UNIVERSITY OF
WEST LONDON

Dissertation Proposal

Blockchain-and ML based Malware Detection and Integrity Checking: A Decentralized Approach

Supervisor: Dr. Waqar Asif

Student Name: Araharan Loganayagam

University ID: 21524785

School of Computing and Engineering
University of West London
2023

Table of Contents

Abstract

This proposal aims to address the need for robust malware detection and integrity checking techniques by leveraging the combined power of blockchain technology and machine learning algorithms. The abstract provides an overview of the background and motivation driving this project, followed by a comprehensive literature review of existing techniques and their limitations. The proposal briefly outlines the problem at hand, emphasizing the importance of effective malware detection and integrity checking in ensuring system and data security. The aim of the project is then articulated, along with specific objectives to guide the research. The proposed solution entails developing a decentralized approach that harnesses the potential benefits of blockchain technology and the adaptability of machine learning algorithms. By leveraging the decentralized nature of blockchain, the proposed solution offers enhanced security and integrity in malware detection. Machine learning algorithms will be utilized to improve the system's ability to detect and respond to emerging threats. Furthermore, a detailed schedule for project completion is outlined, along with the necessary resources required to successfully execute the research. Overall, this proposal seeks to contribute to the advancement of malware detection and integrity checking techniques, paving the way for more robust and resilient systems in the face of evolving cyber threats.

1 Background and Motivation

Malware poses a significant threat to the security and integrity of computer systems and data, necessitating the development of robust detection and integrity checking techniques. malware attacks have become increasingly prevalent, sophisticated and pose a significant threat to the security of computer systems and networks. To address this challenge, researchers have developed various malware detection and integrity checking techniques, ranging from traditional signature-based methods to more advanced machine learning and blockchain-based approaches. Traditional signature-based approaches have limitations in detecting sophisticated and evolving malware variants. Consequently, there is a growing body of research exploring innovative methods to enhance malware detection and integrity checking capabilities. While these techniques have been successful in detecting and preventing malware attacks, the ever-evolving nature of malware requires continued research and innovation to stay ahead of threats. This study is focusing on the current state of malware detection and integrity checking techniques, specifically exploring the potential of a decentralized approach based on blockchain and machine learning.

Previous studies have demonstrated the potential of blockchain technology in various security applications, including malware detection. Blockchain's decentralized and immutable nature offers the possibility of creating a secure and tamper-resistant platform for detecting and mitigating malware attacks. For instance, research by Zhaoyang et al. (2019) proposed a blockchain-based architecture for malware detection, leveraging the distributed consensus mechanism to validate the integrity of system files and detect potential malware tampering.

Furthermore, machine learning algorithms have shown promise in improving malware detection accuracy and adaptability. Researchers such as Kolosnjaji et al. (2018) have explored the application of deep learning models for detecting malware through analysing behavioural patterns and feature extraction. Similarly, studies by Raff et al. (2018) and Saxe et al. (2019) have demonstrated the effectiveness of machine learning techniques, such as support vector machines and convolutional neural networks, in detecting malware samples and classifying them into different families.

The limitations and challenges faced by existing approaches highlight the need for an integrated solution that combines the strengths of blockchain technology and machine learning algorithms. By leveraging the decentralized nature of blockchain and the adaptability of machine learning, it is possible to develop a decentralized approach for malware detection and integrity checking that is more resilient to evolving threats and offers improved accuracy and reliability.

The motivation behind this project is to contribute to the advancement of malware detection and integrity checking techniques by proposing a decentralized approach that harnesses the benefits of blockchain technology and machine learning. By addressing the limitations of existing methods, such as the reliance on centralized authorities and the need for frequent updates, this research aims to enhance the overall security and integrity of computer systems. By leveraging previous research in blockchain-based security architectures (Zhaoyang et al., 2019) and the application of machine learning algorithms for malware detection (Kolosnjaji et al., 2018; Raff et al., 2018; Saxe et al., 2019), this project seeks to develop a novel solution that can adapt to emerging (Zhaoyang Chi, 2019) (Marko Kolosnjaji, 2018) (Kyle Soska, 2015) (Saxe, 2019) malware threats while ensuring data integrity and system security.

By decentralizing malware detection and integrity checking, blockchain technology can provide a transparent, secure, and tamperproof environment for detecting and preventing malware attacks. Machine learning can also enhance the accuracy and effectiveness of malware detection by analysing large amounts of data and identifying patterns that may not be visible to traditional signature-based methods. The traditional signature-based methods, anomaly-based detection, and machine learning-based methods. It will then explore how blockchain and machine learning can be used together to create a decentralized approach for malware detection and integrity checking. Finally, the review will analyse the potential benefits of a decentralized approach, including increased transparency, security, and efficiency.

2 Literature Review

There have been so many approaches in malware detection and integrity checking components of modern cybersecurity systems. One popular approach is utilizing machine learning for malware detection. Kang et al. (2019) introduced a system that employs Convolutional Neural Networks (CNNs) to analyze network traffic flow data and identify potential malware. The system captures real-time network traffic data from diverse sources like routers, switches, and firewalls, which is then processed and transformed to extract pertinent features for malware detection. By training the CNN model on a labeled dataset of network traffic flow data, the model learns to recognize patterns associated with malicious traffic. Subsequently, the trained CNN model can be deployed to identify malware in real-time network traffic flow. However, the paper does have certain limitations. One of these limitations pertains to the utilization of a restricted set of datasets for both training and evaluation. The effectiveness of the proposed system might be influenced by the diversity and representativeness of the datasets employed. Therefore, incorporating a broader range of datasets would enhance the generalizability of the findings and improve the overall robustness of the system.Pejman et al. (2021) introduced an innovative method for detecting malware by employing natural language processing, entity behavior analytics, and machine learning. The approach involves extracting relevant features from system call traces using NLP techniques and analyzing them with EBA to identify anomalies in system entity behavior that may indicate the presence of malware. However, the paper has certain limitations that should be addressed. Firstly, the evaluation and validation of the proposed approach may be limited. Although the authors describe the methodology and techniques utilized, a comprehensive assessment of the approach's performance and effectiveness may be lacking. It is essential to conduct extensive evaluation using appropriate metrics and benchmarks to determine the reliability and generalizability of the malware detection system. Secondly, the scalability of the proposed approach may not be adequately addressed. Processing large-scale system call traces and applying NLP techniques and EBA for analysis can pose computational challenges. It is important to consider the scalability of the approach to ensure its feasibility and efficiency in handling substantial volumes of data. By addressing these limitations, the proposed method can be further enhanced and its applicability in real-world scenarios can be better understood. In contrast, Priya et al. (2023) offers a comprehensive survey of recent advancements in malware classification and detection through the utilization of transfer learning. The paper emphasizes the importance of various techniques such as malware feature extraction, transfer learning, deep learning, ensemble methods, and feature selection in enhancing the performance of malware detection models. However, there are certain limitations that should be taken into account. Firstly, the paper may not sufficiently address the scalability and computational requirements associated with the proposed techniques. Given the substantial amount of data involved in malware detection, it is crucial to consider the scalability of the discussed methods. The authors should provide insights into how these techniques can effectively handle large-scale datasets without compromising detection accuracy. Additionally, the paper may not thoroughly explore the limitations and challenges inherent in applying transfer learning to malware detection. Transfer learning relies on pre-existing models and assumes the availability of relevant and representative source domains. However, the extent to which transfer learning techniques can generalize to diverse malware types and unseen data remains a challenge that should be acknowledged and effectively addressed. By addressing these limitations, the proposed techniques can be further refined and their applicability in real-world scenarios can be better understood. Sanjeev et al. (2016) introduces a technical framework aimed at providing real-time protection against malware through the implementation of semantics-based techniques. The framework encompasses various components such as data collection, pre-processing, malware detection model, decision engine, real-time protection module, and malware analysis and reporting. However, one aspect that could be further addressed in the paper is a comprehensive analysis of the potential false positive and false negative rates associated with the employed malware detection model. Evaluating the accuracy and reliability of the model is essential in minimizing false alarms and ensuring the effective detection and prevention of genuine malware instances. By providing a detailed examination of these rates, the authors can enhance the overall robustness and effectiveness of the proposed framework.

In the context of data integrity checking across different systems, Suchetha et al. (2020) conducted a survey focusing on data integrity and verification techniques specifically designed for cloud storage. The survey categorizes these techniques into cryptographic, erasure coding, replication, and secret sharing methods, shedding light on the strengths and limitations associated with each approach. This valuable information serves as a useful resource for selecting appropriate techniques to ensure data integrity and verification. However, it is worth noting that the paper may not extensively address the dynamic nature of cloud storage environments and the importance of adaptability in data integrity and verification techniques. Cloud storage systems often involve dynamic data updates, scalability considerations, and multi-tenant environments. Consequently, it is crucial to incorporate adaptable strategies that can effectively handle these dynamic aspects and ensure continuous data integrity and verification in cloud storage settings.In a similar manner, Yindong Chen et al. (2017) introduced a methodology that utilizes digital signatures and hash functions to authenticate and verify data. The paper emphasizes the significance of utilizing cloud storage service APIs to interact with the cloud storage platform and manage data blocks and digital signatures. This approach effectively reduces the risk of data tampering and guarantees the genuineness and integrity of data stored in the cloud. These two papers collectively offer a comprehensive overview of data integrity and verification techniques in cloud storage, assisting in the selection of appropriate methods for ensuring data integrity and verification. However, it is important to acknowledge that the discussed paper may not extensively delve into the potential limitations and vulnerabilities associated with the proposed approach. Considering potential attacks or vulnerabilities that could compromise data integrity and authenticity is crucial, even when employing digital signatures and hash functions.Jambulingam et al. (2019) introduces an adaptive methodology that incorporates various techniques, including data replication, hashing, Merkle tree, secret sharing, and adaptive fault detection, to identify single and multiple intrusions in cloud data. The combination of these techniques renders the proposed methodology highly effective in ensuring data integrity within cloud storage. However, the paper may not sufficiently address the adaptability of the methodology in dynamic cloud environments and evolving intrusion techniques. Cloud storage systems undergo continuous changes, such as data updates, system upgrades, and emerging intrusion methods. It would be advantageous for the paper to discuss how the proposed methodology can adapt to these dynamic aspects and successfully detect new and evolving intrusions.In contrast, Danyang et al. (2010) present a system that employs MD5 hashing, a block-based approach, redundant storage, error correction, and user authentication to safeguard data integrity in cloud storage. The system generates a secure MD5 hash of the original file and utilizes a block-based method to identify alterations in specific file sections. Additionally, the file and its corresponding hash values are stored in multiple locations to ensure data availability. While several research papers have proposed techniques for ensuring data and file integrity in cloud storage environments, it is important to note that the paper by Danyang et al. may not thoroughly examine the security vulnerabilities and limitations associated with MD5 hashing. MD5 has known security weaknesses, including susceptibility to collision attacks. Consequently, it would be advantageous for the paper to acknowledge these limitations and explore alternative, more robust hashing algorithms that could be implemented to enhance data integrity.In their paper, Gopalan et al. (2005) explore various techniques, including cryptographic hash functions, error-correcting codes, RAID, data mirroring, data scrubbing, and data verification, to maintain data integrity in storage systems. However, it is worth noting that the paper may not extensively address the individual limitations or vulnerabilities associated with each technique discussed. For instance, cryptographic hash functions, despite their widespread use, may be susceptible to collision attacks or vulnerabilities resulting from advancements in cryptanalysis. It would be beneficial for the paper to provide a more comprehensive analysis of these limitations and vulnerabilities to offer a clearer understanding of the potential risks associated with the employed techniques.

In conclusion, machine learning-based detection techniques such as flow-based detection and API call frequency analysis, and cloud-based malware detection systems with intrusion ontology representation have shown promising results in detecting malware. Integrity checking techniques such as code signing, checksums, and cryptographic hash functions are also important for ensuring the integrity of software. However, with the increasing complexity and sophistication of malware attacks, there is a need for continuous research and development of new techniques for malware detection and integrity checking. However, there are still challenges to overcome, such as adversarial attacks and the need for fast and efficient models. Further research is needed to develop more robust and effective solutions that can keep up with the constantly evolving threat landscape. Overall, blockchain-and ML based malware detection and integrity checking present a promising direction for the development of more effective and efficient malware detection systems.

3 Problem in brief – Scope

The scope of this project encompasses the development and evaluation of a decentralized approach for malware detection and integrity checking, leveraging blockchain technology and machine learning algorithms. The project aims to explore the potential benefits of utilizing blockchain for secure and decentralized storage of malware detection data and system integrity information. It also involves the implementation of machine learning models to enhance the accuracy and adaptability of malware detection. The project will include the design and development of the decentralized system architecture, integration of blockchain components, implementation of machine learning algorithms, and performance evaluation of the proposed solution.

This project aims to address several key questions and challenges that arise from traditional approaches to malware detection and integrity checking. These include:

1. How can a decentralized approach utilizing blockchain technology improve the security and resilience of malware detection and integrity checking systems compared to traditional centralized approaches?

2. Can the immutability and transparency of blockchain technology be effectively leveraged to ensure the integrity and authenticity of malware detection data and system integrity information?

3. How can machine learning algorithms be integrated with blockchain technology to enhance the accuracy and adaptability of malware detection, particularly in detecting new and evolving threats?

4. How can the proposed decentralized approach address the problem of false positives and false negatives in malware detection, ensuring a balance between detection rate and minimizing false alarms?

4 Aim & Objectives

4.1 Aim

The aim of this project is to develop a decentralized approach for malware detection and integrity checking by leveraging blockchain technology and machine learning algorithms. The project seeks to address the limitations of traditional centralized approaches by exploring the potential benefits of a decentralized system architecture that ensures data integrity, enhances security, and improves the adaptability of malware detection.

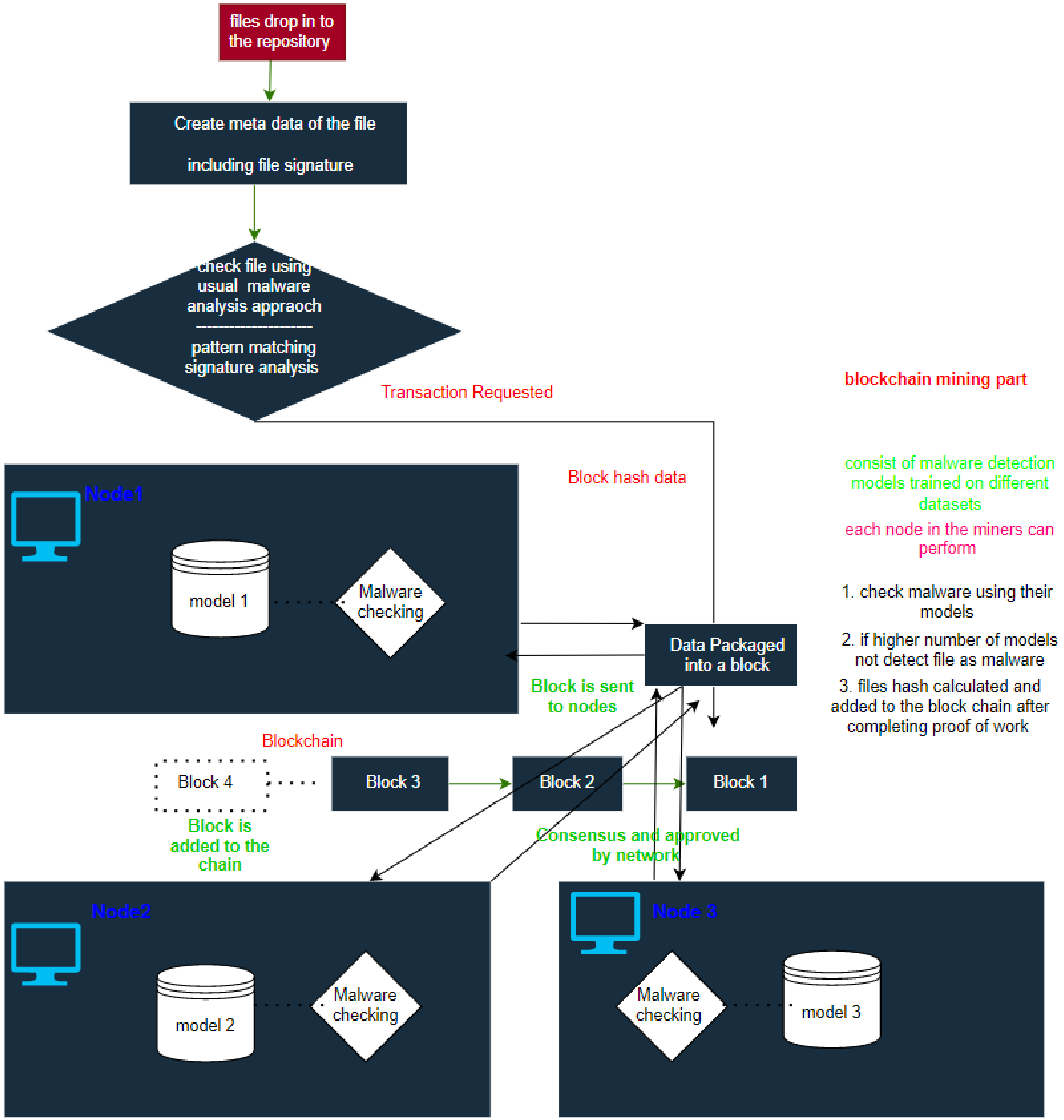
4.2 Objectives

- Design and develop a decentralized system architecture that utilizes blockchain technology for secure and decentralized storage of malware detection and integrity checking.
- Integrate machine learning algorithms into the decentralized system to enhance the accuracy and adaptability of malware detection.

5 Proposed Solution

1. Develop a multi-model approach: The project aims to design and implement multiple malware detection models using different datasets and techniques. These models will be used to analyze files in the organizational repository, providing a more comprehensive and accurate detection mechanism.

2. Implement blockchain-based file verification: The project will integrate blockchain technology into the file integrity checking process. A tamper-proof and transparent record of each file's history, including its signature and metadata, will be stored on the blockchain. This ensures the integrity of the files and provides a transparent audit trail.



files drop in to the repository
Create meta data of the file
including file signature
check file using usual malware analysis appraoch
pattern matching signature analysis
Transaction Requested
blockchain mining part
Block hash data
consist of malware detection models trained on different datasets
each node in the miners can perform
1. check malware using their models
2. if higher number of models not detect file as malware
3. files hash calculated and added to the block chain after completing proof of work
Node1
model 1
Malware checking
Data Packaged into a block
Block is sent to nodes
Blockchain
Block 4
Block 3
Block 2
Block 1
Block is added to the chain
Consensus and approved by network
Node2
model 2
Malware checking
Node 3
Malware checking
model 3

The proposed workflow of the system is as follows:

1. File Submission: Files are dropped into the repository, and metadata is generated for each file, including its signature. This metadata provides information about the file, such as its name, type, and size.

2. Traditional Malware Analysis: The files undergo traditional malware analysis techniques, such as pattern matching and signature analysis. This step helps identify known malware based on predefined patterns and signatures.

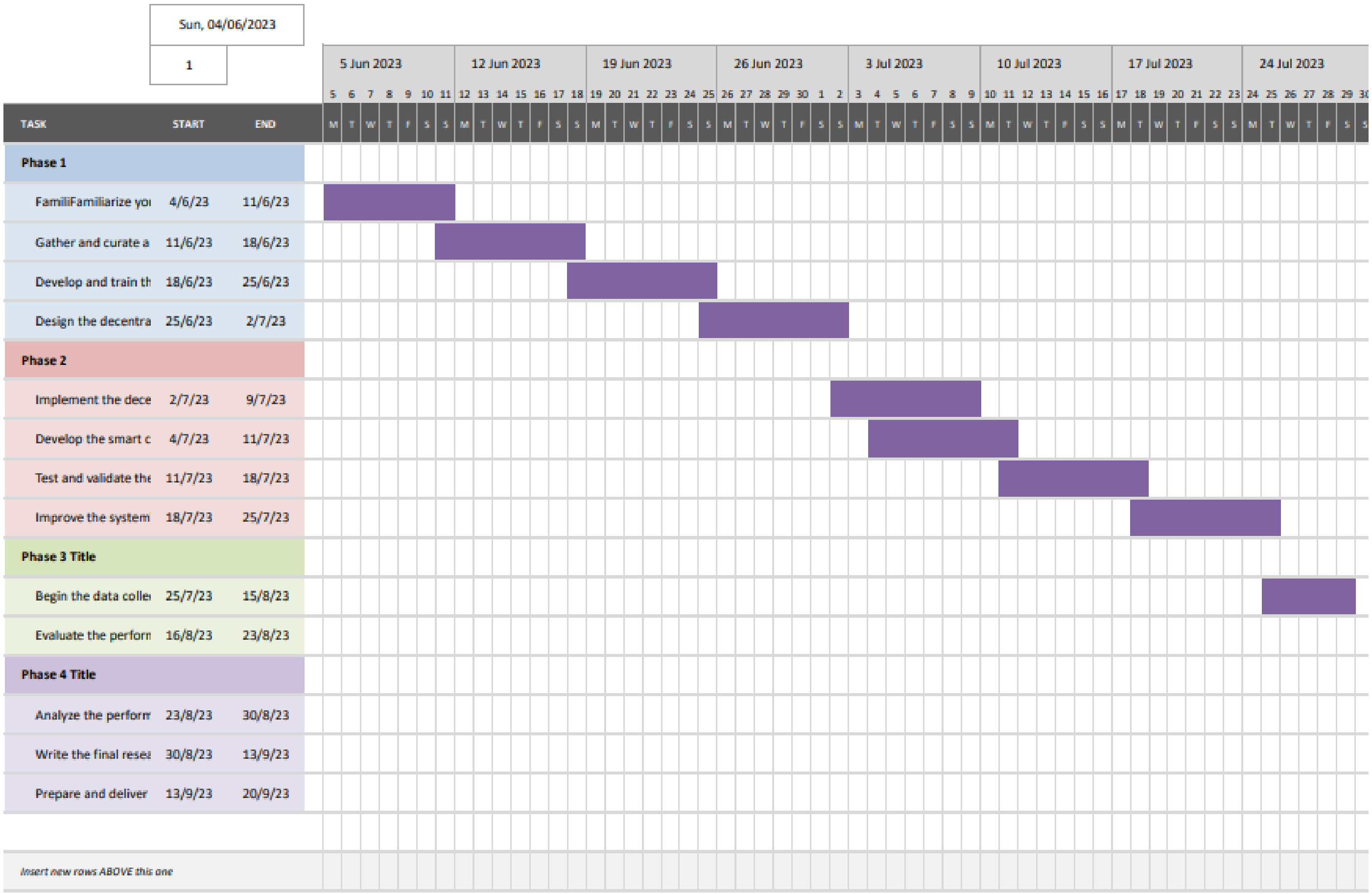
3. Blockchain-Based File Verification: A blockchain-based file verification system is implemented. This involves creating a block hash data that includes the file's metadata, including the file name, type, signature, and data size. This block hash acts as a digital fingerprint representing the file.

4. Malware Detection with Multiple Models: Each node in the mining nodes, which form part of the blockchain network, is equipped with a malware detection model. These models have been trained on different datasets and utilize various techniques. When a new file is added to the repository, each node independently checks the file using its respective malware detection model.

5. Consensus Mechanism: If a higher number of models do not detect the file as malware, a consensus is reached among the nodes. This consensus mechanism ensures that the file's hash is calculated and added to the blockchain, indicating that the file is considered safe based on the collective decision of the network.

6 Schedule

The proposed dissertation is scheduled to commence from the week beginning on June 4th, 2023, and is expected to conclude on September 20th, 2023. The following Gantt chart illustrates the task distribution, organized into groups based on their respective categories.



Sun, 04/06/2023

1

TASK	START	END
Phase 1		
FamiliFamiliarize yoı	4/6/23	11/6/23
Gather and curate a	11/6/23	18/6/23
Develop and train th	18/6/23	25/6/23
Design the decentra	25/6/23	2/7/23
Phase 2		
Implement the dece	2/7/23	9/7/23
Develop the smart c	4/7/23	11/7/23
Test and validate the	11/7/23	18/7/23
Improve the system	18/7/23	25/7/23
Phase 3 Title		
Begin the data colle	25/7/23	15/8/23
Evaluate the perforn	16/8/23	23/8/23
Phase 4 Title		
Analyze the perforn	23/8/23	30/8/23
Write the final rese	30/8/23	13/9/23
Prepare and deliver	13/9/23	20/9/23

Insert new rows ABOVE this one

Month	Week	Task
1	1	Familiarize yourself with existing research on blockchain-based ML systems for malware identification and integrity checking. Define the specific objectives, scope, and requirements of your project.
1	2	Gather and curate a diverse dataset of malware samples for training the ML model. Preprocess and prepare the malware dataset by extracting relevant features and ensuring data quality.

1	3	Develop and train the machine learning model using the prepared dataset. Evaluate the performance of the model using appropriate metrics and techniques.
1	4	Design the decentralized architecture for the malware identification and integrity checking system. Determine the necessary blockchain components and consensus mechanism to be implemented.
2	1	Implement the decentralized architecture and integrate the trained ML model into the system. Set up the blockchain network and configure the necessary components.
2	2	Develop the smart contracts or protocols required for recording the malware identification results and other relevant data on the blockchain. Implement the consensus mechanism chosen for achieving distributed consensus.
2	3	Test and validate the functionality of the integrated system. Conduct initial performance evaluations and address any issues or bugs.
2	4	Improve the system's scalability, fault tolerance, and efficiency by optimizing the decentralized architecture and the blockchain implementation. Prepare for the data collection phase by setting up necessary data storage and retrieval mechanisms.
3	1-3	Begin the data collection phase by processing and analyzing real-world malware samples using the developed system. Record the transaction data on the blockchain, including malware identification results, timestamps, and relevant metadata.
3	4	Evaluate the performance and accuracy of the system in detecting and identifying malware using the collected data. Analyze the results and compare them with existing centralized malware identification solutions.
4	1	Analyze the performance and effectiveness of the consensus mechanism in achieving distributed consensus for malware identification and integrity checking. Evaluate the system's scalability and fault tolerance under different workloads.
4	2-3	Write the final research report, documenting the methodology, findings, and conclusions of the project. Include performance evaluations, analysis of results, limitations, and recommendations for future work.

4	4	Prepare and deliver a presentation summarizing the project's objectives, methodology, and key findings. Review and finalize the research report, ensuring it meets the required standards.

7 Resource Requirements

- Python 3.7 and machine learning frameworks
- Virtual machines
- Diverse and representative malware datasets
- Web API (Flask python package)
- Python Crypto, Blockchain libraries