Advances in Artificial Intelligence and Machine Learning; Research 3 (4) 1654-1670 Received 12-09-2023; Accepted 20-11-2023; Published 27-11-2023

Blockchain Malware Detection Tool Based on Signature Technique

Siti Husna Abdul Rahman siti.husna@mmu.edu.my
Faculty of Computing Informatics, Multimedia University, Cyberjaya, Selangor, Malaysia

Chastan Nevin Gabriel chastangabriel@gmail.com
Faculty of Computing Informatics, Multimedia University, Cyberjaya, Selangor, Malaysia

Su-Cheng Haw sucheng@mmu.edu.my
Faculty of Computing Informatics, Multimedia University, Cyberjaya, Selangor, Malaysia

Ahmad Anwar Zainuddin anwarzain@iium.edu.my
Department of Computer Science, KICT, International Islamic University Malaysia

Corresponding Author: Siti Husna Abdul Rahman

Abstract

Downloading software or files from the internet can be risky as it is hard to know if they are safe and do not contain viruses. Traditional anti-virus software uses a centralized database to identify malware, but this method has drawbacks due to its centralized design, which creates a single point of failure. Blockchain technology has become a solution to many problems faced in the tech industry, including the need for a decentralized and secure way for users to verify and confirm the presence of malware in a file. The decentralized database, permanence, immutability, anonymity, and auditability of blockchain technology make it an ideal solution for malware detection. In fact, malware data has been compiled in databases that antivirus manufacturers use to identify malware. However, blockchain technology provides a more secure and decentralized way to store this data, which can be shared between users and allow them to rapidly update whether a file is safe or not. This paper presents a blockchain-based malware detection tool designed to enhance security and prevent the spread of malware in digital networks. The tool based on Java programming language incorporates signature-based methods to effectively identify and detect malicious codes in malware. The proposed tool contributes to the field of cybersecurity by leveraging blockchain technology to enhance malware detection process.

Keywords: Blockchain, Malware detection, Signature-based detection, Decentralization, Malware.

Citation: Siti Husna Abdul Rahman et al. Blockchain Malware Detection Tool Based on Signature Technique. Advances in Artificial Intelligence and Machine Learning. 2023;3(4):94.

1. INTRODUCTION

Malware, short for malicious software, is any program or file that is intentionally harmful to a computer, network, or server. Malware can take many forms, including viruses, worms, trojan horses, ransomware, and spyware. Malware can get into a computer through a variety of mechanisms, most of which involve exploiting a combination of human and technical factors[1]. As malware has evolved from its beginnings as demonstrations of skill by individual programmers to sophisticated technologies developed by organized crime, the boundaries between the different categories are beginning to be uncertain. The goal of malware is to cause havoc and steal information or resources for monetary gain or sheer sabotage intent. In this context, it is important to understand the different types of malwares and how they can affect computer systems.

The creation of a malware called Shamoon is the perfect example of how serious the issue of malware is. The Shamoon malware targeted the multi-trillion-dollar company Saudi Aramco which is one of the top oil companies in the world. The attack left immense damage destroying 80% of their workstations and servers. This forced Aramco to increase their security[2]. Given the potential threat posed by targeting essential services such as medical devices, particularly those connected to the Internet of Things (IoT), the implications are significant. The attacker could gain remote control of these devices and threaten a patient's life [3]. It is events like these that are bringing together developers and cybersecurity specialist around the world to create a more secure solutions to detect and prevent malware attacks.

Traditionally, database management systems such as SQL or NoSQL systems have been used to store data, but they are prone to single-point-failure, which can corrupt data and render the database system invalid. However, blockchain technology provides a more robust data storage system by having a peer-to-peer database system. This project aims to fight against the mass distribution of malware signatures and detect their presence in submitted files. To detect malware, there have been many attempts, such as anti-virus software that use signature-based, behavior-based, and heuristic-based detection, each with its own algorithm [4]. Distribution of malware has become something so easy, and this is disrupting many aspects of cybersecurity and thus damaging assets. Some reports have said that around 1 million malware files are produced and known to the security community [5].

This project aims to detect malware attacks using a decentralized approach that utilizes blockchain technology. By updating the blockchain with the new malicious file's signature, the distributed and decentralized feature of blockchain is used to detect malware attacks. The proposed blockchain-based malware detection method uses shared signatures of suspected malware files. The approach aims to provide a more secure and robust way of detecting malware attacks. Blockchain technology provides a more secure way of storing malware signatures through its peer-to-peer connection as it provides many benefits in the form of a decentralized database. The decentralized nature of blockchain could be a key factor in this detection tool as it allows for a stable and secure database that can't be tampered with from the outside. Having said that, this project is a reliable and secure signature-based malware detection tool that is able to detect malware in files and add new malware signatured to the database if new variants are detected. This paper is organized as follows: section 2 literature review, section 3 methodology, section 4 result, and discussion finally section 5 concludes.

2. LITERATURE REVIEW

In a nutshell, malware is a compilation of code sometimes known as malicious software. It is hostile and frequently used to damage or abuse a system. Depending on the malware's design aim and the network configuration, introducing malware into a computer network environment can have a variety of effects. Malicious files in computer systems are bypassing malware detection and prevention mechanisms as malware becomes more complicated and prevalent [1]. Malware doesn't only affect companies but also consumer devices. For example, there was ransomware attack happened during the COVID-19 pandemic, the attack disrupted the operations of the major fuel supply chain in 17 states including the Washington DC [7]. The virus could send sensitive documents to people in the victims address book without the user's knowledge [8], According to [9], there are 12 main types of malwares.

- Ransomware

Ransomware is a type of malware that encrypts the files of the targeted user or organization, making them inaccessible. The attacker then demands a ransom payment in exchange for the decryption key to unlock the files. Ransomware can target personal PC users and big organizations alike. To be successful, ransomware needs to gain access to a target system, encrypt the files there, and demand a ransom from the victim. Ransomware can gain access to a system through various infection vectors, such as malicious or compromised websites, attachments from spammed emails, or dropped by exploit kits onto vulnerable systems. Once executed in the system, ransomware can either lock the computer screen or encrypt the files. The costs of ransom payments can range from a few hundred dollars to thousands, payable to cybercriminals in Bitcoin.

- Fileless Malware

Fileless malware is a type of malware that doesn't try to install anything but rather edits existing system files which allows itself to creep around without being detected by an antivirus making this type of malware very dangerous. This allows the malware to work its way to make the operating system more and more vulnerable, eventually making a backdoor for the attacker to implant more malicious software.

- Spyware

Spyware works on gathering information such as passwords, payments details, messages etc. rom the targeted user. Allowing the attacker to sell or use this information to attack on a bigger scale.

- Adware

Adware is somewhat similar to Spyware in the sense that it collects information from the user without their knowledge. The data collected can also be about who their friends with thus increasing the target area. This information can be used by the attacker or be sold to other advertisers.

- Trojans

A Trojan can pose as other relevant or desired software such as games, tools or even as innocent as an email in order to gain the users attention and thus get downloaded. Once it has been downloaded, the trojan gains access and control of the victim's system.

- Worms

Worms are more of a network target malware, and they may gain access through backdoors in software applications, unintentional software vulnerabilities or through flash drives being connected to the system. Once installed, they can easily launch DDoS attacks, steal data, or conduct ransomware attacks.

- Virus

A virus is a section of code that is attached into an application and is executed when the app runs. This makes the system vulnerable and allows for various attacks to take place.

- Rootkits

Rootkit can be attached to applications, kernels, hypervisors, or firmware. They give the attacker remote access and full administrative privileges to the victim's device. They can spread through many ways such as email attachments, phishing sites, and malicious downloads.

- Keyloggers

Keyloggers have actual good uses for it. Some businesses use them to monitor their employees' activities and parents could also use it to monitor their children's online activities. But on the other hand, Keyloggers can also be used as Spyware to monitor users' activity without their knowledge.

- Bots

Just like keyloggers, they have good applications in the industry like allowing for automated tasks to be executed on command. But there are certain malicious applications for it too that can connect back to a central server. They can be used in large numbers to create a botnet (A network malware infected computers buy a bot herder, the person who controls the botnet infrastructure) that can be used to launch broad remotely controlled flood attacks such as a DDoS attack.

- Wiper Malware

Lastly, Wiper malware has the single purpose of wiping user data and ensuring that it cannot be recovered. These are mostly used to cover traces of intrusion by hackers that was previously carried out on company networks.

2.1 Malware Detection

Malware detection is the process of identifying and preventing malicious software from harming computer systems and networks. Malware detection techniques can be divided into three broad categories: signature-based, heuristic-based, and specification-based.

1. Signature-based Detection: Using a database, antivirus software employs signature-based detection. Any object's or file's signature can be created utilizing a few of its distinctive characteristics. There are algorithms that can create a file's signature. A malicious signature that matches any of the signatures in the database is looked for in the file [10]. The file is deemed malicious if its signature matches one of the already existing signatures or if it is a derivative of one. The process for signature-based detection involves the following steps:

- A new type of malware is discovered.
- The malware's signature which is generated based on its unique attributes is added to the virus database.
- The antivirus product is updated to include the updated database.
- The antivirus software will then be able to find the malware during scans by searching for its footprint.

The disadvantages of signature-based detection include the possibility of zero-day attacks, the ability for novel malware to go undetected, the time needed to process incoming information against the signature database, the potential for DDOS attacks, and the potential storage requirements of the signature database. There are several types of signature-based detection methods, which are commonly used to detect malware and other malicious codes. Here are some of the most common types of signature-based detection methods:

- String-based Scanning: This method compares the extracted sequence of bytes stored in the database with the actual sequence of bytes from a file[11]. So, the only way for the file to be passed as non-malicious is if there isn't even one single matching sequence in the file when it is compared to the sequences in the database[5].
- Generic Detection Signature: This type of detection focuses more on grouping malware signatures. There have been studies that show some malware signatures are like other which makes is easier to group them up making it possible to compile malware signatures by finding common areas in their code to group them. So, when searching files for malware, the application only needs to search for similar variants of malware codes [5].
- Top & Tail Detection: This method is more an accelerated way of searching files for malware. This method only searches the first and last 2KB of a file for malware. This is because malware usually attaches itself to the beginning and end of a file [5].
- Entry-Point Detection: This method scans for malware at the entry-point when the application is executed, this allows for malware to be detected and also speeds up the process [5].

- Integrity Checking: This method checks the integrity of a file by creating hash files like MD5, SHA-1, and SHA-256 [5]. When the integrity checks the database to confirm if a file has been modified. But this method does have some disadvantages as it does tend to produce False Positives.

2. Behavior-based detection: These techniques identify malware by watching a program's (or executable's) behavior and predicting what it will do even before it is executed based on that behavior. They attempt to observe the program's behavior before determining whether it is malicious [12]. These techniques fix the problems with signature-based ones. Multiple files can all be categorized as one behavior-signature when they exhibit the same behavior when checked for malware in an isolated environment. These kinds of detection techniques are very helpful for finding malware that continuously produces new variations (like polymorphic malware), as most variants share a common pattern of resource and service usage. Tools like debuggers, simulators, emulators, and sandboxes are frequently used to analyze malware based on its behavior in a controlled environment. In this form of detection, a file is watched for its actions, such as system calls and other programs it may call, as well as any potential harmful behavior, such as the kinds of files it has accessed and the connections it has made. However, even behavior-based detection techniques can be avoided due to the increasing incorporation of more advanced evasion and antidetection tactics into new malware, such as "environmental awareness." The main drawbacks of behavioral approaches are the large amount of scanning time needed and high false-positive ratio (FPR).

3. Heuristic-based detection: Provides a remedy for the drawbacks of both signature- and behavior-based approaches to detection. Data mining and machine learning techniques are used in heuristic malware detection approaches to gather information about the behavior of the file that needs to be screened for maliciousness. Theoretically, a machine learning model should be able to reach the same conclusions as behavior-based detection approaches more quickly. Below are some scenarios where heuristic-based detection may be beneficial [3].

- Defense against brand-new, mutant malware threats.
- Finding specifically targeted malware attacks.
- Determining how malware behaves when files are opened in a particular setting.
- Data gathering regarding malware behavior.

Heuristic-based malware detection has the drawback of having a high false positive percentage.

2.2 Blockchain

Blockchain technology has gained a lot of fame thanks to the cryptocurrency industry, particularly because of Bitcoin. A cryptocurrency is a type of digital currency that, as its name suggests, uses cryptography to protect user and financial information during transactions [13, 19]. Satoshi Nakamoto was the one who created cryptocurrency known as Bitcoin in 2008 beginning with just a piece of paper, and he then implemented it in 2009. Following its decentralized design [14],

it is a network with a collection of computers that communicate with one another using peer-to-peer technology and share a digital ledger, which is a database that is accessible to numerous machines throughout the network. The term "blockchain" refers to a series of blocks, each of which includes some data and is connected to the one before it like a chain [4]. Once a blockchain is constructed, each block will have a parent unless it's the first one in the chain, which is called the genesis block [15]. Furthermore, each block will have a header and a body. The header will contain information such as the timestamp, nonce and previous block hash [16]. The body will hold such as (e.g. transaction) with a maximum size. The larger the entire block size, the more storage it would require, which could slow down network transmission [15]. In that regard, a node typically aids in the consensus process, which gives each member the opportunity to maintain, approve, and update the network, to verify the data and add it to the blockchain [13]. Additionally, the three types of architecture for the blockchain system are as follows:

- Public Blockchain: In this kind of system, anyone can take part in the consensus and all records are open to the public [17].
- Private Blockchain: As the name suggests, this type of blockchain is private since only certain individuals from the same organization can participate in the consensus[15, 17].
- Consortium Blockchain: In this approach, running a node and participating in the consensus process are only permitted by a certain community of N peers [15, 17].

Due to the increasing number of new malwares in recent years, blockchain technology is gaining attention from the research community. This is because blockchain has features such as decentralized, distributed, immutable, and tamper-proof that make it useful for detecting malware. In this system, blockchain is used to update the new malicious file's signature. Based on the peer-to-peer network, every node in the network will be able to detect if the file contains malicious information or not. Unlike the traditional antivirus system which uses database of known virus and require time to update the signature. The study in [4] proposed the used of blockchain as a malware detection, in this system, every node will be updated with the new signature with the help of distributed networks in blockchain, however since the work still on proposal stage, no actual experiment is done. In [18] discussed the used of blockchain technology on sharing signature of suspected malware.

3. METHODOLOGY

This malware detection tool developed with the use of JAVA programming language, thus making it easy to export and use on many different computers. The Java Virtual Machine (JVM) also adds flexibility and portability so that the software can run on several systems. Given that it also offers many other features, like threading, Java is a solid choice for blockchain development. Figures below contain the attributes and variables in their respective classes. This application runs on the NetBeans IDE on a windows machine. The application uses the IDE to run and compile the code which shows the graphical user interface for the user to interact with the application.

The type of blockchain used in this application is a private permissionless one. This allows for an application that is highly scalable. The application can differentiate between files with the "malware

signature" and files without it. Below is the full implementation breakdown of the Blockchain Based Malware Detection Tool.

Class 1: Blockchain

Blockchain class includes functionalities to generate block numbers and block addresses. Additionally, it uses a StringBuilder to create the malware record and writes the blockchain data to a file. Overall, this implementation provides a basic blockchain functionality for storing malware records. The blockchain file maintains a chronological order of the added records, and each record includes the block number, signature hash, and path hash. The class allows for adding new records, counting the existing records, and retrieving the entire blockchain for further analysis or display (see FIGURE 1).

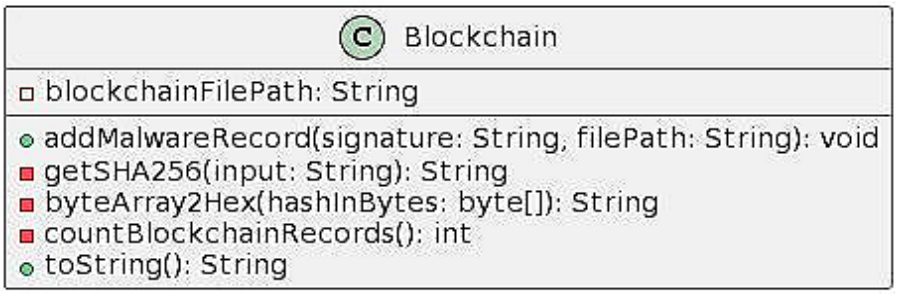

Figure 1: Blockchain class.

Class 2: MalwareDetectorApp

This class represents the entry point of the malware detection application. It contains the main method that initializes the graphical user interface (GUI) of the application. Overall, this class serves as the entry point for the malware detection application. It initializes the GUI and allows users to interact with the application through the graphical interface. FIGURE 2 depicts the MalwareDectectorApp.

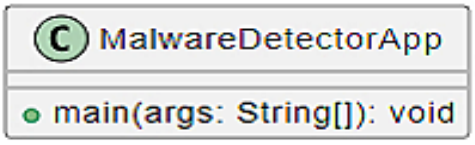

Figure 2: MalwareDetectorApp class.

Class 3: FileScanner

This class represents a file scanner used in the malware detection application. It provides methods for performing integrity checks and top-and-tail scans on files, as well as removing detected malware from a file. Overall, this class provides functionality to scan files for malware by performing integrity checks and top-and-tail scans. It also allows for removing detected malware from the current file. The class interacts with the "signatureManager" object, which manages the collection of malware signatures as depicted in FIGURE 3.

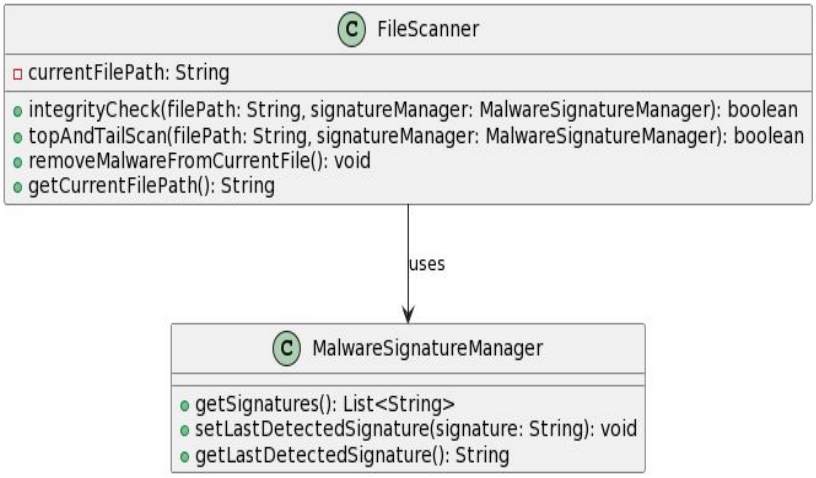

Figure 3: FileScanner class.

Class 4: FileUtil

FIGURE 4 depicts the FileUtil class. This class provides utility methods for file handling and cryptographic operations used in the malware detection application. Overall, this class provides convenient utility methods for reading text files and calculating hashes. These methods can be used by other classes in the malware detection application for various file operations and cryptographic tasks.

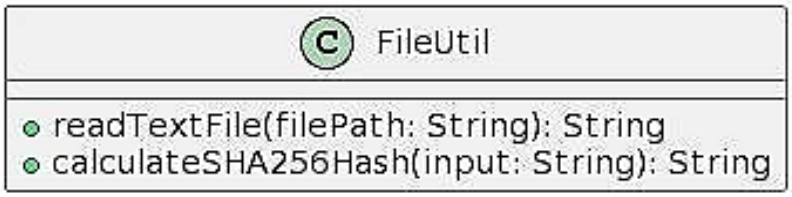

Figure 4: FileUtil Class.

Class 5: MalwareSignatureManager

FIGURE 5 shows the MalwareSignatureManager class. This class manages the collection of malware signatures used in the malware detection application. It provides methods for adding signatures, retrieving signatures, and storing the last detected signature. Here are the implementation details and the behavior of the program. Overall, this class provides functionality for managing malware signatures. It allows for adding signatures, retrieving them, and storing the last detected signature. The class ensures that duplicate signatures are not stored in the collection by using a HashSet, which maintains unique values.

Class 6: MalwareDetectorGUI

This class represents the graphical user interface (GUI) of the malware detection application (see FIGURE 6). It provides a main window with options to add malware signatures and check for mal-

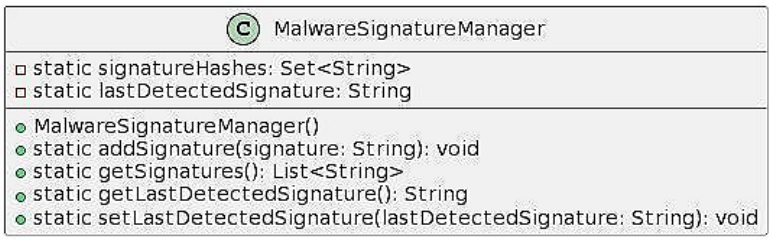

Figure 5: MalwareSignatureManager class.

ware in files. Overall, this class provides the GUI for the malware detection application. It allows users to add malware signatures, check files for malware, and handle the detection results. The class also includes methods for data encryption and decryption, as well as loading signatures from a file.

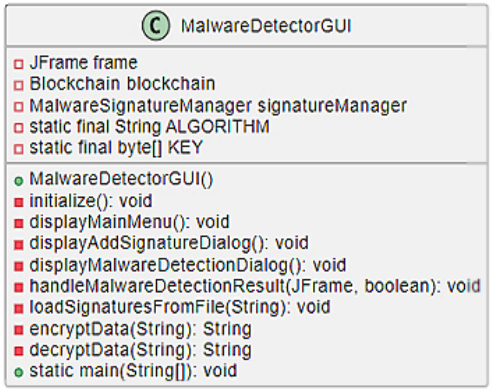

Figure 6: MalwareDetectorGUI class.

The experiment was carried out based on the flow chart in FIGURE 7. The flow chart shows information about the process from submitting the file till the end. The sequence diagram also shows how the application work.

FIGURE 8 shows the processed carried out in the MalwareDetectorGUI class. It shows the whole GUI process of accessing the blockchain if needed, talking to the MalwareSignatureManager class when sending a new signature or wanting to get existing signatures for detection. It makes it easier for the user to interact with the application by providing the user with a graphical user interface with the use of Javax.

FIGURE 9 shows the sequence of processes that occur in when the MalwareDetectorApp class adds the malware record to the blockchain.

FIGURE 10 shows the sequence of processes that occur when the FileScanner class wants to get signatures to carry out the malware detection process while FIGURE 11, shows the sequence of processes that occur when the FileUtil class reads the path and builds a new hash address.

FIGURE 12 shows the sequence of processes that occur when the adds or wants to access signatures.

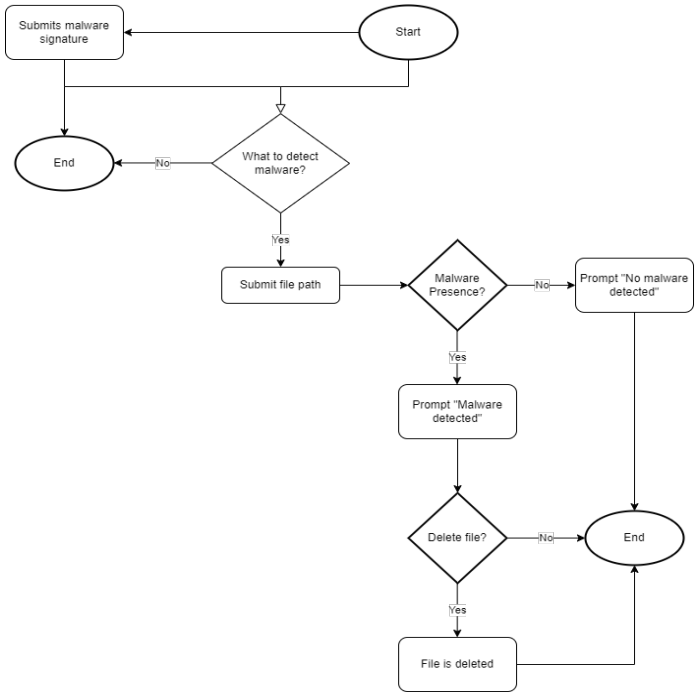

Figure 7: Flow of Process and if-else Statements.

4. RESULTS AND DISCUSSION

One of the aims of this research work is to show the ability of using blockchain technology as a malware detection tool. For this purpose, GUI is developed to prove the working system. This is the home page of the application as shown in FIGURE 13. It allows you choose between "Add malware Signature" and Check for Malware.

This popup message shown in FIGURE 14, allows the user to submit a malware signature. The signature can range from a collection of numbers and alphabets.

Once the signature is submitted, the user will be prompted with a message saying it was successfully submitted as shown in FIGURE 15. The signature is then added to the blockchain.

With the signature now added, users can now submit any text file by pasting the path and name of the file in the given text field as shown in FIGURE 16. This will then be passed to the fileutil class for the next process.

If a file contains a malware signature which is determined by the FileScanner class. It is then passed to the removeMalwareFromCurrentFile in the FileScanner class to be handled. Thus, deleting that section of the code or deleting the file itself as shown in FIGURE 17.

When the user chooses the "Delete file" option, this prompt is then triggered when the file is deleted as shown in FIGURE 18.

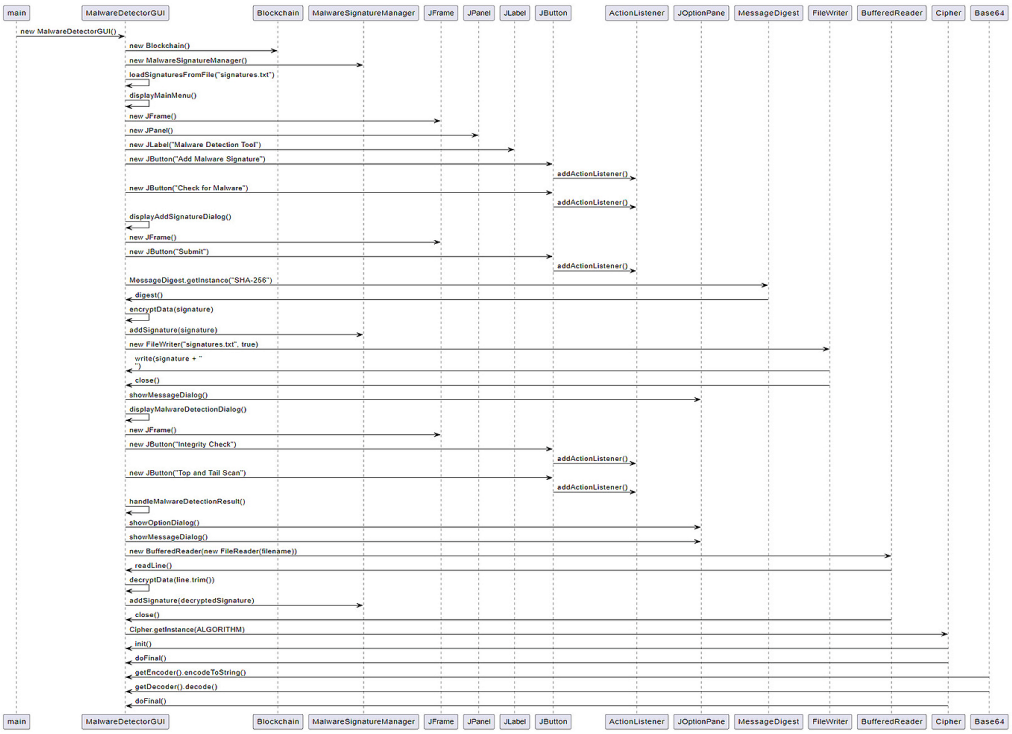

Figure 8: Sequence Diagram for MalwareDetectorGUI Class.

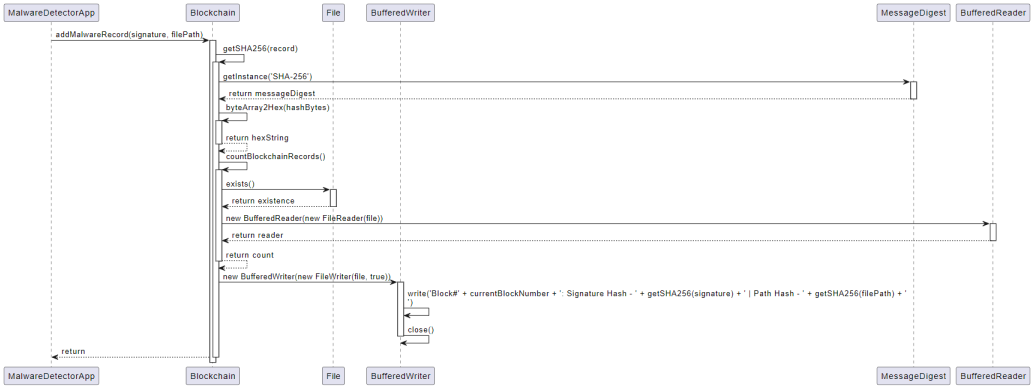

Figure 9: Sequence Diagram for the Blockchain Class.

Despite extensive research, detecting malware remains a significant challenge that affects everyone. The consequences of malware can range from a simple data breach to the loss of millions of currencies. The development of a blockchain based malware detection tool aims to demonstrate the potential of blockchain technology in the field of anti-malware. The blockchain offers several features, such as a decentralized database, persistence, immutability, anonymity, and auditability, that could address many of the issues associated with traditional malware detection methods. Although blockchain technology is still in its early stages, it holds immense potential. The results of

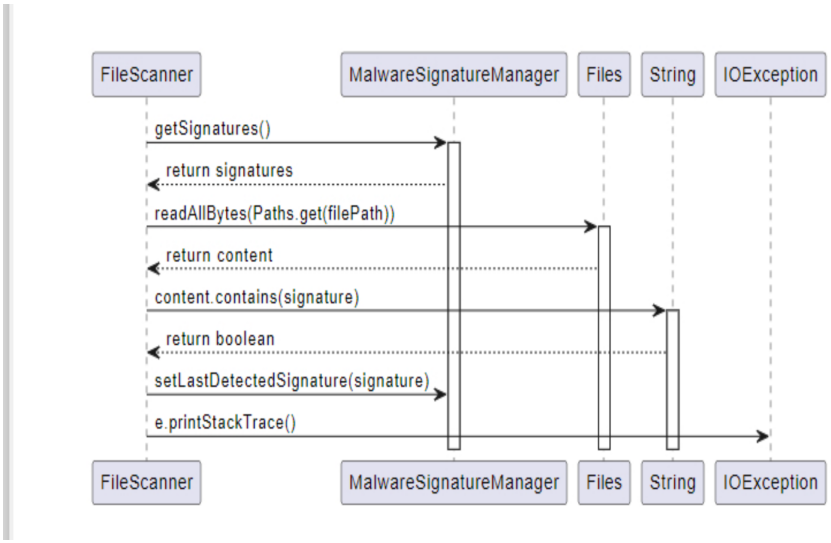

Figure 10: Sequence Diagram for the FileScanner Class.

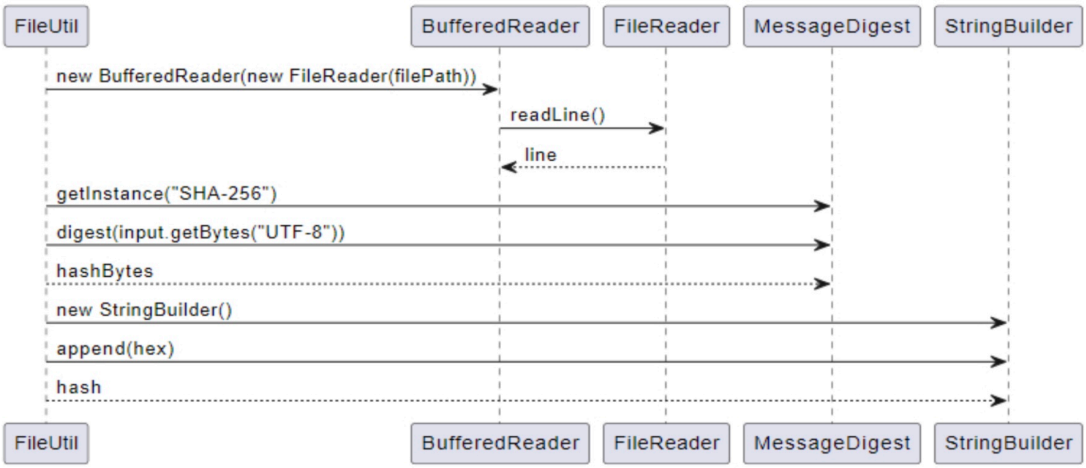

Figure 11: Sequence Diagram for the FileUtil Class.

the analysis and testing of this application could generate interest in its implementation in the field of cybersecurity.

Not forgetting the signature scanning part of this development. The integrity malware detection focuses on identifying unauthorized modifications or tampering of critical files and system components. By verifying the integrity of these elements, it helps detect and prevent malware attacks that aim to alter or compromise system integrity. Integrity malware detection provides an additional layer of defense, ensuring the trustworthiness and reliability of critical system components.

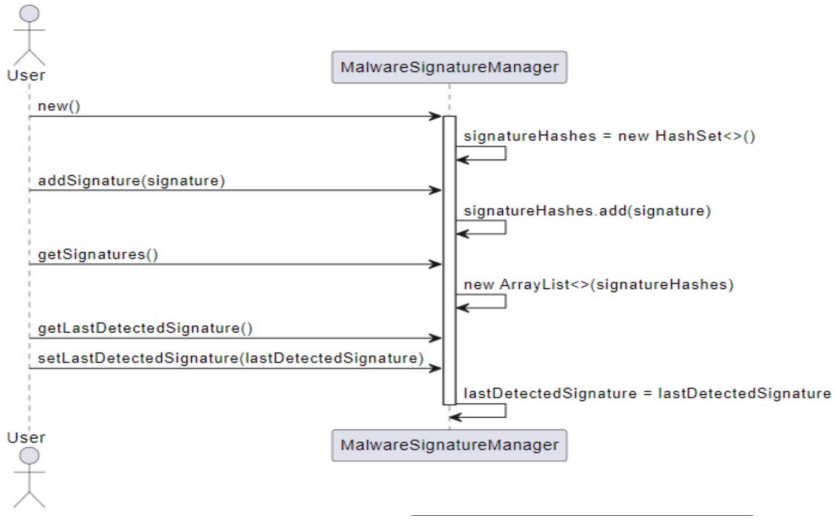

Figure 12: Sequence Diagram for the MalwareSignatureManager Class.

Figure 13: Home page of GUI.

Figure 14: Add Malware Signature.

On the other hand, top-and-tail malware detection examines the characteristics of file headers and footers. By analyzing these specific portions of files, it can identify malware that has been appended or embedded within legitimate files. This approach is effective in detecting malware that tries to disguise itself by hiding within legitimate files or using file manipulation techniques. Top-and-tail malware detection helps improve the accuracy of malware detection systems by focusing on specific file regions that are commonly targeted by malware. The combination of integrity malware detection

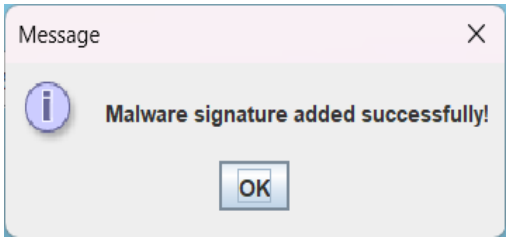

Figure 15: Signature Added.

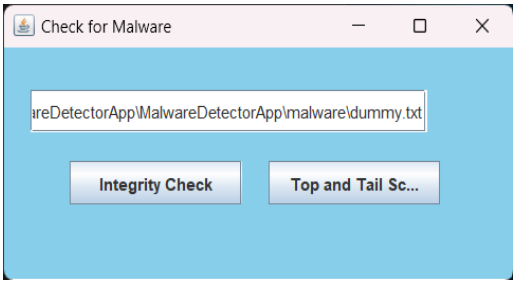

Figure 16: Check for malware.

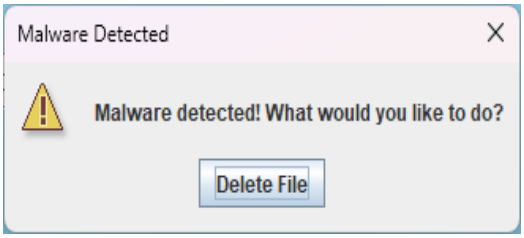

Figure 17: Malware Detected.

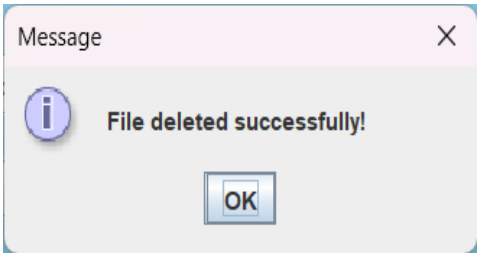

Figure 18: File deleted successfully.

and top-and-tail malware detection techniques provides a comprehensive and robust defense against various types of malwares. By leveraging both techniques, organizations can enhance their malware detection capabilities and mitigate the risks associated with advanced and evasive malware threats.

5. CONCLUSION AND FUTURE WORK

In conclusion, the proposed blockchain-based malware detection tool based on signature technique is a promising solution to the problem of detecting malicious software. By using blockchain technology, the tool can provide a secure and decentralized way of detecting malware, which can be more effective than traditional centralized methods. The tool can also be used to detect new and unknown malware samples, which is a significant advantage over traditional signature-based detection methods. The proposed tool can be further improved by incorporating other detection techniques, such as behavioral-based detection, and by using a consortium blockchain to improve the accuracy of the detection process. Overall, the proposed tool has the potential to significantly improve the security of computer systems and protect against the growing threat of malware attacks.

Improvements can be made to the blockchain based malware detection tool, starting with the type of blockchain used. The current application uses a public permissionless blockchain, which is easy to implement and scalable but comes with significant security risks. Implementing a public permission or hybrid blockchain would be a good addition to the application. Another feature that could be added is the ability to scan files other than text files to provide a more comprehensive and thorough coverage of the tool. Lastly, a more comprehensive and better-looking graphical user interface should be developed to make the application more inviting and user-friendly.

References

[1] Namanya AP, Cullen A, Awan IU, Disso JP. The World of Malware: An Overview. In: 6th International Conference on Future Internet of Things and Cloud (FiCloud). IEEE. 2018:420-427.

[2] https://malwareindepth.com/shamoon-2012/

[3] Wazid M, Das AK, Rodrigues JJ, Shetty S, Park Y. IOMT Malware Detection Approaches: Analysis and Research Challenges. IEEE Access. 2019;7:182459-182476.

[4] Pichikala SM, Rachana G, Sanjanapatel H, Shanu S, Vineeth N. Malware Detection Using Blockchain Technology 2nd International Conference for Emerging Technology, INCET. 2021:1-4.

[5] Aslan O, Samet R. A Comprehensive Review on Malware Detection Approaches. IEEE Access. 2020;8:6249-6271.

[6] Chowdhury MJM, Colman A, Kabir MA, Han J, Sarda P. Blockchain Versus Database: A Critical Analysis. In: 17th Ieee International Conference on Trust, Security and Privacy in Computing and Communications. IEEE Publications. 2018:1348-1353.

[7] Alqahtani A, Sheldon FT. A Survey of Crypto Ransomware Attack Detection Methodologies: An Evolving Outlook. Sensors. 2022;22:1837.

[8] Garber L. Melissa Virus Creates a New Type of Threat. Computer. Aug 1999;32:16-19.

[9] Pachhala N, Jothilakshmi S, Battula BP. A comprehensive survey on identification of malware types and malware classification using machine learning techniques. In: 2nd International

Conference on Smart Electronics and Communication (ICOSEC). IEEE Publications. 2021;1207-1214.

[10] https://informationsecurity.report/Resources/Whitepapers/920fbb41-8dc9-4053-bd01-72f961db24d9_ICIT-Analysis-Signature-Based-Malware-Detection-is-Dead.pdf

[11] https://www.researchgate.net/publication/350722779_Robust_Static_Analysis_of_Portable_Executable_Malware

[12] Jacob G, Debar H, Filiol E. Behavioral Detection of Malware: From a Survey Towards an Established Taxonomy. J Comput Virol. 2008;4:251-266.

[13] Elrom E. The Blockchain Developer. Apress. 2019.

[14] Raje S, Vaderia S, Wilson N, Panigrahi R. Decentralised Firewall for Malware Detection International Conference on Advances in Computing, Communication and Control. 2017:1-5.

[15] Zheng Z, Xie S, Dai H, Chen X, Wang H. An Overview of Blockchain Technology: Architecture, Consensus, and Future Trends. In: IEEE International Congress on Big Data. IEEE. 2017:557-564.

[16] Wylde V, Rawindaran N, Lawrence J, Balasubramanian R, Prakash E, et al. Cybersecurity, Data Privacy and Blockchain: A Review. SN Comput Sci. 2022;3:127.

[17] Gu J, Sun B, Du X, Wang J, Zhuang Y, Wang Z. Consortium Blockchain-Based Malware Detection in Mobile Devices. IEEE Access. 2018;6:12118-121128.

[18] Fuji R et al. Investigation on Sharing Signatures of Suspected Malware Files Using Blockchain Technology. In International Multi Conference of Engineers and Computer Scientists. 2019:94-99.

[19] Lai JF, Heng SH. Secure File Storage on Cloud Using Hybrid Cryptography. Journal of Informatics and Web Engineering. 2022;1:1-18.