing complexity and diversity of APIs as technology develops. As APIs become more complex and interconnected, it becomes more difficult to ensure their safety. Implement strong authentication and authorization procedures to keep developers updated with the latest security best practices and to protect them from unauthorized access and data breaches Will need to- Furthermore, with the proliferation of Internet of Things (IoT) devices and their integration with the Windows environment, securing APIs becomes even more important- The sheer number of interconnected devices and the potential for vulnerabilities in their APIs pose significant security risks, which are severely tested‹ Weaknesses need to be addressed through assessments and constant monitoring- As APIs continue to play an important role in facilitating seamless communication and integration‹ Organizations must be proactive in adopting their own security measures to reduce emerging threats and ensure the integrity and privacy of their Windows API environment [33].

8. CONCLUSION

Finally, the combination of malware and Windows APIs offers a powerful and dangerous pairing in the realm of cybersecurity- Malware continues to evolve, using state-of-the-art techniques to take advantage of vulnerabilities in Windows APIs, compromising systems, stealing sensitive data, and individuals, organizations‹ And even significant damage to critical infrastructure- The inherent strength and capability of Windows APIs, while essential for enabling smooth integration and functionality‹ It also provides opportunities for attackers to take advantage of these APIs for harmful purposes- It is important for researchers, developers, and security professionals to understand the emerging scenario of malware and Windows APIs, to implement strong security measures‹ and be vigilant and dynamic in constantly updating and patching systems to reduce risks- Furthermore, knowledge sharing, identifying emerging threats‹ and cooperation between industry,

academies, and government agencies is essential to developing innovative solutions to protect the Windows environment from the ever-present threat of malware- By recognizing the dangerous pair of malware and Windows APIs and implementing comprehensive security strategies، we can strive for a secure digital ecosystem that protects consumers and their valuable information.

REFERENCES

[1] Ucci, L. Aniello, and R. Baldoni, "Survey of machine learning techniques for malware analysis," *Computers & Security*, vol. 81, pp. 123-147, 2019.

[2] N. Pachhala, S. Jothilakshmi, and B. P. Battula, "A Comprehensive Survey on Identification of Malware Types and Malware Classification Using Machine Learning Techniques," *IEEE Xplore*, 2021.

[3] Tahir, "A Study on Malware and Malware Detection Techniques," *International Journal of Education and Management Engineering*, vol. 8, no. 2, pp. 20-30, 2018.

[4] S. Subrahmanian, M. Ovelgönne, Tudor Dumitras, and B. S. Prakash, "Types of Malware and Malware Distribution Strategies," 2015,

[5] Gupta, H. Sharma, and S. Kaur, "Malware Characterization Using Windows API Call Sequences," pp. 271-280, 2018,

[6] Rabadi and S. G. Teo, "Advanced Windows Methods on Malware Detection and Classification," *Annual Computer Security Applications Conference*, 2020,

[7] P. Robillard, "What Makes APIs Hard to Learn? Answers from Developers," IEEE Software, vol. 26, no. 6, pp. 27-34, 2009.

[8] Klamt and A. von Kamp, "An application programming interface for CellNetAnalyzer," Biosystems, vol. 105, no. 2, pp. 162-168, 2011.

[9] P. Shelton, P. Koopman, and K. Devale, "Robustness testing of the Microsoft Win32 API," *IEEE Xplore*, 2023.

[10] M. Ijaz, M. H. Durad, and M. Ismail, "Static and Dynamic Malware Analysis Using Machine Learning," *IEEE Xplore*, 2019.

[11] Idika and A. Mathur, "A Survey of Malware Detection Techniques," 2007.

[12] T. Alsmadi and N. Alqudah, "A Survey on malware detection techniques," 2021 International Conference on Information Technology (ICIT), 2021.

[13] Ray and J. Ligatti, "Defining code-injection attacks," ACM SIGPLAN Notices, vol. 47, no. 1, p. 179, 2012.

[14] L. Castro, C. Schmitt, and G. D. Rodosek, "ARMED: How Automatic Malware Modifications Can Evade Static Detection," *IEEE Xplore,* 2019.

[15] Varlioglu, N. Elsayed, Z. ElSayed, and

M. Ozer, "The Dangerous Combo: Fileless Malware and Cryptojacking," *IEEE Xplore*, 2022.

[16] Mamoun Alazab, S. Venkataraman, and P. A. Watters, "Towards Understanding Malware Behaviour by the Extraction of API Calls," 2010,

[17] Mohurle and M. Patil, "A brief study of Wannacry Threat: Ransomware Attack 2017," International Journal of Advanced Research in Computer Science, vol. 8, no. 5, 2017.

[18] Baezner and P. Robin, "Stuxnet," www.research-collection.ethz.ch, 2017.

[19] Sophos Labs Research Team, "Emotet exposed: looking inside highly destructive malware," Network Security, vol. 2019, no. 6, pp. 6-11, 2019.

[20] Mohaisen and O. Alrawi, "Unveiling Zeus," Proceedings of the 22nd International Conference on World Wide Web, 2013.

[21] Y. A. Fayi, "What Petya/NotPetya Ransomware Is and What Its Remidiations Are," Advances in Intelligent Systems and Computing, pp. 93-100, 2018.

[22] Akinbi, E. Pereira, and C. Beaumont, "Evaluating security mechanisms implemented on public Platform-as-a-Service cloud environments case study: Windows Azure," 8th International Conference for Internet Technology and Secured Transactions (ICITST-2013), 2013.

[23] Ki, E. Kim, and H. K. Kim, "A Novel Approach to Detect Malware Based on API Call Sequence Analysis," International Journal of Distributed Sensor Networks, vol. 11, no. 6, p. 659-660. 2015.

[24] S. Z. Mohd Shaid and M. A. Maarof, "In memory detection of Windows API call hooking technique," *IEEE Xplore,* 2015.

[25] Y. C. Cheng, T.-S. Tsai, and C.-S. Yang, "An information retrieval approach for malware classification based on Windows API calls," *IEEE Xplore*, 2013.

[26] Xiao, C. Zhu, J. Xie, Y. Zhou, X. Zhu, and W. Zhang, "Dynamic Defense Strategy against Stealth Malware Propagation in Cyber-Physical Systems," IEEE Xplore, 2018.

[27] C. D. Elia, S. Nicchi, M. Mariani, M. Marini, and F. Palmaro, "Designing Robust API Monitoring Solutions," *IEEE Transactions on Dependable and Secure Computing*, pp. 1-6, 2021.

[28] V. Clincy and H. Shahriar, "Web Application Firewall: Network Security Models and Configuration," *2018 IEEE 42nd Annual Computer Software and Applications Conference* (COMPSAC), 2018.

[29] S. Galal, Y. B. Mahdy, and M. A. Atiea, "Behavior-based features model for malware detection," *Journal of Computer Virology and Hacking Techniques*, vol. 12, no. 2, pp. 59-67, 2015.

[30] Thompson, "Threat Hunting," pp. 205-212, 2020.

[31] Peter Leo Gorski, Y. Acar, Luigi Lo Iacono, and S. Fahl, "Listen to Developers! A Participatory Design Study on Security Warnings for Cryptographic APIs," 2020.

[32] Mamoun Alazab, S. Venkataraman, and P. A. Watters, "Towards Understanding Malware Behaviour by the Extraction of API Calls," 2010.

[33] A. Adamov and A. Carlsson, "The state of ransomware. Trends and mitigation techniques," *2017 IEEE East-West Design & Test Symposium (EWDTS*), 2017.

Asif *et al. (IJECI)* 2023

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

International Journal for
Electronic Crime Investigation

DOI: https://doi.org/10.54692/ijeci.2023.0704166

Research Article

Vol. 7 issue 4 Oct-Dec 2023

Incorporating the Future: Optimizing Cybersecurity through Seamless Integration of Artificial Intelligence

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

[1]Department of Mathematics, The University of Lahore, Lahore.

[2] Institute of Quality and Technology Management, University of the Punjab, Lahore, Pakistan.

Corresponding author: khuramshah6515@gmail.com

Received: September 27, 2023; **Accepted:** November 21, 2023; **Published:** December 22, 2023

ABSTRACT

Cyber-attacks are becoming more sophisticated and common in today's environment. Artificial intelligence (AI) is being used by enterprises to boost their defenses against these developing threats. AI is rapidly altering the cybersecurity field, providing several benefits in terms of improving security measures. However, its implementation causes significant changes in cybersecurity occupations and necessitates the acquisition of new skills by specialists. This article investigates the impact of AI on cybersecurity employment, presents real-world instances of AI integration in the sector, analyzes the future of AI in cybersecurity, and identifies the problems involved with its adoption.

Keywords: Artificial Intelligence, Cyberattack, Cybersecurity, employment, skills.

1 INTRODUCTION

The Development of Personnel in Changeovers One of the major consequences of AI on cybersecurity is the growth of cybersecurity employees. By 2022, 40% of cybersecurity professionals, according to Gartner, will be using AI in adding to their job, contributing 100% to the ground's work force. Automation of monotonous duties like data processing, monitoring, and risk assessment is rendered possible by AI. Professionals are able to focus on more complex duties like developing and carrying out new security rules, resolving incidents, and risk management because AI relieves them of these monotonous activities [1]. Improving Job Classifications: As AI becomes integrated into cybersecurity, the cybersecurity workforce's descriptions of employment and requirements for skills are evolving. Personnel with knowledge of data analytics, machine learning, and other AI-related abilities are in higher demand among enterprises. The World Economic Forum states that cybersecurity professionals with experience in artificial intelligence, data analytics,

and big data are in considerable demand. The study demonstrates that artificial intelligence and machine learning are going to play an essential role in the future for cybersecurity professionals. Professionals must enhance these AI-related skills and evolve as AI keeps influencing the landscape of cybersecurity in order to be effective in the field [2].

Illustrations of AI in cybersecurity in the real world to enhance their efficiency, a number of cybersecurity enterprises have begun to embrace AI and implement it into the products and services they offer. Here are two notable instances: Palo Alto Networks and Cortex XDR are two companies that collaborate. Major cybersecurity firm Palo Alto Networks automates cyber threat identification and response with artificial intelligence. Their solution, Cortex XDR, utilizes machine learning algorithms to analyze data from various places and find inconsistencies that can point to a security breach. Cortex XDR enables security teams to respond to threats more quickly and efficiently by automating responding to incidents. Cylance security for endpoints and CylanceProtect Cylance utilize es artificial intelligence in order to recognize and prevent attacks online. Their AI-powered initiative, CylanceProtect, utilizes machine learning techniques to evaluate files and find malicious code before they execute. The system additionally utilizes behavioral evaluation to detect unusual behaviors while taking appropriate action. The eradication of new risks has been demonstrated to be extremely effective with this proactive approach to protecting against threats. These practical illustrations demonstrate how AI could significantly improve cybersecurity operations and protect corporations from a variety of threats [3].

2. THE FUTURE OF AI IN CYBERSECURITY JOBS

It is projected that AI is going to continue to play a greater part in cybersecurity. Markets & Markets anticipates that the global market for artificial intelligence in cybersecurity will grow at an exponential yearly rate of growth, approaching 23.3%, from $8.8 billion in 2019 to $38.2 billion in 2026. The expanding application of AI in threat identification, mitigation, and responses is what is generating this peak. AI utilization in cybersecurity comes with challenges; however, A lack of AI-savvy cybersecurity specialists is one of the main challenges. The rapid growth of AI technology has resulted in shortages since it has displaced the marketplace's expertise and understanding of development. Thus, in order to remain competitive in the industry, cybersecurity professionals must acquire additional expertise and learn AI-related skills [2].

3. AI A GAME-CHANGER IN THE FIELD OF CYBERSECURITY

AI has the capability to transform cybersecurity by advancing its efficacy and effectiveness. One of the most significant effects of AI on cybersecurity is the recognition of threats. AI systems have the ability to sort through huge quantities of data and detect trends steering to potential threats. In addition, by determining

and repairing vulnerabilities before they could be manipulated, AI is able to assist with managing vulnerabilities. AI may additionally help with behavioral exploration, which comprises analyzing user behaviour to identify unusual patterns. This may help to identify threats from inside and prevent worker-triggered breaches of information. Accordingly, automation for security could use artificial intelligence (AI), which could assist automated repetitive tasks and free up cybersecurity professionals to concentrate on more challenging tasks. Cybersecurity assistances tremendously through AI, and this impact is only going to expand in the years to come. To preserve the security and safety of the modern technological environment, investment in artificial intelligence (AI) and cybersecurity is imperative. As the world evolves more and more digitally, it is of the highest importance to use cutting-edge innovations like artificial intelligence (AI) to remain ahead of the curve and to be vigilant when it concerns cyber threats [2].

4. AI REVOLUTION IN CYBERSECURITY

Enormous amounts of information can potentially be managed in real time by artificial intelligence (AI) threat detection systems, which may detect potential dangers before they cause an impact. Comparing with conventional, signature-based on signatures antivirus programs, these systems are more effective because they apply machine learning algorithms that obtain information from previous attacks and respond to new attacks. Another field where artificial intelligence has had substantial effects is handling vulnerabilities. Security professionals are better competent to sense vulnerabilities in the network earlier attackers take benefit of them. by using AI-powered vulnerability indicators that can detect and classify network vulnerabilities. Another field where artificial intelligence could be applied to improve cybersecurity is behavioral analysis. Behavioral analytics programs powered by AI have the capability to keep track of user behaviors and detect variations that might indicate a cyberattack. This can speed up the method by which security professionals identify and tackle cyberattacks And last, a key field where AI may assist cybersecurity is automating procedures. Cybersecurity professionals may concentrate on challenging problems by automating everyday tasks like patching, upgrading, and monitoring. However, partner confidence is necessary for the effective implementation of AI solutions for cybersecurity. Establishing trust among partners is a necessity for transmitting information in an appropriate way and combating cyber threats. Because of the exponential growth of data and the increasing sophistication of cyber threats, traditional cybersecurity solutions are no longer suitable to defend our electronic environment. AI-based cybersecurity tools have become essential in detecting and preventing cyber-attacks. AI provides considerable advantages to firms who use it into their defense operations. Given human limits, it is impossible to discover new malware variants, phishing methods, and every single threat encountered by a company and its cloud-based services. Furthermore, evaluating the possibili-

ty of a threat is considerably more difficult due to the intensity and vulnerabilities it may draw to a server. In reaction to a danger, an unknown, undetected threat may inflict tremendous harm to a system [4].

5. BENEFITS OF USING AI IN CYBER SECURITY

AI and ML go hand in hand in advanced cyber security. They eliminate time consuming tasks done manually by human experts. AI and ML are assigned the duty to scan a vast amount of data to identify potential threats and minimize false positives. This lets the human experts to focus on more critical threats. AI never stops learning It analyzes network activity using machine learning modules and deep learning algorithms. It will detect deviations or security issues from the typical course of events. This enables for immediate action and improves future security measures by preventing possible threats with similar behavioral features from entering the system. Because AI is always learning, it is difficult for hackers to outwit its intellect. Similar patterns on the network are detected by AI, and when they are recognized, the AI technology will cluster them together and then proceed to determine whether there were any deviations or if any security issue happened in the usual traffic. It ultimately reacts to them after evaluating the traffic. AI will detect undiscovered risks, but detecting all possible threats to a corporation might be daunting owing to hackers' ever-changing techniques. This makes it critical to implement current solutions, such as AI technology, to efficiently identify and prevent unknown dangers, which may cause significant damage if they go undiscovered [5].

AI will be used Managing massive volumes of data within a company's network, resulting in massive amounts of network and system traffic. It takes time for cybersecurity engineers to carefully evaluate all activity for possible risks. AI technology will automatically scan and identify any disguised threats, speeding up the detection process and improving overall system security. AI improves vulnerability management. Given the regular assaults and dangers that various firms encounter, it is critical in addressing network vulnerabilities. It will review existing security procedures to identify the weakest security links, allowing these organizations to concentrate on important security responsibilities. It enhances problem-solving skills and safeguards firm systems faster than cybersecurity engineers [4].

AI is capable of enhancing overall security when hackers and different kinds of threat actors constantly change their attack tactics, making it tough for cybersecurity engineers to prioritize security related tasks. Even when dealing with many threats at the same time, AI is highly useful in recognizing all forms of assaults and prioritizing protection. Human mistake and neglect can also pose security concerns, but AI's self-learning skills can equip it to deal with them. AI technology reduces data redundancy to great extent, it can do numerous activities at once and repeat critical security duties that can weary cybersecurity workers. It will perform frequent detection and

prevention of fundamental security threats, as well as extensive analysis to discover potential security breaches and vulnerabilities. AI empowers enterprises to maintain network security by employing best practices that are regularly applied without the danger of human mistake or boredom [6].

AI can respond quickly and complete the detection phase quicker, when AI technology is used with security software, risks may be detected and responded to quickly. It will prevent permanent damage to enterprises and corporations. When compared to humans, AI can scan whole networks and security systems to spot dangers sooner and simplify security chores. It is Easier to perform Authentication while using AI enabled technologies. There are Websites that have user account features and contact forms containing user credentials and other kinds of sensitive information which is requires as an additional security layer of security for protection. This security technique may be provided by AI by utilizing various technologies such as facial recognition, CAPTCHA, and fingerprint scanners to ensure authentication during routine login attempts. It will detect fraudulent login attempts and prevent credentials from being stolen or stuffed. It is capable of detecting brute force attacks. These brute force attacks could lead to a potential security breach from company network [7].

Machine learning and AI reduces the processing time of threats and system vulnerabilities. AI is the most used critical technology in cybersecurity. It shortens the processing time of several time-consuming jobs that are performed more slowly by human specialists. It then searches massive amounts of data to identify potential dangers. After that, it filters out non-threatening activity to prevent false positives. So that human specialists may devote their attention to more vital security responsibilities. The task of detecting and eliminating bots is much easier on AI enables systems. These Bots are still an emerging threat in cybersecurity. But they are still deadly and dangerous. They can lay havoc on networks and systems through DOS or DDOS attacks. Bots are responsible for malicious activities like spreading malware and stealing data. AI can detect and stop these bots based on their behavioral patterns by producing more secure captchas. They are detected and their mode of operation is kept in the system. AI security software will use several honeypots to trap and destroy them [3].

6. UNVEILING THE DISADVANTAGES OF AI IN CYBER SECURITY

AI has made significant advances in a range of fields, including cyber security. Its ability to rapidly analyze enormous amounts of data and detect patterns has reignited enthusiasm in the battle against digital threats. AI in cyber security nevertheless comes with drawbacks comparable to any other type of advancement in technology. Vulnerability to Adversary Attacks: The susceptible nature of AI to adversarial attacks is an important concern in the field of cyber security. Consistently providing malicious material to AI systems with the

objective of misleading or fooling the algorithms is an adversary assault. Such assaults have the ability to deliver biased or erroneous results by taking advantage of deficiencies in AI systems. These vulnerabilities enable cybercriminals to cross through AI -based safety procedures to achieve unauthorized exploitation of networks. Attacks such as those demonstrate how important it is to constantly monitor, update, and upgrade artificial intelligence systems with the aim of reducing the risks associated with adversarial attacks [4].

6.1. False Positives and False Negatives

Even though artificial intelligence algorithms aren't error-free, processes related to cyber security can confront errors such as false positives and false negatives. False positives are the consequence of artificial intelligence (AI) systems inadequately identifying risk-free behavior as harmful, resulting in redundant warnings or interferences. False negatives occur when artificial intelligence systems lack the capacity to recognize actual risks, which leaves illegal activities undiscovered. Such errors might cause pressure on security capabilities, leading to vulnerabilities. To reduce the number of false positives and false negatives, artificial intelligence models need to be improved and validated constantly; however, establishing an appropriate equilibrium can be complex [8].

6.2 . Lack of Human Oversight

AI has the ability to enhance human abilities in cyber security; yet, depending just on AI systems without oversight from humans could be dangerous. While AI is intended to streamline and automate operations, it is not nearly as capable of thinking clearly as human experts and doesn't possess comparable knowledge of the context. Therefore, AI systems might make inappropriate decisions on the basis of imbalanced or inadequate data, which might result in safety breaches. Examining AI outcomes, detecting inconsistencies, and reaching sensible conclusions are all made accessible by human experts. Operative cyber security is contingent upon determining the right balance between human knowledge and artificial intelligence (AI) mechanization [9].

6.3. Ethical and Privacy Concerns

Enormous amounts of data, particularly sensitive and personal data, are necessary for AI systems in cyber security to determine correlations and detect vulnerabilities. Considerations about ethics and privacy have been brought up by this dependence, specifically if the information is inappropriately archived. Privacy laws may be breached by unapproved access, breaches of data, or exploitation of personal data. It is necessary for maintaining effective information governance, encryption, and adhering to privacy principles in order to moderate these risks and keep public confidence in AI -driven cyber security resolutions [5].

6.4 . Evolving Threat Landscape

A question confronting artificial intelligence-based cyber security technologies is the continuously evolving view of cyberattacks. Even though artificial intelligence algorithms are trained on past data and patterns, newer

procedures for attack or emerging risks may be recognized for the very first time. AI systems could therefore be powerless to identify and responding to threats that weren't previously unexplored. In order to remain up to date with new threats, AI models need to be constantly evaluated, updated, and trained. AI and human capability, along with current threat information, may enhance cyber threat recognition and mitigation. AI is not a comprehensive examination response, even if it could considerably enhance cyber security expertise. Adversary assaults, false positives and negatives, a non-existence of human control, moral concerns, and an evolving threat environment constitute some of the difficulties encountered by artificial intelligence-powered cyber security systems. More research, collaboration, and an extensive plan that carries concurrently the expertise of artificial intelligence with human expertise will be necessary to overcome these obstacles. By resolving these encounters, we can use the benefits of artificial intelligence while determining vigorous and reliable cyber security measures [7].

7. CHALLENGES WHILE IMPLEMENTING AI

The use of AI-based elucidations in the area of cyber security continues to develop, along with the increasing utilization of AI in several other domains. AI can be used to identify, stop, and react to cyberattacks. It has been shown to be an effective instrument to strengthen an organization's overall security architecture. However, while integrating artificial intelligence into cyber security, several problems need to be taken into consideration. These challenges contain a variety of challenges, such as human capability, moral encounters, and technological restrictions. In this section, we will look at the challenges accompanying with using AI for cyber security and offer sustainable solutions. Data Quality: preceding to ever contemplating incorporating artificial intelligence into cybersecurity research, the quality of data is a further problem that has to be addressed. Data quality is an additional problem that has to be addressed prior to contemplating using artificial intelligence in cybersecurity research. When collecting and analyzing data, many different kinds of errors could occur, including inaccurate information or sources that are skewed [8].

7.1 . Lack of Transparency in AI System

Although being observed as a game-changer, AI in cybersecurity does not come outside its drawbacks. The lack of transparency in AI systems is one of the remarkable concerns. This is because AI systems can intermittently be thought of as "black boxes," which employ extensive mathematical computational models that are difficult for humans to understand or analyze. Due to this, it might be challenging for human experts to fully recognize the conclusions generated by these algorithms and to implement the necessary variations to improve their performance. There are several major implications for cybersecurity from the lack of transparency. unobserved threats to security could go undetected, and false positives might result in the system sounding

unnecessary warnings or cautions [9].

In additive, moral issues specifically predispositions and discrimination—are addressed by the transparency of AI algorithms. Without human operatives acknowledging it, AI systems may make racist or discriminatory choices. This could have significant concerns, mainly in law enforcement or employment operations. Researchers are examining into new methods for "explicable “AI” with the objective of addressing this task and making it possible for humans to understand the verdict-making methods of these AI systems. Among the techniques used are conversational explanations, visualization, and definitions of rules that can be easily realized by humans and that regulate the AI computer's decision-making procedure. These attempts assist individuals to identify any deficiencies as well as improve their abilities by providing them with a greater identification of how AI systems function [10].

7.2 . Adversarial Attacks in AI Cybersecurity

AI cyberattacks, and countermeasures have been fascinating a lot of interest in cybersecurity research over the past few years. Many research analyses have investigated the use of AI to identify and combat adversarial aggression, but various challenges must yet be tackled before these systems can be extensively implemented. The potential of artificial intelligence-based cybersecurity attacks and defenses has been accentuated by recent research. A survey by MIT Technology Review Insights and Darktrace of more than 300 C-level executives, directors, and managers exhibited that the majority of them think artificial intelligence will become important to cybersecurity in the near future [11].

The extensive implementation of these systems remains to combat numerous difficulties, including a few studies addressing the use of artificial intelligence (AI) to identify and react to adversarial attacks. Integrating with present systems: The technique of incorporating AI systems into current environments can be confronting because of the difficulties of the technology involved. To avoid interrupting the functionality of the remaining systems, the incorporation procedure must be properly planned and conducted. This is remarkably significant because any interruption to the organization's events might have devastating effects. According to AI research in workflow management systems, the employment of AI planning attempts has the ability to address the issue of AI algorithms integrating with occurring systems. Nonetheless of the obstacles, integrating AI systems with remaining systems is vital to realizing the full promise of AI technology in many businesses. As a result, organizations must carefully plan their incorporation strategy and work together closely with their technology collaborators to ensure a successful implementation [12].

8. CONCLUSION

Ending on a positive note, AI is revolutionizing cybersecurity. AI simplifies actions and frees up cybersecurity professionals to deliberate on new complex duties; by 2022, 40% of these professionals will have used AI. The efficacy of threat detection and response is revealed by

applications from the industry leaders. With an estimated annual growth speed of 23.3% and attainment of $38.2 billion by 2026, the future of AI in cybersecurity seems optimistic. Ethical concerns, the delicate balance between false positives and negatives, and the adversarial security of data threats are models of challenges that need continuous consideration and model enhancement. The approach must be vigilant and systematic, with an emphasis on the incorporation of AI and human expertise. A complete fortification of security standards is guaranteed by this collaboration. If these challenges can be successfully overcome, AI will be able to recognize its full potential and play a vital role in safeguarding our digital environment from new threats.

REFERENCES

[1] E. Benishti, "The Benefits and Risks of Using AI for Cybersecurity: A Balanced Perspective," Ironscale, 2023.

[2] World Economic Forum, "4 ways AI can help us enter a new age of cybersecurity," 2021.

[3] L. Lazic, "Benefit From AI in Cybersecurity," in The 11th International Conference on Business Information Security (BISEC-2019), Serbia, 2019.

[4] N. Papernot, P. McDaniel, A. Sinha, M. Wellman. SoK: Security and Privacy in Machine Learning. Proceedings of the IEEE Symposium on Security and Privacy, San Francisco, CA, USA. 2018.

[5] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In 2017 IEEE Symposium on Security and Privacy (SP). IEEE. pp. 39-57, 2017.

[6] S. Nithya. “Everyone wants to do the model work, not the data work”, Data Cascades in High-Stakes AI " proceedings of the 2021 CHI Conference on Human Factors in Computing Systems. 2021.

[7] I. Jada and T. Mayayise, "The impact of artificial intelligence on organisational cyber security: An outcome of a systematic literature review," Science Direct, vol. 100063, no. 100063, 2013.

[8] H. Nassaji. Qualitative and descriptive research: Data type versus data analysis. Language Teaching Research, vol. 19, no. 2, pp. 129-132. 2015.

[9] T. L. Lash, M. P. Fox, R. F. MacLehose, G. Maldonado, L. C. McCandless and S. Greenland, “Good practices for quantitative bias analysis”, International Journal of Epidemiology, Vol. 43, No. 6, pp. 1969-1985, 2014.

[10] M. Kearns and A. Roth. The ethical algorithm: The science of socially aware algorithm design. Oxford University Press. M. I. T. T. R. Insights, “Preparing for AI-enabled cyberattacks,” MIT Technology. 2022.

[11] A. Klubnikin, "Top 5 AI challenges &

how your company could overcome them," Ritrex, 2023.

[12] E. Anthi, L. Williams, . M. Rhode, P. Burnap and A. Wedgbury, Adversarial attacks on machine learning cybersecurity defences in Industrial Control Systems, Vol. 58, 2021.

Kausar *et al. (IJECI)* 2023

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

International Journal for
Electronic Crime Investigation

DOI: https://doi.org/10.54692/ijeci.2023.0704167

Research Article

Vol. 7 issue 4 Oct-Dec 2023

Cookie Hijacking: Privacy Risk

Kausar Parveen and Noor Fatima

Department of Computer Sciences, University of Engineering and Technology, Lahore

Corresponding author: kausarnawaz6@gmail.com

Received: September 29, 2023; **Accepted:** November 27, 2023; **Published:** December 22, 2023

ABSTRACT

Users may accept more cookies than they require because of suspicious behavior in cookie claimers. And, like, more than they even know. So, as we were measuring real behavior, we were also assessing each of these components' effectiveness, correct? Learning about user opinions of those cookie disclaimers and their entire process of deciding which cookies to accept or, like, reject, bro, was very fascinating. So, guess what? We have conclusively shown that the various images associated with the accept/reject option significantly influence the judgments made by consumers. You won't believe it, but we also discovered that assigning a label to the rejected option has a really big impact. Furthermore, we have confirmed previous studies showing that, well, biased content doesn't actually have a significant impact on customers' judgments. To sum up, black patterns in cookie disclaimers are really important. Users are forced to accept more cookies than they really need, and decision-making is greatly influenced by images and labels. Getting folks to accept those cookies is the main goal. According to our research on user attitudes about cookie disclosures, the presence of the disclaimer only slightly affects the way that various user types make decisions. We provide advice on how to improve. Conditions that apply to different user groups.

Keywords: Cookie utilization, Regulations, Types, literacy of cookies, cyber security, Cookie Notice Analysis.

1. INTRODUCTION

Because the data they contain is always the same, cookies by themselves don't present a concern. They could not infect a computer with a virus or any other malicious programmed. Conversely, if cookies are stolen, certain network attackers could be able to gain access to the browsing session. A significant portion of the internet population uses cookies, making them a pertinent and vital piece of technology. Many individuals in the globe nowadays. Cookies can be used to implement the functionality and history of cookies, but they are not designed to be secure. The accuracy, security, or dependability of the information is not guaranteed. When the security of the transport layer is not enough to prevent online browsing, security load limits cookies to a secure route. The http only feature of the cookie can also

help prevent attackers from perfecting their transmission requirements to secure websites. Hackers breached systems and networks to obtain comprehensive and confidential messages [1].

1.1. Cookie utilization

For several features of web browsing, cookies are required. They facilitate website navigation, login, product addition to a shopping cart, and persistent sign-in for the user agent. The language and style preferences of the user can be saved on the website. As of 2023, Cookiepedia. The way people navigate a website can be tracked thanks to cookies. "Web analytics" is the term used within the area to describe this type of tracking. The web host can use it to enhance the website by finding out about and highlighting the most common methods that users navigate it. Additionally, this allows the website to provide the customer more material, product recommendations, and, well, tailored adverts. Even though this tracking is anonymous and doesn't require sensitive information, it is nonetheless considered private information according to EU data protection laws. Bolinger [2] They definitely offer users better support and a more pleasurable browsing experience in this way. There are other parties besides website hosts or users who may benefit from similar uses of cookies. User profiling and targeted advertising are two uses for cookies. Multiple uses, such as online advertising and espionage, may benefit from web user tracking or profiling. The internet and other websites can be used by third parties to track users. Parties or other similar cookie setups [3].

1.2. Cookie regulations

Official legislation has addressed cookies because of privacy, surveillance, and profiling concerns. The majority of online operations are funded by advertising . The practise of advertising online is super-duper common, and like, super quick growing industry and stuff, like seriously, it's worth like, a lot of money, estimated to be around 227 billion US dollars in the year 2018. It was stated that over 600 billion USD had been spent total on digital advertisements. by a Statista study conducted in 2021. A rise in digital advertising is expected. Notably, expected exceed USD 870 billion by 2026 [4].

1.3. Types of cookies

1.3.1. Session cookies

Session cookies are those that are briefly stored in the memory of the browser. Upon closing the browser, the cookies will be removed. Even if the user leaves the page momentarily, they will still be in the same surfing period! They can save things like login passwords and that kind of thing. In that instance, the user can't just browse the website without having to log back in between sessions and other things! And, they have to repeatedly log in because they can't remember their login information forever and other things [3].

1.3.2. President cookies

Persistent cookies have a tendency to remain on a user's device or browser after the user ends their browsing session. The server establishes cookies that are persistent and have a defined expiration date. Users can delete the cookie prior to its expiration. By supplying cookies with an expired date that are erased upon the client's request, websites have the option to remove cookies from users' browsers. Servers are able to change persistent cookies and change when they expire [3].

1.3.3. First party cookies

First-party cookies are similar to those made by the website the user is visiting or has visited before. They are absolutely necessary for using a certain website's browsing features. And a first-party cookie's domain name and host property are exactly the same. And if you can set and get those cookies, you know, you don't have to log in every time you surf [5].

1.3.4. Third party cookies

Websites other than the one the user is now on are setting third-party cookies, and they are coming from a different domain than what is seen in the address bar of the browser. Since third-party cookies allow advertisers to track visitors across numerous irrelevant websites, advertising is the most popular use case for them. In reality, websites other than the one the user is currently using that is, websites with a domain other than the one seen in the browser's address bar are the ones that set third-party cookies. Third-party cookies are most frequently used in advertising! It makes it possible for advertisers to follow consumers across numerous irrelevant websites [3].

1.3.5. DNS Hijacking

An essential part of the internet infrastructure that enables websites to be identified by their domain names and other attributes rather than their IP addresses is the Domain Name System (DNS). An attacker may be able to steal cookies and personal information through an exploit called DNS hijacking. DNS hijacking involves the compromising of the victim's DNS queries. The DNS queries may end up being answered by a hacked DNS server or one controlled by an attacker. For example, the attacker could employ malware that is placed on the device, infect a nameserver, access the user's router settings and change them. Because of its many guises and methods, it is difficult to defend against [6].

2. LITERATURE REVIEW

2.1. Literacy of cookies

2.1.1. Cookies are not malware however it does brought risks

Cookies don't always represent a risk. They are only text files that assist in coordinating the browser and the remote webserver so that the entire feature set of the website can be accessed. In the midst lie lurking automata and verification. Logging in oh so delightfully allows you to access shopping cart features, preference settings, and some very fancy third-party add-on services. In order to allow users to visit restricted pages without constantly thinking about verifying themselves, cookies are employed to store authentication information. You know, usernames and passwords. Therefore, it is imperative to guarantee the secrecy and integrity of cookies, without a doubt. Alternatively, the user might be anyone with access to the cookies. Even so, the fact that when a server-specific form is filled out, credential data stored in cookies is frequently displayed, where the Even if the contents are hidden from the viewer, an attacker might still be able to replicate the intercepted cookie and assume the identity of the user. The safety of customers may also be jeopardised by implementation problems, particularly when they include web browsers. Internet Explorer versions five and four for Windows 98, 95, 2000, and NT expose a vulnerability allowing websites to view cookies used by other websites, as web browsers hide webpages behind lengthy URLs [7].

2.1.2. Fundamental Types of Cookies

Every action a new user takes on the website is viewed as a fresh business. Online shopping is an excellent example of how session cookies are put to use because they make an item easier to find. When a user checks in again, these cookies will take note of the modifications they made to the website. Similar to these cookies, computers operate by storing all of their settings, such as language selections and bookmarks, on each login. Usually, these incredibly tasty cookies aren't kept on an extremely sophisticated hard drive. And they are stored for an extremely long time forever and beyond. The third group of these incredible treats are, third-party cookies. Another name for them is tracking cookies. They gather data regarding a person's internet activity. Every time a user visits a website, a variety of information about their activities is gathered and sent to the website that set these cookies. Advertising is thus purchasing the gathered data. An individual's preferences, interests, and trends are monitored in this way. Hence, marketers may send a customised advertisement based on the information these cookies gathered. In many ways, nevertheless, this is seen as a faith in the user's online privacy [8].

2.2. Literacy of cyber security

2.2.1. Corporation situation for cyber security

Digital businesses face significant cybersecurity challenges. Companies must adapt security, fraud prevention, and product development teams to design secure, convenient experiences. They must also recognize risks associated with large data sets containing sensitive consumer information. Analytical solutions, which may not have followed conventional software development processes, must also include security safeguards. Companies that use robotic process automation need to make sure that cases with unusual or unexpected components, input cases, and border instances managed and that robotic credentials don't go beyond accepted bounds and endanger public safety. In addition to learning how to enforce and set acceptable developer access rules, companies that create application programming interfaces for external customers also need to learn how to detect vulnerabilities produced by the interaction of different APIs and services. When transitioning from waterfall application design to agile application construction, they need to maintain their tight application security policies. Businesses hoping to increase their digital consumer contacts, for instance, must figure out how to modify the teams in charge of product development, security, and fraud protection so they can build controls and provide experiences that are. Absolutely safe and convenient with authentication! Businesses that use a lot of data analytics must learn how to identify the hazards. Analytical solutions, which may not have followed conventional software development processes, must also include security safeguards. Companies that use robotic process automation need to make sure that cases with unusual or unexpected components, input cases, and border instances are appropriately managed and that robotic credentials don't go beyond accepted bounds and endanger public safety. In addition to learning how to enforce and set acceptable developer access rules, companies that create application programming interfaces for external customers also need to learn how to detect vulnerabilities produced by the interaction of different APIs and services. When transitioning from waterfall application design to agile application construction, they need to maintain their tight application security policies [9].

2.2.2. Data management for cyber security

Data governance is the foundation of privacy. It is helpful to first describe the many forms of data before examining how each one relates to people's security and privacy, as the term "data" is ambiguous and can apply to a vast range of information. It might as well become a gold mine for dishonest advertisers if the information falls into the wrong hands. Internet service providers monitor the websites and browsing habits of their customers and have the ability to take control of them. Cookies are text segments that are downloaded and retained by the web browser, and even while consumers cannot avoid attacks at the level of internet service providers, they may still be able to track websites they visit. Furthermore, browser plug-ins have the ability to track activity on multiple websites [10].

2.3. Cyber security in data management of finance sector concerns

In a calculated attack, fraudsters are calling phone service providers pretending to be consumers by employing sociological engineering techniques. This is the process they use to transfer a phone number, even if it's only temporarily, and keep it in their control long enough to obtain the two-factor authentication associated with it and gain access to the intended account, bank, cryptocurrency wallet, or email. As the phone bug gets unmanageable, there is a chance that any internet accounts connected to this number could be penetrated, which means that the two-factor authorization code could be stolen. In a calculated attack, fraudsters are calling phone service providers pretending to be consumers by employing sociological engineering techniques. This is the process they use to transfer a phone number, even if it's only temporarily, and keep it in their control long enough to obtain the two-factor authentication associated with it and gain access to the intended account, bank, cryptocurrency wallet, or email. As the phone bug gets unmanageable, there is a chance that any internet accounts connected to this number could be penetrated, which means that the two-factor authorization code could be stolen [11].

2.3.1.. Cyber security in data management of medical sector concern

Another recent addition to the market is hospitals. For family DNA services, which preserve genetic information about their clients to be offered in the event of a sought-after medical inquiry or to trace family history, they are currently transitioning to electronic records. If a person's medical records are lost, it could cause them a lot of grief and have tragic consequences. The choice to distribute DNA information is private, with the exception of those who share it first law enforcement officers and often people working in ancestral services. A decrease in sales of several popular family ancestry kits was ascribed to privacy issues related to DNA searches [9].

2.4. Developing security best practices for cookies

To prevent cookie hijacking, a lot of research have been carried out. RPS checks implemented session ID, IP address, OTC, and browser fingerprint. According to Lee et al. [12], a secure and efficient three-stage cookie protection technique is as follows: cryptographic key generation, cookie issuing, and login.

- - -

able 1: Websites that recognize errors found with Newton

Site	Fixed	Notes
Yahoo	Incorrect	Attackers can still access a user's search history, notes, and stock listings even if they only use Yahoo over HTTPS. Yahoo informed us that they will not be fixing this issue at this time due to the intricacy of the code.
Vimeo	Correct	Even if a user only ever uses HTTPS to access Vimeo, his whole account could still be compromised. We alerted Vimeo, and they fixed it.
Magento (250K sites)	?	If a patch is being developed, it was not made clear by the Magento developers.
WooCommerce (650K sites)	Incorrect	WooCommerce developers blamed the vulnerability on the underlying WordPress framework.
BigCommerce (50K sites)	Correct	We applaud the BigCommerce developers for their quick verification process and for releasing a fixed version into production.
Amazon	In progress	A session hijacking attack may be caused by a cross-site scripting attack. An attacker has total access to the user's account. In response, Amazon stated that they are aware of the issue and are developing a solution.

2.4.1. HTTP cookie

A user's web browser receives an HTTP cookie, which is generated by a server and functions something like a small message. To maintain the session's direction and to confirm that both requests are coming from the same browser. These cookies are used for session management (because sessions need to be managed), customization, and tracking. They are returned in answers and other formats as HTTP headers!

Set – Cookie:<name for cookie >=<value for cookie >

Other characteristics that the server may set include expiration, which indicates how long the cookies are valid, and Http Just Bag, which indicates whether the cookie was sent over a secure channel [9].

Table 2: HTTP queries to the domain we have identified as being at risk

Protocol	Connections	Requests	Vulnerable Requests	Exposed Account
HTTP	685,500,365	1,398,044,178	29.908,099	282,459
HTTPS	772,562,024	-	-	-

2.4.2. Cookie notice studies

Cookie alerts are used by websites, you know, to ask for users' permission and, most of the time, to provide them control over these cookies. Consent Management Platforms (CMPs) provide APIs for handling cookie notices, which helps websites comply with, uh, rules. These platforms are third-party interfaces that offer user permission and convenient data storage options, among other things. The rate at which various CMPs are being implemented listen, by 2020 was, like, restricted to the top 10% of websites, many of which, elect to use customized versions of the cookie notice [8].

2.4.3. Cookie notice analysis

2,000 websites inside the European Union were manually inspected in order to determine the scope of cookie-based tracking. It was discovered that there were cookie notices on 57% of them. Using a list of popular CSS selectors were able to recognize cookie notifications on 17,000 websites in Greece and the UK! They noticed that 45 percent of these offer a cookie notice. Based on their study of the notices to determine compliance, very few of them provide a straightforward opt-out choice. To ascertain how user location impacted cookie notice visibility [11].

Table 3: Specifics of the websites that were examined. The percentage is computed using the total number of websites with a cookie notice (45, 044) after the first row.

Type	% websites	Avg, #settings (per site)	One click opt- out (% websites)
No notice	47.3	-	-
Single view	64.6	2.17 (3.01)	11.5
Multiple views	35.4	28.7 (103)	9.96

Thus, it can be said that a good deal of frameworks had security flaws brought on by a variety of circumstances. It is concerning to note that these problems have been discovered in 37 out of the 44 frameworks that were evaluated. Despite its stated purpose as a security precaution, the synchronizer token pattern has been linked to implementation errors. These errors, which result from combining various libraries, could make it easier for attackers from

the same website to get around security protections. Moreover, the CORF token fixation attack is one particular instance of this type of attack. The Flask framework, which was thought to be secure until this specific attack, is now vulnerable. The CORF token fixation attack's security breach emphasizes how critical it is to find and solve these framework flaws. Priority one when developing useful privacy instruments for decision support is usually accorded to understanding users' mental models [8].

3. METHODOLOGY

3.1. Secondary data collection

Secondary data was utilised by basic researchers as a starting point for further investigation. On the other hand, applied researchers are more focused on using the body of knowledge at least in a few different forms to solve specific issues. Secondary data is a prerequisite for good practise. Secondary research aids in determining the direction of Lead to additional primary investigations by highlighting issues with the previous findings. Secondary data provides the measuring tools, relevant interview participants, and instruments for doing primary research into the issues that need to be addressed [12].

The intended goal of the collection of secondary data may have given rise to additional concerns. It's possible that some metrics, classifications, or therapy effects aren't the best fit for the current situation. Definition of secondary data as outdated data. This means that for some uses, this information may not be very current.

4. RESULTS

4.1. Address based authentication

Addresses serve as the basis for authentications. You must possess the IP address of that user. That's when IP Cookie, comes into play. In order to facilitate authentication, it aids in retrieving the IP address. The fact is that an environment is associated with an IP address. Variables for Web users that make it incredibly simple for a Web server (cookie issuer) to obtain the user's IP address and add it to the IP Cookie. When Alice, the user, tries to get in touch with a Web server that accepts the IP Cookie, the server, say, first makes sure that Alice's current IP address matches the IP Cookie that she submitted. If they seem exactly the same, the server believes Alice to be the real owner. Address-based authentication is a very convenient authentication method, even though the authentication process is completely transparent to consumers. But, it's not always the greatest choice [13].

4.2. Password-based authentication

Password-based authentication is supported d by dynamic IP addresses, proxy servers, and prevents IP spoofing. It is essential to make sure that credentials are sent securely when they are transferred from the browser to the Web server across the network. SSL (Secure Sockets Layer) is used in this situation. Through the use of the SSL protocol, secure network communication is possible. However, servers can also validate the cookie owner in another method. Passwords that are encrypted and kept in the Pswd Cookie can also be used by them. Upon receiving Alice's login credentials, the Web server hashes the passwords to increase their security and other features. The Pswd Cookie then stores

these hashed passwords. Thus, Alice just needs to enter her previous passwords each time she wants to log in to a server that accepts the cookie. In the event that the password hash matches the one in Pswd Cookie, the server will identify Alice as the cookie's legitimate owner [13].

4.3. Digital signature based authentication

The idea of digital signature-based authentication advances in this new and advanced technological era. If web servers know users' public keys, they can use digital signature technologies like DSA7 or RSA8. Through the intriguing notion of cookies, it is possible to positively and definitively confirm a user's identity. Thus, users can enjoy the nice benefit of setting up a cookie with a signed time stamp, which removes the requirement for bulky, inconvenient add-on browser software. For example, we discover that, startlingly and enlighteningly, secure cookies are amazingly compatible with various authentication methods like as Kerberos and Radius9.10, when the endearing Alice, bless her heart, has to connect to a faraway Web server that knows Alice's public key. The information about Alice's authentication procedure can be used in conjunction with their authentication methodology, even though it is dependent on a number of variables. These crucial facts can be protected because of a group of secure cookies. Secure cookies are the foundation of our client-to-server authentication strategy. The best part is that SSL can even be used in situations when server-to-client authentication is required [14].

4.4. Maintaining Integrity

Integrity issues also affect cookies. An attacker may, for example, duplicate Alice's IP Cookie and alter it. Using an IP address, then afterwards pose as Alice in front of a Web server. Alice has the ability to modify the contents of her own cookies. The Life Cookie's Cookie Value field allows the Web server to verify the lifetime (expiration date) of the secure-cookie established. Integrity of the lifetime of the secure-cookie set, provided that the cookies are legitimate. Despite the fact that the browser only transmits to the Web server the pertinent Cookie Name and Cookie Value fields in order to verify the integrity of other fields, the Web servers inside the domain can set up a policy with the cookie-issuing server. For instance, the Web server utilises the values that the policy presets for the Domain, Flag, Path, and Secure fields acme.com, True, /, and False, respectively to verify the integrity of the cookies [13].

4.5. Implementation

For user identification, session management, and preference tracking, cookies are widely utilized. Cookies are saved and then given back to the server with each request, enabling it to identify and, for example, personalize the user experience. When employing cookies, security needs to be taken into account. It is recommended that developers utilize secure and HTTP-only settings to safeguard confidential information and prevent malicious attacks like cross-site scripting (XSS). Increased surveillance has resulted from privacy concerns, and laws like the GDPR mandate that websites get user consent before storing certain kinds of cookies and other data [14].

Table 5: Browser and their Connect over HTTP

Browser	Connect over HTTP
Desktop	
Chrome (v.45)	Correct
Firefox (v.41)	Correct
Safari (v.8.0)	Correct
Internet Explorer (v.11)	Correct
Opera (v.32)	Correct
Mobile	
Safari (IOS 9)	Correct
Chrome (v.46, Android 5.1.1)	Incorrect (conditionally)

5. CONCLUSION

Cookies allow customers to save the schedule as soon as they visit the page. Giving more personalised content, more focused advertising, and a better online shopping experience are all made possible by it. However, companies looking for ways to deal with the stricter rules on data security and consumer protection now have access to a wide range of workable solutions. From consumer-facing processes to operations and infrastructure phases, these activities encompass every phase of the company data lifecycle. Usually, a website records twenty cookies. Cookies do have an expiration date, but in the near future, there might be much more. Nevertheless, people need to understand cookies better on a fundamental level.

REFERENCES

[1] K. Renaud and L. A. Shepherd, ‘‘How to Make Privacy Policies both GDPR-Compliant and Usable’’, in 2018 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (Cyber SA), IEEE, pp. 1-8, 2018.

[2] I. Sanchez-Rola, M. DellAmico, D. Balzarotti, P. A. Vervier, and L. Bilge, ‘‘Journey to the Center of the Cookie Ecosystem: Unraveling Actors’ Roles

and Relationships'', in 2021 IEEE Symposium on Security and Privacy (SP), IEEE, pp. 1990-2004. 2021.

[3] C. Matte, N. Bielova, and C. Santos, ''Do Cookie Banners Respect my Choice: Measuring Legal Compliance of Banners from IAB Europe's Transparency and Consent Framework'', in 2020 IEEE Symposium on Security and Privacy (SP), IEEE, pp. 791-809. 2020.

[4] G. Kampanos and S. F. Shahandashti, ''Accept All: The Landscape of Cookie Banners in Greece and the UK'', pp. 213-227. 2021.

[5] M. Hils, D. W. Woods, and R. Böhme, ''Measuring the Emergence of Consent Management on the Web'', in Proceedings of the ACM Internet Measurement Conference, New York, NY, USA: ACM, pp. 317-332. 2020.

[6] J. A. Alharbi, A. S. Albesher, and H. A. Wahsheh, ''An Empirical Analysis of E-Governments' Cookie Interfaces in 50 Countries'', Sustainability, vol. 15, no. 2, pp. 1231-1237, 2023.

[7] C. J. Hoofnagle, B. van der Sloot, and F. Z. Borgesius, ''The European Union general data protection regulation: what it is and what it means'', Information & Communications Technology Law, vol. 28, no. 1, pp. 65-98, 2019.

[8] C. Utz, M. Degeling, S. Fahl, F. Schaub, and T. Holz, ''(Un)informed Consent'', in Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security, New York, NY, USA: ACM, pp. 973-990. 2019.

[9] E. Ma and E. Birrell, ''Prospective Consent: The Effect of Framing on Cookie Consent Decisions'', in CHI Conference on Human Factors in Computing Systems Extended Abstracts, New York, NY, USA: ACM, pp. 1-6. 2022.

[10] B. M. DiCosola III and G. Neff, ''Nudging Behavior Change: Using In-Group and Out-Group Social Comparisons to Encourage Healthier Choices'', in CHI Conference on Human Factors in Computing Systems, New York, NY, USA: ACM, pp. 1-14, 2022.

[11] K. Bergram, M. Djokovic, V. Bezençon, and A. Holzer, ''The Digital Landscape of Nudging: A Systematic Literature Review of Empirical Research on Digital Nudges'', in CHI Conference on Human Factors in Computing Systems, New York, NY, USA: ACM, pp. 1-16, 2022.

[12] R. Houser, S. Hao, Z. Li, D. Liu, C. Cotton, and H. Wang, ''A Comprehensive Measurement-based Investigation of DNS Hijacking'', in 2021 40th International Symposium on Reliable Distributed Systems (SRDS), IEEE, pp.

210-221, 2021.

[13] Q. Chen, P. Ilia, M. Polychronakis, and A. Kapravelos, ‘‘Cookie Swap Party: Abusing First-Party Cookies for Web Tracking’’, in Proceedings of the Web Conference 2021, New York, NY, USA: ACM, pp. 2117-2129. 2021.

[14] X. Hu and N. Sastry, ‘‘Characterising Third Party Cookie Usage in the EU after GDPR’’, in Proceedings of the 10th ACM Conference on Web Science, New York, NY, USA: ACM, pp. 137-141. 2019.

Aftab *et al. (IJECI)* 2023

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

International Journal for
Electronic Crime Investigation

DOI: https://doi.org/10.54692/ijeci.2023.0704168

Research Article

Vol. 7 issue 4 Oct-Dec 2023

Innovative Technologies in Countering Extremism and Terrorism

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: draftab-malik@yahoo.com

Received: September 29, 2023; **Accepted:** November 29, 2023; **Published:** December 22, 2023

ABSTRACT

Terrorism and extremism are distinct concepts often interconnected in discussions about political, religious, or ideological violence. Terrorism mentions to the use of vehemence, pressure, or force to attain party-political, spiritual, or conceptual areas, disrupting normal life and exerting pressure on governments or societies. Extremism, characterized by extreme political or religious views, often involves radical ideologies and intolerance towards opposing beliefs, often seeking dramatic changes in society or politics, rejecting moderation. Extremist ideologies can sometimes lead to terrorist acts, but not all extremists resort to violence. It's crucial to differentiate between extreme beliefs and violent actions, as not all terrorists are driven by extremism. Recommendations to safeguard against security threats are proposed.

Keywords: Extremism, terrorism, criminal gangs, exploiters, Security threats.

1. INTRODUCTION

In recent years, as technology is advancing, the criminals also have access to the tools and technical know-how to modern technology, particularly the terrorists and gangsters involved in white collar crimes, bank frauds, robbery and having access to information to harm others. There are three kinds of terrorism namely revolutionary, sub-revolutionary and establishment; means formation, creation, launching. “Ideologically motivated terrorism” is frequently observed. Smart identification systems, cutting-edge tools for acquiring and analyzing intelligence and use of the Internet can be a consequence in terrorism; these are examples of how technology can be a crucial weapon in counter-terrorism. Individual acts of terrorism are occurring round the word such as shooting in Mosque in New Zealand and recently in Texas USA where, two families suffered heavy loss of their loved ones; the

Police had to shoot the terrorist in Texas Mall. Another attack of deadly shooting occurred in Farmington, N.M. (Northwestern New Mexico) in May 2023, killing and wounding several innocent people by 18 years' boy. Therefore, it is pertinent and important to engage the most recent and innovative technology for surveillance integrated with CCTV cameras, for online identification and detection of groups of terrorists and individuals operating in public places. The devices must be installed in parks, schools, stores and other important buildings to save people from trauma and anguish [1].

2. PRESENTATION AND DISCUSSION

Countering terrorism and extremism involves intelligence gathering, law enforcement, community engagement, de-radicalization programs, addressing root causes, promoting tolerance, and countering extremist propaganda. Terrorism can manifest in various forms, including religious, political, state-sponsored, and ideological terrorism, with religious terrorism often stemming from extremist religious interpretations and political terrorism aiming to achieve political goals through violence. State-sponsored terrorism and ideological terrorism are driven by political and ideological motives, with complex motivations including political change, fear, revenge, chaos, and attracting attention. Terrorist acts are motivated by various factors such as political change, fear, revenge, chaos, or attracting attention, often using violence strategically to advance their agendas [2].

Terrorism significantly impacts societies, economies, politics, and international relations, causing widespread fear and instability by targeting civilians, infrastructure, and government institutions. Terrorism significantly impacts societies, economies, politics, and international relations by causing widespread fear and instability by targeting civilians, infrastructure, and government institutions. Extremism can cause social polarization, erode social cohesion, and foster tensions among different groups, potentially leading to hate crimes, discrimination, and conflicts. To combat terrorism, multi-faceted strategies include prevention through community engagement and education, and counterterrorism measures like intellect gathering, rule application, security, and international cooperation. Programs aimed at de-radicalization and rehabilitation involve providing psychological support, education, and reintegration of individuals involved in extremist activities, addressing root causes. Addressing underlying grievances like socioeconomic disparities, political grievances, lack of opportunities, and discrimination can help reduce extremism. Reducing extremism can be achieved by addressing socioeconomic disparities, political grievances, lack of opportunities, and discrimination [1].

Some important policy features of US Department of State are relevant regarding transparency, anti-corruption, arms control, combating drugs and Crime and countering the terrorism. Cyber issues. Programs of public diplomacy to educate people must be initiated to inculcate the importance of science and technology in collaboration with other nations. The Counter

Terrorism Department (CTD) in Pakistan, has been entrusted with very important responsibilities such as crime scene investigations, cross-examinations, interrogations, dealing with intelligence and anti-terrorism programs. Apart from suicide bombing and shooting to create harassment, the terrorists also commit several types of frauds to grab money and fund raising; such as, blackmailing, bank frauds, fraud using phone, computer frauds, credit card fraud, investment schemes, currency schemes, forgery and insurance fraud. Most of these frauds are committed abusing the Information Technology (IT) systems and breaking the security barriers of infrastructure of I.T Networks. In order to avoid such incidents, the financial institutions, banks and other entrepreneurs must at the first instance install the most recent licensed software for their working and that for Network Security. Banks are advised to manage their dedicated Intranet and extranet and to have departmental communications as well as with stock holders and stakeholder [3].

Techniques for detonating explosives early or avoiding their triggering are part of the science and technology developed specifically to combat terrorism. The majority of S&T counterterrorism technologies are very beneficial for general intelligence, law enforcement, or public health-related goals. According to [4], the psychological issues and factors cannot be ignored as they contribute towards terrorism in Pakistan, the terrorist attacks against the environment of Pakistan's history, present geopolitical environment and contemporary societal environment. The results might also serve as a roadmap for resolving this fundamental problem. In most of the religions, there exist provision for ritual murder, martyrdom and self-sacrifice, the terrorists exploit this concept to convince their suicide bombers, attackers and shooters. This is in fact a kind of mental disorder. However, in Islam “self-sacrifice” is “haram”.

One of the most salient aspects in terrorism is the extremism in various forms. In Pakistan, the NACTA (National Counter Terrorism Authority) is contributing towards eradication and countering terrorism and extremism through education, awareness and by all other relevant means. A useful document presented by NACTA in [5] is about Pakistan’s narrative in connection with Terrorism and Extremism. It discusses a popular FATWA signed by 1800 renowned muslin religious scholars of all sects of Islam including scholars of Al-Azhar University and Imam-e-Kaba. It redirects the salient features of Quran and Sunnah to create harmony between all religious sects in Islam. The educational institutions must be kept under close scrutiny to observe if they are spreading, endorsing militancy hatred, terrorism, violence, extremism must be taken to law enforcement agencies for legal action. There exist different definitions and analyses of violent extremism, it is a complicated issue, used in academic settings. In different communities, depending on their particular structures, it is viewed and tolerated in different ways. The academic institutions should be prohibited and forbidden to endorse and promote hatred for Pakistan and extended training for terrorism. In the presence of Pak Army, Air Force and Pakistan Navy,

no individual, groups or provinces can declare Jihad. The role of implementation of Science and Technology is of immense importance. According to [1], technological solutions exist for almost all problems related to terrorism. Pakistan being the front state as one of NATO allies suffered heavy losses, special in terms human casualties about 70,000. The terrorists used explosives against innocent citizens by means of conservative weapons. Majority of attacks were led by terrorists infiltrating from Afghanistan into Pakistan; though the borders are well-fenced now.

Pakistan Army vacated about 50,000 square km area sideways with western border from the terrorists capturing their arms and ammunition. For this purpose, the requisite technology equipped with modern sensor devices using Artificial Intelligence algorithms is most suitable for observing, checking and monitoring suspicious people and vehicles. The technology industrialized Midas detection System by Turkey namely Bayraktar TB2, STM Kargu–2 for_integrated security solutions is useful for constant watching and surveillance.

The present author of this essay had an opportunity to work with highly skilled and qualified Chinese team of experts of Department of Automation, Shanghai Jiao Tong University, Shanghai China, having position in World QS Subject Ranking: 26 and Engineering & Technology World QS University Ranking: 47. The team developed the systems for monitoring and surveillance based on automatic detection and analysis without human intervention. The entire systems have been published and reported in [2] and [3] based on "Multi-Stream 3D latent feature clustering for abnormality detection in videos" linked with CCTV at important places. The second System works with "Multi-Level Two Stream Fusion based Spatio-temporal Attention Model for Violence Detection and Localization". These systems may be installed at impotent places. The process of abnormal behavior in surveillance videos and its detection is essential for public safety and monitoring, which requires constant focus.

The other innovative technology termed as Geo-fencing has been very effective to detect criminals and terrorists as reported in [6]. It works in conjunction and combination with RFID (Radio-Frequency Identification), CRD (Call Record Data) along with GPS (Global Positioning System) and WiFi. The procedure is to create a computer-generated, simulated, cybernetic and virtual geographical boundary to cause and activate a marketing action to a mobile device (may be mobile phone), so that when user enters into or exits that virtual boundary around certain location in GPS or RFID is detected.

The use of innovative technology called "De-militarized Zone" with Networks is of extreme impotence as strongly recommended in [7]. This technology provides access to the untrusted gangsters who intervene into a Network and it delivers extra layer of security to Network or LAN (Local area Network). Its major goal is to have access to untrusted networks. The companies store Data from external source like, Voice over Internet Protocol (VoIP), (Domain Name System) DNS, FTP (File Transfer Proto-

col), Mail, proxy and their web-servers in the Demilitarized Zone for security and sensitivity of DATA. The Demilitarized Zone makes it difficult for hackers and trackers to have access to the company's important information. It works with two fire walls (the hardware firewall and software firewall) to protect the working of router from hackers and trackers. Using this procedure, the disturbances caused by terrorists in communication, stealing servers, spoiling websites and causing financial loss can be avoided [8].

3. RECOMMENDATIONS

1. The well tested, innovative, cultured and refined technologies must be used to combat and fight with terrorists. The system for gathering intelligence must be coupled with new technology to minimize pressures and strains.

2. Methods for averting and triggering of explosives or guaranteeing their early detonation are part of the science and technology and important to fight with terrorism.

3. The terrorist use cyberattacks, DDoS and different malware, phishing to have access to their targets. Therefore, precautionary measures must be taken.

4. Following technologies being effective in the areas of combating terrorism are also recommended:
   - Biometrices
   - Data sharing
   - Drones intelligence, surveillance and targeting

5. The border security is required to be strengthened using Video surveillance of terrorists.

6. The majority of "Science and Technology policies" regarding counterterrorism technologies are very beneficial for areas related to general intelligence, law enforcement, or public health.

7. The law enforcement agencies must be strengthened with adequate funding, most recent innovative ecologies, technically skilled man power for using the technologies concerned to counter terrorism.

8. The specific countering terrorism strategies must include premature blasts recovered from terrorists to safe guard the public health.

9. In order to counter the de-radicalization, terrorism, violent extremism, sectarianism in Islam promoted by madrassas must be closely observed by task force already exiting in Government files. The concerned persons promoting such dogmas, creeds, doctrines and beliefs must be arrested.

10. The Social scientists frequently use the Information Technology, Computer Software operational research and various statistical packages, therefore they examine the patterns in occurring of

terrorist actions and carryout depth studies in the areas of terrorism by gangs, hate crime related to extremism and their social networks along with their modus operandi.

11. The Social Scientists have to focus, while conducting surveys on different terrorist groups and their dynamics with respect to society and how terrorist groups are formed and operate; investigate if these groups are instigated, prompted and activated for mass destruction using armaments by religious fanatics.

12. The Social scientists must use as advocated in [9], the internationally known and developed, during last 60 years, the Databases and famous open Data Sets on terror incidents.

13. For the sake of oversight and public safety, it is crucial to identify unusual behavior in surveillance footage. However, it is a difficult process that requires ongoing emphasis and "human care" for human-based monitoring systems. Automatic detection of such situations is therefore very important.

14. Several terrorists are arrested and produced before the court, who are acquitted. This amounts to injustice with victims of terrorists. The major reasons are that naïve cases are prepared against them; such as defective FIR against unknown accused persons, eye-witnesses are not mentioned, statement under section 161 is defective or absent from the record file, material and forensic evidence is not properly collected from scene of offence, late submission of challan and finally poor prosecution destroys the case and hence the terrorists are set free.

15. It is strongly recommended that all the defects in registration of FIR, investigation, identification parade, confessional statement and contradiction in medico-legal report must be eradicated and removed.

16. It is recommended that cell phone and telephone surveillance technology for counterterrorism purposes may be provided to the Counter Terrorism Department (CTD) Pakistan. There is an urgent need to adopt, develop and implement Data fusion and Data mining in the field of intelligence and its analysis. Moreover, introduction of protective gear, sensors for the purpose of communication is most required technology.

17. The UN Security Council and the Counter-Terrorism Committee Executive Directorate (CTED) helps nations countering Terrorist and has been providing consultation to Pakistan on counterterrorism. Pakistan may implement the useful suggestions.

18. While enhancing the efforts to implement Science and Technology, the

Universities must be encouraged to take up research projects at MS, M.Phil and PhD level in the areas of counter-terrorism coupled with other disciplines for more effective use of new technologies; in collaboration of industry with academia. The Higher Education Commission may play a pivotal role to award Research Grants.

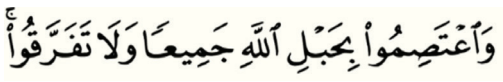
19. The Government may also overhaul, refit and update the existing Technologies to counter terrorism.

20. The key to success is to concentrate in the Surveillance, constant security and arranging technology where ever required to combat terrorism and to detect the suspicious behavior.

4. CONCLUSION

21. This is right time to use all available technologies and measures to avert and counter terrorism and extremism in all forms. We must control smuggling in the name of trade and infiltration of terrorists from Afghan borders; causing serious economic crises. The essential commodities like edible oil, wheat, rice, lentils, poultry products and meat along with several imported items from Pakistan are smuggled on daily bases. Pak Army is best in the world and all Pakistanis love and salute them. In recent acts of grave violent extremism, terrorism and harming national property and attacking the army installations, the entire nation is with Pak Army. Strict action against the terrorists and extremists as well as persons camouflaging, masking, obscuring, protecting and instigating must be taken to task and punished according to law regarding terrorism. The terrorist activities happened on 9th May 2023, a number of terrorists are alleged to be from Afghanistan, who accompanied Pakistanis to destroy Army installations and national assets countrywide.

A beautiful quotation from Al-Quran: Sure Al-e-Imran Verse 103 is presented:



Muslim scholars must promote this Quranic philosophy to counter extremism and to create coordination, congruence and harmony between various sects of Islam.

REFERENCES

[1] S. Ahmad, “Combating Terrorism Through Technology in Pakistan”, CSRC Centre for Strategic and Contemporary Research. 2022.

[2 C. Zdanowicz, C. Alvarado and K. McCleary, “Texas Shooting”, CNN, 2023.

[3] M. Asad, H. Jiang, J. Yang and Aftab Ahmad Malik, “Multi-level Two Stream Fusion based Spatio-temporal Attention Model for Violence Detection and

Localization", International Journal of Pattern Recognition and Artificial Intelligence. vol. 36, no. 1, pp 51-63. 2021.

[4] A. Tamizuddin and T. Mahmood, "Terrorism in Pakistan: the psychosocial context and why it matters", B. J-Psych International. vol. 15, no.1, pp. 20-22. 2018.

[5] NACTA," Pakistan's National Narrative against Terrorism and Extremism, Developed & Maintained by IT Wing (NACTA), 2023.

[6] A. A. Malik, M. Asad and W. Azeem, "Child Kidnapping and Abuse by Gang-Criminals and the Legitimate Custody of Minor to Parents after Rescue and Use of Geo-fencing to arrest the Absconding Criminals", International Journal for Electronic Crime Investigation. Vol 6, no. 3, pp 1-8. 2022.

[7] A. A. Malik, M. Asad and W. Azeem, "Frauds in Banking and Entrepreneurs by Electronic Devices and Combating Using Software and Employment of Demilitrized Zone in the Networks", International Journal for Electronic Crime Investigation, vol. 6, no. 4, pp. 1-6. 2022.

[8] M. Asad, H. Jiang, J. Yang, E. Tu and A. A. Malik. Multi-Stream 3D latent feature clustering for abnormality detection in videos. Applied Intelligence, vol 52, pp 1126-1143. 2021.

[9] J. Ebner," Fighting International Terrorism with Social Science Knowledge", Footnotes; Public Information Office. 2021.

Asif *et al. (IJECI)* 2023

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

International Journal for
Electronic Crime Investigation

DOI: https://doi.org/10.54692/ijeci.2023.0704169

Research Article

Vol. 7 issue 4 Oct-Dec 2023

Volatile Data Acquisition and Analysis by Using Memory Forensics Techniques

Rabia Mehmood

Department of Computer Sciences, COMSATS University, Lahore

Corresponding author: rabiamehmoodciit@gmail.com

Received: September 30, 2023; Accepted: November 29, 2023; Published: December 22, 2023

ABSTRACT

Memory forensics is a vital component of digital investigations, involving the analysis of volatile memory (RAM) in computer systems to gather evidence, identify malicious activities, and reconstruct cybercrime incidents. This paper provides an overview of memory forensics, highlighting its definition, importance, purpose, and scope. It explores the evolution and significance of memory forensics in response to increasingly complex cyber threats. The memory forensics process is discussed, covering memory acquisition and analysis. Legal and ethical considerations related to the admissibility of memory evidence and privacy protection are examined. The paper also discusses the types of memory, including physical and virtual memory, and their characteristics and significance in memory forensics. Furthermore, it explores the memory acquisition process, different methods, tools, and techniques used, as well as the importance of preserving evidence integrity. Finally, the paper introduces various tools for memory analysis, such as Volatility, Volatility Workbench, FTK Imager, Encase, Hibernation Recon, and Xplico, and highlights their role in extracting valuable evidence from memory dumps.

Keyword: Memory forensics, volatile memory, digital investigations, evidence, malicious activity, cybercraime.

1. INTRODUCTION

Memory forensics is the investigation and examination of volatile memory (RAM) in computer systems in order to gather data, spot malicious activity, and reconstruct instances involving cybercrime. It is essential for gathering information, recognizing malicious activity, and looking into cybercrime incidents. Memory analysis is crucial since traditional disk-based forensics may not be able to capture all pertinent data on their own. Memory forensics can provide information about current system activity, encryption keys, network connections, and more [1].

Examining and analyzing a computer's volatile memory is a part of memory forensics. The operating system and any open apps keep their active data and code in volatile memory. Memory forensics are crucial because they give investigators access to data that might not be

saved on disc, like passwords, encryption keys, network connections, and active processes. Reconstructing the sequence of events during an incident and determining the existence of malware or unauthorized activity both benefit from this knowledge. Extraction of useful data from a system's volatile memory is the main goal of memory forensics. Investigators can find information on cyberattacks, data breaches, system intrusions, and other security problems by looking into memory. Memory forensics aids in retracing the timeline of events, spotting malicious activity, and comprehending the offenders' actions. Due to the present cyber threats' increasing complexity, memory forensics has quickly developed. Traditional disk-based forensics by themselves would not be able to paint a complete picture of the occurrence due to the increasing sophistication of malware and advanced persistent threats (APTs). Memory analysis adds to disk-based forensics by providing details on the system's configuration at the time of the incident, including loaded drivers, active network connections, and in-memory artifacts. Memory forensics has evolved into a crucial investigative tool for cybercrime, inspiring the creation of specialized tools and procedures [2].

2. MEMORY FORENSICS PROCESS

2.1. Acquisition of Memory

The target system's memory must be acquired as the initial stage in memory forensics. Memory acquisition can be accomplished by a number of techniques, such as physical acquisition and live acquisition. The target system's memory modules are taken out and imaged using specialized hardware during physical acquisition. Contrarily, live acquisition is removing memory from a functioning system without shutting it down [3].

2.2. Memory Analysis

The next stage is to analyze the memory to draw out important information after it has been acquired. The process of extracting, decoding, and interpreting data from images of memories that have been acquired is known as memory analysis. To recognize active user sessions, open network connections, and operating processes, many strategies can be employed. Data structures including process lists, file handles, registry hives, and kernel objects are also examined as part of memory analysis to look for signs of malicious activity [2].

3. LEGAL AND ETHICAL CONSIDERATIONS

3.1. Admissibility of Memory Evidence

The admissibility of volatile memory evidence in court proceedings is one of the difficulties in memory forensics. The brittleness of memory raises concerns about its dependability and vulnerability to manipulation. Memory forensics findings have, however, successfully been admitted as evidence in the past. Memory analysis, for instance, was essential in locating encrypted data and supplying proof of criminal activity in the United States v. Stewart case. Courts are setting standards for the admission of memory forensics findings as they increasingly recognize the significance of memory forensics in locating crucial evidence [4].

3.2. Privacy and Data Protection

Memory forensics investigations entail accessing and examining private data kept in a system's volatile memory. During these investigations, it is crucial to make sure that privacy rules and data protection laws are being followed. Investigators must take reasonable efforts to preserve the privacy of those concerned and handle sensitive

material discovered during memory analysis responsibly. This entails putting appropriate data anonymization procedures into place and making sure that only authorized individuals have access to the data gleaned via memory forensics. Proper handling of privacy and data protection issues is crucial to maintain the integrity of the investigation and avoid potential legal and ethical complications. Investigators should follow established guidelines and best practices for conducting memory forensics to protect the rights and privacy of individuals involved while still obtaining valuable evidence. It is crucial for investigators to keep up with the most recent methods, devices, and regulatory requirements as memory forensics develops. Memory forensics can be effectively used to obtain important evidence and advance the area of digital forensics by adhering to suitable protocols and best practices [3].

4. TYPES OF MEMORY

4.1. PHYSICAL MEMORY

4.1.1. Definition and Characteristics

RAM (Random Access Memory), another name for physical memory, is the actual memory modules that are installed in a computer system. Its job is to keep track of the information that the system is currently using. Performance of the system is impacted by physical memory's unique properties, including capacity, speed, and access time [5].

4.1.2. Memory Organization

A computer system's memory components are arranged hierarchically as part of memory organization. It covers all memory types, including cache memory, main memory (RAM), and virtual memory. A memory hierarchy that maximizes data access and storage efficiency is created by the different capacities, speeds, and costs at each level [5].

4.1.3. Volatility and Data Persistence

Physical memory is volatile, meaning its contents are lost when the power is turned off. This characteristic poses challenges for memory forensics investigations, where investigators analyze memory contents for evidence or artifacts. To preserve data during system shutdown or when physical memory is insufficient, data persistence mechanisms such as hibernation files and page files come into play [6].

5. VIRTUAL MEMORY

5.1. Introduction and Purpose

Modern operating systems use the memory management method known as virtual memory. In situations where physical memory is insufficient to support all current processes, it enables the system to utilize disc space as an extension of that memory. Programs won't run out of memory because to virtual memory's wider accessible memory area and efficient memory allocation [5].

5.2. Address Translation

Address translation is a critical process in virtual memory management. It involves mapping virtual addresses used by processes to physical addresses in the underlying physical memory. This mapping is typically maintained through page tables. Analyzing virtual memory mappings is crucial in memory forensics investigations to understand the memory layout and locate specific data or artifacts [7].

5.3. Page File Analysis

The page file, also known as the swap file, is a file used by operating systems to store pages of memory that are not actively being used. During memory forensics investigations,

analyzing the page file can provide valuable insights and potential artifacts. Investigators examine the page file contents to reconstruct system activities, identify relevant evidence, and gain a deeper understanding of the system's state at a given time [7].

6. MEMORY ACQUISITION

It is an important component of digital forensics, and it comprises the process of obtaining data from a computer system's volatile memory. Memory acquisition provides vital information about a system's state, including active processes, network connections, and system configurations in digital forensics investigations. This information can be used to investigate security incidents, detect malware, and detect harmful activities on a system.

The memory forensics acquisition process can be divided into the following steps:

- Identify the system: The first step is to identify the system that will be acquired. This involves identifying the operating system, hardware configuration and any installed security software.
- Prepare the acquisition media: The next step is to get the acquisition media prepared. This media must be large enough to hold the memory dump and forensically sound.
- Acquire the memory dump: A variety of tools can be used to obtain the memory dump. These tools can be used to acquire the entire system's memory or to acquire a particular portion of memory.
- Verify the acquisition: After acquiring the memory dump, it should be verified to ensure that it is complete and accurate.

The primary goal of memory acquisition is to collect the state of the system at a certain point in time. There are two distinct methods of memory acquisition: live acquisition and dead acquisition [8].

6.1. Live Acquisition

The process of acquiring memory data while the system is still running is known as live acquisition. This method can offer real-time information on the current state of the system, such as active processes, network connections, and system configuration. Live acquisition is beneficial in instances where the system's current status must be captured, such as when an incident is ongoing [7].

6.2. Dead acquisition

It is the collection of memory data after the system has been shut down. This method is useful when live acquisition is not available, for as when a system is not responding or has been turned off. Dead acquisition can also be beneficial in situations where the system state must be preserved, such as in cybercrime investigations or when dealing with sensitive systems that must not be disturbed [8].

6.3. Hardware-based tools

It include physically removing memory chips from the system and reading the memory contents with specialized hardware devices. This method is usually more difficult and requires higher levels of expertise than software-based memory acquisition methods. However, it may be more dependable than software-based methods since it is less susceptible to interference from running processes or malware. Magnet AXIOM Live and Access Data FTK Imager are a few common hardware-based tools for live memory acquisition [9].

6.4. Software-based tools

Running software on the system to capture memory contents and save them to a file for later analysis acts as what software-based tools do. The Sleuth Kit, X-Ways Forensics, Volatility, Rekall, and Redline are some notable software-based methods for live memory acquisition. These tools scan the system's memory for specified data structures or patterns, and then copy the relevant data to a file for further analysis [8].

6.5. Hybrid memory acquisition tools

It combine hardware and software-based methodologies to provide a more efficient and eliminated approach. Hybrid tools often involve connecting a hardware device to the system and then running software to retrieve the contents of the memory. As the hardware device may provide direct access to the memory, this method may be more efficient and reliable than software-based methods. Magnet AXIOM, Cellebrite UFED Cloud, and Access Data FTK Imager Enterprise are some popular Hybrid acquisition solutions that combine the features of hardware-based and software-based tools. To protect the integrity and admissibility of evidence, certain processes must be followed when undertaking memory acquisition. This includes documenting the acquisition process, verifying the memory image's integrity, and securely storing and transporting the evidence. Documentation should include information regarding the system being analysed, the date and time of the acquisition, the method utilised for acquiring memory, and any pertinent acquisition process details [9].

6.6. Verifying the memory images integrity

It entails comparing the acquired memory data to a known good copy of the system's memory. This is referred to as validation, and it ensures that the acquired data is correct and has not been manipulated. One typical method for validation is to create a digital fingerprint of the memory image using a hash algorithm. To confirm that the obtained data has not been altered or changed, this fingerprint can be compared to the hash value of a known good copy of the memory. Secure evidence storage and transport are critical to ensuring that the memory image is not compromised during the investigation. To safeguard against unauthorized access, memory images should be kept in a secure environment. It is also critical to preserve the memory image during transport, which can be accomplished through the use of encryption or other secure transfer methods. It is critical to follow the chain of custody procedures while transporting evidence to ensure that the evidence is not corrupted or altered during transit. Memory forensics is an important part of computer forensics investigations. Memory acquisition enables investigators to gain access to crucial information about the state of a system, such as active processes, network connections, and system settings. To maintain the integrity and admissibility of evidence, proper procedures, including documentation, validation, and secure storage and transit of evidence, must be followed [10].

6.7. Analyzing

So, whenever we start analyzing the memory forensics, we have different option and different tools for analyzing its content A Memory image is basically a dump of RAM i.e., volatile memory and contain the information of processes and different sectors of computer. The analysis part of memory is crucial as every single artifact has its important in a case for this section, we are using different variety of tools to analyze a basic memory image from a victim’s hard drive as an example. And see how to recover most of the evidence from the memory sample the The memory

sample using in this case is downloaded from an online website:https://github.com/volatilityfoundation/volatility/wiki/2.6-Win-Profiles [7].

6.8. Tools

The tools require for the analysis of memory forensics are given below:

- Volatility
- Volatility workbench
- Ftk imager
- Encase
- Hibernation recon
- Xplico

We are using some of the tools mentioned above for analysis purpose [8].

6.9. Volatility

The volatility framework is an open-source collection of different tools. This framework is written in python language and it is under the License of GNU General Public License

This tool was used by mostly forensic examiner to examine the memory analysis Dump Lets start the Analysis part

The volatility framework official link is: https://github.com/volatilityfoundation/volatility

So, the volatility framework comes with different profile options for every type of Operating System e.g., for window it has a Win profile same for mac and Linux Systems. The command for checking the OS profile is:

vol.py -f (location of the dump) image info

Basically, this is used to determine whether we are working on a right System or not As the profile is confirmed we move to the different option for Analysis (Depending on the case) Mostly the other option used are:

amcache - Print AmCache information

cmdscan - Extract command history by scanning for COMMAND_HISTORY

dlldump - Dump DLLs from a process address space

evtlogs - Extract Windows Event Logs (XP/2003 only)

filescan -Pool scanner for file objects

netscan -Scan a Vista (or later) image for connections and sockets

pslist - Print all running processes by following the EPROCESS lists

These are the options that are mostly used in the analysis part and most of the time we can see the malicious activity from the output of these commands

Now move to the next tool i.e., volatility workbench. The volatility workbench is also a part of volatility framework but has A GUI Framework. The volatility workbench is used as it is fast and has a GUI interface and easy to use. The volatility workbench can be downloaded from the given link https://www.osforensics.com/tools/volatility-workbench.html.

Now we move to the other tools i.e., hibernation recon. The memory dump is not a little piece of evidence. It has a variety of different artifact inside of it some of them includes:

- Hiberfill.sys
- Pagefile.sys
- Swapfill.sys

Every .sys file has his importance in the analysis. The hiberfill.sys file contains all the information when a computer goes in the hibernation state. Basically, when a person

wants to go somewhere and he doesn't want to shut down his system and close their apps and website he use this method. The advantages of using this method are that it saves all the running processes into memory and when we back and power on the computer we have the same screen as we left off. Mostly people use this method to save their battery life of laptops as in hibernation mode the system go in sleep but doesn't use power. During the analysis this file help us a lot if some attacker hibernates their PC and go somewhere we can extract the information of their PC by analyzing this file [12].

6.10. RECOVERY

The practice of recovering erased or concealed data from the volatile memory (RAM) of electronic devices, particularly computers, is known as memory forensics recovery. This feature of memory forensics is essential because it allows access to data that would not be available using conventional disk-based forensic techniques. Data recovery in memory forensics involves extracting and reconstructing obscured or deleted data from the volatile memory (RAM) of electronic devices. This can be done using software-based or hardware-based methods and tools. In this paper, software-based tools are the main topic [9].

6.10.1. Volatility

Volatility is an open-source memory forensics framework for malware analysis and incident response. Microsoft Windows, Mac OS X, and Linux are all supported by this Python-written program. One of the best open-source software tools for 32-bit and 64-bit systems to analyze RAM is Volatility. It has the ability to examine a variety of dump types, including raw dumps, crash dumps, VMware dumps (.vmem), virtual box dumps, and more. [13] The RAM from which the data can be recovered is examined using a volatility tool. With the aid of HashCalc, the hash value of the gathered evidence from stored files, deleted files, encrypted emails, and password-protected files can be calculated. This value is then compared with the files that were successfully retrieved [8].

6.10.2. Autopsy

A GUI-based open-source digital forensic program called Autopsy can efficiently analyze hard drives and mobile devices. Thousands of users all over the world use autopsy to determine what actually transpired in the computer. Military investigators and corporate examiners both frequently use it because of some of its features:

a) File type detection

b) Media playback

c) Registry analysis

d) Photos recovery from memory card

e) Extract geo-location and camera information from JPEG

f) Extract web activity from browser

g) Show system events in graphical interface

h) Timeline analysis

i) Extract data from Android – SMS, call logs, contacts, etc

j) It has extensive reporting to generate in HTML, XLS file

k) Format Alphabetical Memory forensics tools are used to acquire and/or analyze a computer's volatile memory (RAM) [9].

6.10.3. MANDIANT Memoryze

A memory analysis tool is MANDIANT Memoryze, formerly MANDIANT Free Agent. Memoryze is able to analyze live memory while a computer is running in addition to acquiring the physical memory from a Windows system. Any analysis can be performed on a live system or an acquired image [8].

6.10.4. Belka soft Evidence Center

An investigator can easily gather, search, analyze, store, and share digital evidence from computers and mobile devices using Belkasoft Evidence Centre. The toolkit analyses hard drives, drive images, memory dumps, iOS, Blackberry, and Android backups, as well as chip-off dumps, to quickly extract digital evidence from a variety of sources. The most crucial forensically significant artefacts are automatically analyzed by the Evidence Centre and presented for review, closer inspection, or addition to the report [9].

6.10.5. WxHex Editor

WxHexEditor is an open-source cross-platform hex editor written in C++ and wxWidgets. It uses 64-bit file descriptors (supports files or devices up to 264 bytes). It does not copy the whole file to your RAM. This makes it faster and lets it open very large files. Some of the features are; you can copy/edit your Disks, HDD Sectors with it [8].

6.10.6. HELIX3

This tool can collect data from physical memory, network connections, user accounts, executing processes and services, scheduled jobs, Windows Registry, chat logs, screen captures, applications, drivers, environment variables and Internet history. And then data is analyzed on the basis of that report is generated [10].

7. REPORTING

In the discipline of memory forensics, reporting entails making the findings and conclusions that result from the analysis of memory data explicit and understandable, particularly for non-technical people [15]. The main goal is to give a succinct overview of the investigation's findings so that someone who might not have a strong technical background can easily grasp it. The report serves as a tool for bridging the communication gap between the technical analysis and the intended audience by streamlining the findings' presentation. It uses plain English rather than technical jargon to efficiently communicate the main points. The focus is on presenting important findings, such as suspicious activity, proof of unauthorized access, and potential security flaws, without getting bogged down in complex technical details unless absolutely necessary for understanding. The study also describes how these conclusions can affect the concerned organization or people. It is significant to note that the report preserves the correctness and integrity of the findings and conclusions obtained from the memory analysis despite being aimed at a non-technical audience [9].

7.1. Purpose of the Report

This report's objective is to describe the results and recommendations of the memory forensics investigation done in connection with the XYZ case. We sought to find any malicious activity, find possible security holes, and make suggestions for increasing system security by examining the memory image obtained from the compromised machine [7].

7.2. Executive Summary

Memory forensic analysis revealed several important findings. First, several instances of suspicious processes were identified, indicating

the presence of malware in the system. These processes were found communicating with external IP addresses, suggesting an unauthorized network connection. Additionally, the analysis uncovered evidence of file tampering and attempts to cover up malicious activity. Based on these findings, we recommend immediate incident response actions to mitigate the potential risks associated with a compromise [6].

7.3. Case Background

A security incident involving unauthorized access to the company's network was reported in the XYZ case. To ascertain the scope of the compromise, track down the attacker, and evaluate the effect on the organization's systems and data, a memory forensics investigation was started. Utilizing best practices, the analysis was done on a memory image that was taken from a compromised server [12]

7.4. Methodology and Tools

The memory forensic analysis was conducted with tried-and-true methods and tools. To ensure proof integrity, the memory image was acquired using a hardware-based write blocker. The volatility framework, a popular open-source memory forensics tool, was then used to process and analyze the acquired memory image. Additionally, thorough analysis and the extraction of pertinent artifacts were done using the Rekall framework [11].

7.5. Finding and Analysis

Several patterns were discovered during the memory analysis, which gave important information about the attacker's activities. Malicious software is present on the system if suspicious processes like "backdoor.exe" and "malware.exe" are present. These processes' network connections revealed communication with well-known command and control servers connected to malware campaigns. Additionally, a sophisticated attack intended to avoid detection was revealed by the analysis of memory structures, which revealed attempts to alter crucial system files. These results strongly support the need for a focused intervention [13].

7.6. Interpretation and Conclusions

The XYZ system has been infected by sophisticated malware, according to the findings of the memory forensic analysis. In order to maintain persistence, the attacker set up command and control channels, modified system files, and gained unauthorized access using sophisticated methods. The confidentiality, integrity, and availability of data within an organization are seriously at risk from a compromise. It is advised to take immediate corrective action to stop the breach, get rid of the malware, and secure the compromised system [14].

8. CONCLUSION

Several suggestions are made to improve the security posture and stop upcoming incidents in light of the analysis's findings.To stop further communication with the command-and-control infrastructure, isolate and disconnect the compromised system from the network. To comprehend the capabilities and potential effects of the identified malware, conduct a thorough malware analysis. In order to fix flaws that an attacker has exploited, all software and operating systems should be updated and patched. For the purpose of quickly identifying and responding to similar incidents, enhance network monitoring and intrusion detection capabilities. Enhance user education and training initiatives to inform staff about typical attack vectors, phishing attempts, and social engineering strategies.

REFERENCES

[1] E. Casey, A. Richard, and J. M. James, "Handbook of digital forensics and investigation," Academic Press, 2014.

[2] H. Carvey, "Windows Forensic Analysis Toolkit: Advanced Analysis Techniques for Windows 8," Elsevier, 2014.

[3] A. Marshall, E. Casey, and T. Mørk, "Digital forensics: digital evidence in criminal investigations," John Wiley & Sons, 2014.

[4] J. Schatz and T. Yu, "Memory forensics using volatility framework in a virtual environment," Digital Investigation, vol. 10, no. 4, pp. 326-335, 2013.

[5] I. Ghafir, "The legal admissibility of memory forensics: An overview," in International Conference on Information Networking (ICOIN), pp. 628-633, 2018.

[6] H. Richardson, D. O'Sullivan, and N. A. Le-Khac, "A framework for the forensic investigation of live systems," Digital Investigation, vol. 9, no. 1-2, pp. 32-41, 2012.

[7] S. Silberschatz, P. B. Galvin, and G. Gagne, "Virtual Memory: An Overview," in Operating System Concepts, 9th ed. Wiley, 2013.

[8] B. Carrier and E. H. Spafford, "Forensic Analysis of Volatile System Memory," Digital Investigation Journal, 2011.

[9] A. Walters, "Memory Analysis and the Windows Page Cache," Digital Investigation, vol. 13, no. Suppl. 1, pp. S55-S64, 2015.

[10] G. Soghoian, "Analyzing the Windows Page File and Memory," IEEE Security & Privacy, vol. 12, no. 3, pp. 72-75, 2014.

[11] A. A. Lin, Y. N. Liu, and Y. H. Hu, "Memory Analysis and Reconstruction Techniques for Virtual Machine Forensics," in Proceedings of the 2012 International Symposium on Information Technologies in Medicine and Education, pp. 24-27, 2012.

[12] D. Rahevar, "Study on Live analysis of Windows Physical Memory," Journal of Computer Engineering (IOSR-JCE), vol. 15, no. 4, pp. 76-80, 2013.

[13] R. Yang, J.-c. Ren, S. Bai, and T. Tang, "A Digital Forensic Framework for Cloud Based on VMI," in "2nd International Conference on Computer Science and Technology (CST 2017)," 2017, ISBN: 978-1-60595-461-5

[14] N. Maurya, J. Awasti, R. P. Singh, and A. Vaish, "Analysis of Open Source and Proprietary Source Digital Forensic Tools," International Journal of Advanced Engineering and Global Technology, vol. 3, no. 7, pp. 916-922, 2015.

[15] E. Casey, "Digital evidence and computer crime: Forensic science, computers, and the internet," 3rd ed., Academic Press, 2011.

[16] E. Casey, "Digital Evidence and Computer Crime," Academic Press, 2014.

[17] R. Pal, "Memory Forensics in Digital Forensics," International Journal of Computer Science and Information Security, vol. 15, no. 2, pp. 180-188, 2017.

[18] NIST, "Guidelines on Electronic Evidence Collection and Preservation," NIST Special Publication, 80-86, 2014.

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

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