CYBERCRIME BLOCKCHAIN-ENABLED SECURITY FRAMEWORK TO DETECT AND DEFEND RANSOMWARE ATTACKS

A PROJECT REPORT

Submitted by

NALLAPANENI PENCHALA BALA TEJA [211420104174]

BHARATH N [211420104036]

PARSAM CHAITANYA [211420104189]

in the partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

In

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University,Chennai)

MARCH 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this Project report "**CYBERCRIME BLOCKCHAIN-ENABLED SECURITY FRAMEWORK TO DETECT AND DEFEND RANSOMWARE ATTACKS**" is the bonafide work of **NALLAPANENI PENCHALA BALA TEJA(211420104174),BHARATH N (211420104036) & PARSAM CHAITANYA (211420104189)** who carried out the project work under my supervision

Signature of the HOD with date

Dr.L.JABASHEELA M.E.,Ph.D.,

Professor and Head,

Department of Computer Science and Engineering,

Panimalar Engineering College,

Chennai – 123

Signature of the Supervisor with date

S.SOPHANA JENNIFER M.E..

Assistant Professor

Department of Computer Science and Engineering,

Panimalar Engineering College,

Chennai – 123

Submitted for the Project Viva-Voice examination held on .....

INTERNAL EXAMINER **EXTERNAL EXAMIER**

DECLARATION BY THE STUDENT

We **NALLAPANENI PENCHALA BALA TEJA (211420104174), BHARATH N (211420104036) & PARSAM CHAITANYA (211420104189)** hereby declare that this project report titled **"CYBERCRIME BLOCKCHAIN-ENABLED SECURITY FRAMEWORK TO DETECT AND DEFEND RANSOMWARE ATTACKS"**, under the guidance of **Mrs.SOPHANA JENNIFER S, M.E**., is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

NALLAPANENI PENCHALA BALA TEJA

BHARATH N

PARSAM CHAITANYA

ACKNOWLEDGEMENT

Our profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr. P. CHINNADURAI, M.A., Ph.D.,**for his benevolent words and fervent encouragement. His inspirational support proved instrumental in galvanizing our efforts, ultimately contributing significantly to the successful completion of this project

We want to express our deep gratitude to our Directors, **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHI KUMAR, B.E.,M.B.A., Ph.D.,** for graciously affording us the essential resources and facilities for undertaking of this project.

Our gratitude is also extended to our Principal, **Dr. K. MANI, M.E., Ph.D.,** whose facilitation proved pivotal in the successful completion of this project.

We express my heartfelt thanks to **Dr. L. JABASHEELA,M.E., Ph.D.,** Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of project.

We would like to express our sincere thanks to **Dr. PUGHAZENDI N,M.E., Ph.D.,** and **Mrs**.**SOPHANA JENIFFER S, M.E** and I also thank my parents ,friends and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

NALLAPANENI PENCHALA BALA TEJA(211420104174)

BHARATH N(211420104036)

PARSAM CHATANYA(211420104189)

ABSTRACT

Ransomware is a type of malicious program or software that encrypts the contents on a hard disc and prevents the users from accessing them unless they pay an amount (called a ransom). Most of the organizations, such as financial institutes and healthcare sectors (i.e., smart healthcare) are targeted by ransomware attacks. Ransomware assaults are among the most frightening types of cyber-attacks, and they are not confined to a specific sector or the countries. Blockchain is a tamper-proof technology, which is more secure, robust and decentralized in nature. Features of blockchain can add more security for detection and mitigation of ransomware more effectively. In this paper, we propose a new blockchain-enabled security framework to detect and defend the ransomware attacks. The conducted security analysis proves the security of the proposed against the ransomware attacks. The performance is significantly better than the other similar existing mechanisms as it achieves better accuracy and F1-score than other compared mechanisms. Furthermore, the practical demonstration is provided to estimate the impact on important performance parameters. Detecting ransomware is complex due to its varied forms and evasion techniques, posing significant risks to data integrity and confidentiality. There's a critical need for proactive defense mechanisms leveraging blockchain and machine learning to detect and mitigate ransomware threats effectively, safeguarding sensitive data and ensuring operational continuity.

Cybercriminals utilise ransomware as a sort of malware (malicious software). If a ransomware attack happens on a device, it either limits access or encrypts the data files of the device.Criminal demand ransom (some amount of money) from their victims in exchange for releasing the data .

TABLE OF CONTENTS

CHAPTER No	TITLE	PAGE
	ABSTRACT	v
	LIST OF FIGURES	viii
	LIST OF ABBREVATIONS	ix
1	**INTRODUCTION**	1
1.1	Problem Definition	1
2	**LITERATURE SURVEY**	2
3	**SYSTEM ANALYSIS**	9
3.1	Existing system	9
3.2	Proposed system	9
4	**REQUIREMENTS SPECIFICATION**	10
4.1	Hardware and Software specification	11
4.2	Technologies Used	12
5	**PROJECT PURPOSE AND SCOPE**	15
5.1	Purpose	15
5.2	Project Scope	15
5.3	Design and Implementation Constraints	16
5.4	Other Nonfunctional Requirements	16

6	**SYSTEM DESIGN**	18
7	**PROJECT DESCRIPTION**	26
7.1	Modules	26
8	**CODING AND TESTING**	30
8.1	Coding	30
8.2	Naming Conventions	30
8.3	Testing	32
9	**RESULT AND DISCUSSION**	33
10	**CONCLUSION AND FUTURE WORK**	35
	REFERENCES	36
	APPENDICES	37
A1	SOURCE CODE	37
A2	SCREEN SHOTS	63
A3	PLAGIARISM REPORT	66
A4	CONFERENCE PAPER	67

LIST OF FIGURES

FIG NO	TITLE	PAGE NO
6.1	Architecture Daigram	18
6.2	Sequence Diagram	19
6.3	Use Case Diagram	20
6.4	Activity Diagram	21
6.5	Collaboration Diagram:	22
6.6	Data Flow Diagram	23
6.7	Class Diagram	25

LIST OF ABBREVATIONS

ABBREVIATION	DEFINITION
DEX	Dalvik Executables
TCP	Transmission Control Protocol
IP	Internet Protocol
HTTP	Hyper Text Transfer Protocol
ADT	Android Development Tool

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION

A Cybercrime blockchain-enabled security framework to detect and defend ransomware attacks to ensure high protection and prevention Most of the organizations, such as financial institutes and healthcare sectors are targeted by ransomware attacks. Ransomware assaults are among the most frightening types of cyber- attacks, and they are not confined to a specific sector or the countries. Blockchain is a tamper- proof technology, which is more secure, robust and decentralized in nature. Features of blockchain can add more security for detection and mitigation of ransomware more effectively. The application of machine learning algorithms and techniques is to identify and recognize patterns, anomalies, or specific objects within data.In this paper, we propose a new blockchain- enabled security framework using machine learning to detect and defend the ransomware attacks.The problem addressed is the increasing threat of ransomware attacks. Traditional security measures are struggling to keep pace with evolving attack tactics, leaving individuals and organizations vulnerable to data breaches and extortion. Detecting ransomware is complex due to its varied forms and evasion techniques, posing significant risks to data integrity and confidentiality. There's a critical need for proactive defense mechanisms leveraging blockchain and machine learning to detect and mitigate ransomware threats effectively, safeguarding sensitive data and ensuring operational continuity.

CHAPTER 2

LITERATURE SURVEY

A literature review is a body of text that aims to review the critical points of current knowledge on and/or methodological approaches to a particular topic. It is secondary sources and discuss published information in a particular subject area and sometimes information in a particular subject area within a certain time. Its goal is to bring the reader up to date with current literature on a topic and forms the basis for another goal, such as future research that may be needed in the area and precedes a research proposal and may be just a simple summary of sources. Usually, it has an organizational pattern and combines both summary and synthesis.

Project Title :Ransomware: A Survey and Trends

Author Name :Sana Aurangzeb, Muhammad Aleem, Muhammad Azhar Iqbal and Muhammad Arshad Islam

Year : 2022

Ransomware are a recent scareware, a threat that is increasing gradually for last couple of years. Usually it encrypts users's files or steal/delete important information and holds the decryption key until a ransom is paid by the victim, which is mostly in bitcoins due to their untraceable properties. In this work, we have conducted a survey to comprehend more than 40papers that consists of Windows based ransomware families from last 4 years by creating a benchmark for evaluating ransomware attacking methodologies, payment methods. Thus,providing a way of instigating for other researchers to come out with some new solutions to control their growth and to avoid their attacks. This work would be the ultimate opportunity to expand and organize research efforts in future work by applying some early preventions and strategies to defeat these malicious software's.

Project Title : Deep Learning vs. Traditional Computer Vision

Author Name : Niall Mahony, Sean Campbell, Anderson Carvalho, Suman Harapanahalli, Gustavo Velasco Hernandez, Lenka Krpalkova, Daniel Riordan, Joseph Walsh

Year : 2021

Deep Learning has pushed the limits of what was possible in the domain of Digital Image Processing. However, that is not to say that the traditional computer vision techniques which had been undergoing progressive development in years prior to the rise of DL have become obsolete. This paper will analyse the benefits and drawbacks of each approach. The aim of this paper is to promote a discussion on whether knowledge of classical computer vision techniques should be maintained. The paper will also explore how the two sides of computer vision can be combined. Several recent hybrid methodologies are reviewed which have demonstrated the ability to improve computer vision performance and to tackle problems not suited to Deep Learning. For example, combining traditional computer vision techniques with Deep Learning has been popular in emerging domains such as Panoramic Vision and 3D vision for which Deep Learning models have not yet been fully optimised.

Project Title: Crypto-ransomware detection using machine learning models in file-sharing network scenario with encrypted traffic

Author Name: Eduardo Berrueta, Daniel Morato, Eduardo Magaña, Mikel Izal

Year : 2020

Ransomware is considered as a significant threat for most enterprises since the past few years. In scenarios wherein users can access all files on a shared server, one infected host can lock the access to all shared files. We propose a tool to detect ransomware infection based on file-sharing traffic analysis. The tool monitors the traffic exchanged between the clients and the file servers and using machine learning techniques it searches for patterns in the traffic that betray ransomware actions while reading and overwriting files. The proposal is designed to work for clear text and for encrypted file-sharing protocols. We compare three machine learning models and choose the best for validation. We train and test the detection model using more than 70 ransomware binaries from 26 different strains and more than 2500 hours of ‘not infected’ traffic from real users. The results reveal that the proposed

tool can detect all ransomware binaries, including those not used in training phase (unseen). This paper provides a validation of the algorithm by studying the false positive rate and the amount of information from user files that the ransomware could encrypt before being detected.

Project Title :Blockchain as privacy and security solution for smart environments: A Survey
Author Name :maad ebrahim, abdelhakim hafid, and etienne elie
Year : 2022

Blockchain was always associated with Bitcoin, cryptocurrencies, and digital asset trading. However, the benefits of Blockchain are far beyond that. It has been recently used to support and augment many other technologies, including the Internet-of-Things (IoT). IoT, with the help of Blockchain, paves the way for futuristic smart environments, like smart homes, smart transportation, smart energy trading, smart industries, smart supply chains, and more. To enable these smart environments, IoT devices, machines, appliances, and vehicles, will need to intercommunicate without the need for a centralized trusted party. Blockchain can replace third trusted parties by providing secure means of decentralization in such trustless environments. They also provide security enforcement, privacy assurance, authentication, and other important features to IoT ecosystems. Besides the benefits of Blockchain-IoT integration for smart environments, other technologies also have important features and benefits that attracted the research community. Software-Defined Networking (SDN), Fog, Edge, and Cloud Computing technologies, for example, play an important role in enabling realistic IoT applications. Moreover, the integration of Machine Learning and Artificial Intelligence (AI) algorithms provides smart, dynamic, and autonomous decision making capabilities for IoT devices in smart environments. To push the research further in this domain, we provide in this paper a comprehensive survey that includes state-of-the-art technological integration, challenges, and solutions for smart environments, and the role of Blockchain and IoT technologies as the building blocks of such smart environments. We also demonstrate how the level of integration between these technologies has increased over the years, which brings us closer to the futuristic view of smart environments. We further discuss the current need to provide general-purpose Blockchain platforms that can adapt to different design requirements of different applications and solutions. Finally, we provide a simplified architecture of futuristic smart environments that integrates all these technologies, showing the advantage of such integration.

Project Title :A Survey on Detection Techniques for Cryptographic Ransomware

Author Name: eduardo berrueta, daniel morato, eduardo magaña,mikel izal

Year : 2022

Crypto-ransomware is a type of malware that encrypts user files, deletes the original data, and asks for a ransom to recover the hijacked documents. It is a cyber threat that targets both companies and residential users, and has spread in recent years because of its lucrative results. Several articles have presented classifications of ransomware families and their typical behaviour. These insights have stimulated the creation of detection techniques for antivirus and firewall software. However, because the ransomware scene evolves quickly and aggressively, these studies quickly become outdated. In this study, we surveyed the detection techniques that the research community has developed in recent years. We compared the different approaches and classified the algorithms based on the input data they obtain from ransomware actions, and the decision procedures they use to reach a classification decision between benign or malign applications. This is a detailed survey that focuses on detection algorithms, compared to most previous studies that offer a survey of ransomware families or isolated proposals of detection algorithms. We also compared the results of these proposals.

Project Title :Blockchain for Internet of Things: A Survey

Author Name :Hong-Ning Dai, ZibinZheng, Yan Zhang

Year : 2020

Internet of Things (IoT) is reshaping the incumbent industry to smart industry featured with data-driven decision making. However, intrinsic features of IoT result in a number of challenges such as decentralization, poor interoperability, privacy and security vulnerabilities. Blockchain technology brings the opportunities in addressing the challenges of IoT. In this paper, we investigate the integration of blockchain technology with IoT. We name such synthesis of blockchain and IoT as Blockchain of Things (BCoT). This paper presents an in-depth survey of BCoT and discusses the insights of this new paradigm. In particular, we first briefly introduce IoT and discuss the challenges of IoT. Then we give an overview of blockchain technology. We next concentrate on introducing the convergence of blockchain and IoT and presenting the proposal of BCoT architecture. We further

discuss the issues about using blockchain for 5G beyond in IoT as well as industrial applications of BCoT. Finally, we outline the open research directions in this promising area.

Project Title : Blockchain Technologies for IoT

Author Name : V. Dedeoglu, R. Jurdak, A. Dorri, R. C. Lunardi, R. A. Michelin, A. F. Zorzo and S. S. Kanhere

Year : 2020

The exponential increase in connected devices with built-in sensing, processing, and communication capabilities has fuelled the development of IoT applications, which creates new ecosystems for device-to-device interactions, supports smart environments, and leads to new business models. Empowered by these capabilities, IoT devices interact with each other and their environments to collect, process, and share data. Security, privacy, and reliability of data are major concerns that need to be addressed for the development of IoT applications. Recently, blockchain technology has attracted significant interest from researchers and industry leaders due to its potential for enhancing security, privacy, and reliability of the data. Blockchain offers distributed and immutable ledgers for IoT communications in the form of tamper-proof records, built-in cryptocurrency support for transactions between devices and other entities, and smart contracts to execute automated programs when certain conditions are met. Although there are potential benefits of the integration of blockchain technology to IoT, the integration introduces new challenges, such as scalability, in the design of blockchains suited for IoT applications. In this chapter, we explore key benefits and design challenges for blockchain technologies, and potential applications of blockchain technologies for IoT.

Project Title : Design of Intrusion Detection Honeypot Using Social Leopard Algorithm to Detect IoT Ransomware Attacks

Author Name : s. Sibi chakkaravarthy 1, d. Sangeetha2,3,4, meenalosini vimal cruz2,3,4,v. Vaidehi2,3,4, and balasubramanian raman5.

Year : 2020

In recent times, ransomware has become the most significant cyber-attack targeting individuals, enterprises, healthcare industries, and the Internet of Things (IoT). Existing security systems like

Intrusion Detection and Prevention System (IDPS) and Anti-virus (AV) as a single monitoring agent is complicated and time-consuming, thus fails in ransomware detection. A robust Intrusion Detection Honeypot (IDH) is proposed to address the issues mentioned above. IDH consists of i) Honeyfolder, ii) Audit Watch, and iii) Complex Event Processing (CEP). Honeyfolder is a decoy folder modeled using Social Leopard Algorithm (SoLA), especially for getting attacked and acting as an early warning system to alert the user

during the suspicious file activities. AuditWatch is an Entropy module that verifies the entropy of the files and folders. CEP engine is used to aggregate data from different security systems to confirm the ransomware behavior, attack pattern, and promptly respond to them. The proposed IDH is experimentally tested in a secured testbed using more than 20 variants of recent ransomware of all types. The experimental result confirms that the proposed IDH significantly improves the ransomware detection time, rate, and accuracy compared with the existing state of the art ransomware detection model.

Project Title : Lightweight and Privacy-Preserving Remote User Authentication

Author Name : k. Nimmy1, (student member, ieee), sriram sankaran1, (member, ieee), Krishnashree achuthan1, (senior member, ieee), prasad calyam2, (senior Member, ieee)

Year : 2021

The rapid proliferation of embedded devices has led to the growth of the Internet of

Things (IoT) with applications in numerous domains such as home automation, healthcare, education and agriculture. However, many of the connected devices particularly in smart homes are the target of attacks that try to exploit security vulnerabilities such as hard-coded passwords and insecure data transfer. Recent studies show that there is a considerable surge in the number of phishing attacks targeting smart homes during the COVID-19 pandemic. Moreover, many of the existing user authentication protocols in the literature incur additional computational overhead and need to be made more resilient to smart home targeted attacks. In this paper, we propose a novel lightweight and privacy-preserving remote user authentication protocol for securing smart home applications. Our approach is based on Photo Response Non-Uniformity (PRNU) to make our protocol resilient to smart home attacks such as smartphone capture attacks and phishing attacks. In addition, the lightweight

nature of our solution is suitable for deployment on heterogeneous and resource constrained IoT devices. Besides, we leverage geometric secret sharing for establishing mutual authentication among the participating entities. We validate the security of the proposed protocol using the AVISPA formal verification tool and prototype it on a Raspberry Pi to analyze the power consumption. Finally, a comparison with existing schemes reveals that our scheme incurs a 20% reduction in communication overhead on smart devices. Furthermore, our proposed scheme is usable as it absolves users from memorizing passwords and carrying smart cards.

Project Title : Ransomware Detection Using the Dynamic Analysis and Machine Learning: A Survey and Research Directions

Author Name : Umara Urooj 1,*, Bander Ali Saleh Al-rimy 1, Anazida Zainal 1, Fuad A. Ghaleb 1 and Murad A. Rassam 2,3

Year : 2021

Ransomware is an ill-famed malware that has received recognition because of its lethal and irrevocable effects on its victims. The irreparable loss caused due to ransomware requires the timely detection of these attacks. Several studies including surveys and reviews are conducted on the evolution, taxonomy, trends, threats, and countermeasures of ransomware. Some of these studies were specifically dedicated to IoT and android platforms. However, there is not a single study in the available literature that addresses the significance of dynamic analysis for the ransomware detection studies for all the targeted platforms. This study also provides the information about the datasets collection from its sources, which were utilized in the ransomware detection studies of the diverse platforms. This study is also distinct in terms of providing a survey about the ransomware detection studies utilizing machine learning, deep learning, and blend of both techniques while capitalizing on the advantages of dynamic analysis for the ransomware detection. The presented work considers the ransomware detection studies conducted from 2019 to 2021.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

In the Existing System, they mainly focused only on the aftermath of ransomware attack by detection of ransomware and recovery of data. They didn't proposed how to prevent the system from the ransomware attack.

Disadvantages

The proposed system didn't focused on the antecedence of ransomware attack will continue to be a significant drawback.

3.2 PROPOSED SYSTEM

In the proposed system, we focused on antecedence of ransomware attack as well as aftermath. We also provided a demo attack of Locker Ransomware and Crypto Ransomware to get real-time experience and to aware of it. We will also continuously monitor the system, if any suspicious files is detected it will be removed automatically. We will take backup of the user data to recover it later, in case of ransomware attack so that we don't need to pay ransom to the attacker.In detection on ransomware we will mainly focuses on the portable executable files which plays a major role in gaining access of the system. We will do the detection of ransomware in Portable Executable files using Honeypot dataset and then the features/signatures are extracted followed by Feature Selection, Correlation and Classification using Machine Learning to detect that the executable file is Legitimate or Infected.

Advantages

The continuous monitoring of the system and automatic removal of files, in the event of suspicious activity followed by detection of ransomware and recovery of data make this proposal to stand out than the others.

CHAPTER 4

REQUIREMENT SPECIFICATION

RANSOMWARE is a type of malware attack in which the attacker encrypts and locks the victim's data and crucial files, and then demands money (ransom) in exchange for the data to be unlocked and decrypted. Cybercriminals utilize ransomware as a sort of malware (malicious software). If a ransomware attack happens on a device, it either limits access or encrypts the data (files) of the device. Cybercriminals demand ransom (some amount of money) from their victims in exchange for releasing the data (i.e., to provide the decryption key). A close eye and security software are recommended to protect against ransomware outbreak. After being infected with ransomware, victims have only few options, such as a) pay the money (ransom), b) try to remove the ransomware, and c) reset the device. Extortion malware commonly exploits remote systems via phishing emails and other software flaws. A ransomware attack can affect both the individual users and the companies of commerce. Ransomware is one of the most effective ways to attack businesses, key infrastructure (i.e., smart healthcare system, power grid, nuclear power plant) and the associated people. This type of malware infects computers and prevents users or external software from accessing devices until ransom demands are paid. The healthcare digital experience can become like the other digital interactions. For example, we use smartphone applications for ordering of some foods online which is just done with the help of few clicks. Where we order, pay, and even customize how our foods should be cooked, we can tell them when and where it should be delivered. Consumer electronics and technology have led to the personalized, integrated, and seamless experiences in the consumer-facing industries. This can be followed in the healthcare domain (i.e., smart healthcare). For instance, we can have elderly people at home (i.e., smart home). To monitor their health or to support their day-to- day activities, we can deploy different consumer electronics devices and products (i.e., smart coffee makers, smart air conditioners and wearable healthcare devices), which can do this task with the help of installed software and Internet connectivity. The smart healthcare system should provide a more seamless and customized experience to fulfil rising customer expectations as we

transit to a more consumer-centric future of health. Therefore, smart healthcare system becomes a par of consumer electronics. It can be considered as the consumer healthcare technology, where consumer healthcare electronics devices (i.e., wearable healthcare devices) support the people in various ways. Consumers (i.e., patients) expect healthcare technologies should function like those of other industries. In this direction, a better technology and digital experience can improve their satisfaction if it helps to increase the efficiency of the system and provide more convenience. It may include virtual visits to hospitals/clinics, registration, online scheduling of appointment and bill payments are among those being adopted by hospitals so that they can interact directly with the consumers for accessing, communicating and delivering better healthcare and bill payments.

4.1 HARDWARE AND SOFTWARE SPECIFICATION

4.1.1 HARDWARE REQUIREMENTS

Hard Disk : 250GB and Above

RAM : 6GB and Above

Processor : i3 and Above

4.1.2 SOFTWARE REQUIREMENTS

- Windows 10 and Above
- Visual Studio Code
- Node.js
- React
- Ganache
- MetaMask

4.2 TECHNOLOGIES USED

- Block Chain
- Python

4.2.1 Blockchain

With the emergence of Digital Currency (aka Crypto currency), several enterprises or financial institutions are experimenting with the Distributed Ledger system as a trusted way to track the ownership of the assets without any central authority.The core system behind the new currency system is Blockchain technology. A walkthrough of the basic building blocks of the Blockchain technology is described below.A Blockchain is basically a chain of Blocks. Blocks are hashed using SHA-256 hashing algorithm to generate the signature of the data associated with it.

Imagine a Blockchain as a linked-list whose node contains below attributes:

1. Block number – a sequence number (monotonically increasing) assigned to the block
2. Nonce – a random number which is used to generate Hash (as in #5) value which starts with 4 zeroes (0000). The process of generating this Nonce is called Mining.
3. Data – the actual user data associated with the block
4. Prev – contains the Hash of the previous block (e.g. current block # -1). The value for the first block in the chain is 64 zeroes .
5. Hash – current block's Hash value (generated using SHA-256). All of the above attributes excluding Hash e.g. Block #, Nonce, data, Prev are used to calculate the Hash of this block.

[#=1, Nonce=3409, Data=x, Prev=00..0, Hash=0000ffgr5rg67j] <- [#=2, Nonce=4986, Data=x, Prev=0000ffgr5rg67j, Hash=000045tggr5rg..77yh] <-……and the chain goes on…… e.g. in above block #1, the value for Hash=0000ffgr5rg67j is generated using the values 1,3409,x,00..0. In case value for any of these 4 attributes changes, it will change the Hash value

of this block. Once the Hash value of this Block changes (e.g. from 0000ffgr5rg67j to 34sdffgr5rg67j), it will break the next Block (#2) as its Prev field will point to invalid Hash (0000ffgr5rg67j doesn't exist anymore). This leads to a ripple effect and turns whole chain as invalid/tampered.

One way to fix it is to run mining and recalculate the Hash value of Block #1 which basically will generate new value for Nonce and hence leading to a valid Hash value which starts with 4 zeroes. Copying this to next Block #2's Prev field will fix these 2 Blocks. However in order to fix the whole Blockchain, we need to continue with this process for all the Blocks in the chain so that all Blocks point to new & valid Hash codes of their previous blocks.

The cost of fixing the tampered Blockchain as described in above process is very high. Because we have to go and fix the Chain from the starting Block to the last one. In case the Chain is large, it becomes costly operation. In case of Distributed Blockchain where several Peers are involved in the process and keeping the copy of the Blockchain, the repairing the Blocks becomes even more costly operation.

The other and more efficient process is to come up with the compensating data and add this Block at the end of the Chain. E.g. In case your Chain contains the financial transaction (money movement) in Data field of the Block, then instead of fixing each of the Block's Data with corrected financial transaction, come up with the adjusted financial transaction (aka compensating transaction) and create a Block (with Data=adjusted transaction record) and add this Block to the Blockchain (adds to the end of the Chain).

Fig 4.2.1.1 hash key generation

fig 4.2.2 block generation

React

React.js is an open-source JavaScript library, crafted with precision by Facebook, that aims to simplify the intricate process of building interactive user interfaces. Imagine a user interface built with React as a collection of components, each responsible for outputting a small, reusable piece of HTML code. In React, you develop your applications by creating reusable components that you can think of as independent Lego blocks. These components are individual pieces of a final interface, which, when assembled, form the application's entire user interface.

CHAPTER 5

PROJECT PURPOSE AND SCOPE

5.1 PURPOSE

The purpose of the project titled "Cybercrime Blockchain-Enabled Security Framework to Detect and Defend Ransomware Attacks" is to develop an innovative cybersecurity solution that leverages blockchain technology to effectively combat the growing threat of ransomware attacks. By integrating blockchain principles into traditional cybersecurity frameworks, the project aims to enhance detection and defense mechanisms, ultimately bolstering the resilience of organizations against ransomware threats. Through thorough research, design, and implementation, the project seeks to contribute to the advancement of cybersecurity practices and mitigate the impact of ransomware attacks on systems and data integrity.

5.2 PROJECT SCOPE

The project entails a comprehensive examination of recent ransomware attacks to decipher their methodologies and impacts on targeted systems and organizations. This analysis sets the foundation for exploring the integration of blockchain technology into cybersecurity frameworks, with a specific focus on enhancing detection and defense mechanisms against ransomware threats. Subsequently, a tailored blockchain-enabled security framework will be designed and developed, amalgamating traditional cybersecurity measures with innovative blockchain solutions. This framework will be fortified with advanced ransomware detection mechanisms leveraging blockchain technology, encompassing anomaly detection, behavior analysis, and machine learning algorithms. Additionally, the project involves the development and implementation of defense strategies to mitigate the impact of ransomware attacks, encompassing dynamic access controls, secure backup solutions, and incident response protocols. Rigorous testing and evaluation will be conducted to assess the effectiveness, scalability, and resilience of the developed security framework under diverse simulated attack scenarios. Finally, meticulous documentation of the project's development process, including design decisions, implementation details, testing results, and encountered challenges, will be compiled into comprehensive reports to facilitate knowledge dissemination and serve as a reference for future endeavors.

5.3 DESIGN AND IMPLEMENTATION CONSTRAINTS

5.3.1 Constraints in Analysis

5.3.1.1 Constraints as Informal Text

5.3.1.2 Constraints as Operational Restrictions

5.3.1.3 Constraints Integrated in Existing Model Concepts

5.3.1.4 Constraints as a Separate Concept

5.3.1.5 Constraints Implied by the Model Structure

5.3.2 Constraints in Design

5.3.2.2 Determination of the Involved Classes

5.3.2.3 Determination of the Involved Objects

5.3.2.4 Determination of the Involved Actions

5.3.2.5 Determination of the Require Clauses

5.3.2.6 Global actions and Constraint Realization

5.3.3 Constraints in Implementation

A hierarchical structuring of relations may result in more classes and a more complicated structure to implement. Therefore it is advisable to transform the hierarchical relation structure to a simpler structure such as a classical flat one. It is rather straightforward to transform the developed hierarchical model into a bipartite, flat model, consisting of classes on the one hand and flat relations on the other. Flat relations are preferred at the design level for reasons of simplicity and implementation ease. There is no identity or functionality associated with a flat relation. A flat relation corresponds with the relation concept of entity-relationship modeling and many object oriented methods.

5.4 OTHER NON-FUNCTIONAL REQUIREMENTS

5.4.1 Performance Requirements

The application at this side controls and communicates with the following three main general components.

- embedded browser in charge of the navigation and accessing to the web service;

- Server Tier: The server side contains the main parts of the functionality of the proposed architecture. The components at this tier are the following. Web Server, Security Module, Server-Side Capturing Engine, Preprocessing Engine, Database System, Verification Engine, Output Module.

5.4.2 Safety Requirements

1. The software may be safety-critical. If so, there are issues associated with its integrity level
2. The software may not be safety-critical although it forms part of a safety-critical system. For example, software may simply log transactions.
3. If a system must be of a high integrity level and if the software is shown to be of that integrity level, then the hardware must be at least of the same integrity level.
4. There is little point in producing 'perfect' code in some language if hardware and system software (in widest sense) are not reliable.
5. If a computer system is to run software of a high integrity level then that system should not at the same time accommodate software of a lower integrity level.
6. Systems with different requirements for safety levels must be separated.
7. Otherwise, the highest level of integrity required must be applied to all systems in the same environment.

CHAPTER 6

6.1 Architecture Diagram

Architecture diagramming is the process of creating visual representations of software system components. It plays a crucial role in showcasing the various functions of a software system, their implementations, and how they interact with each other. By providing a bird's-eye view of the system, architectural diagrams facilitate better decision-making and enhance the clarity of complex software structures**.**

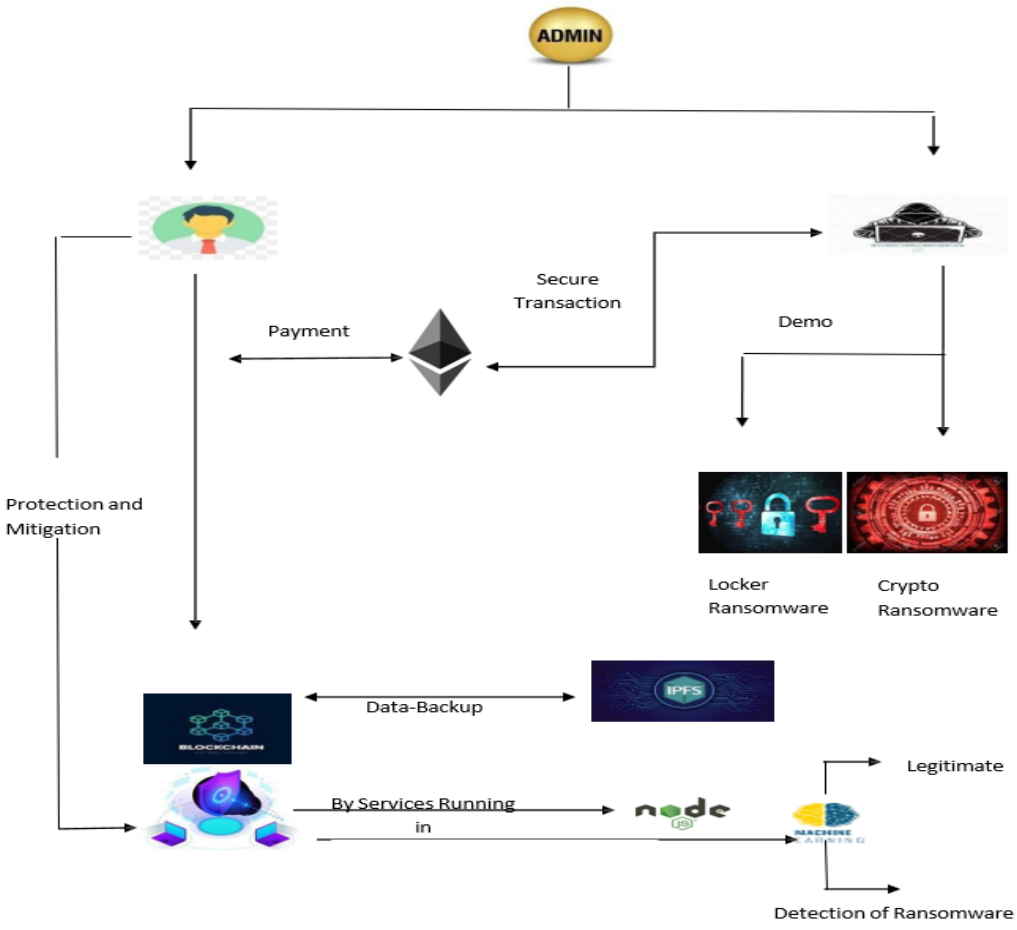

Fig 6.1.1 Architecture Diagram

6.2 Sequence Diagram

A Sequence diagram is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of Message Sequence diagrams are sometimes called event diagrams, event sceneries and timing diagram.

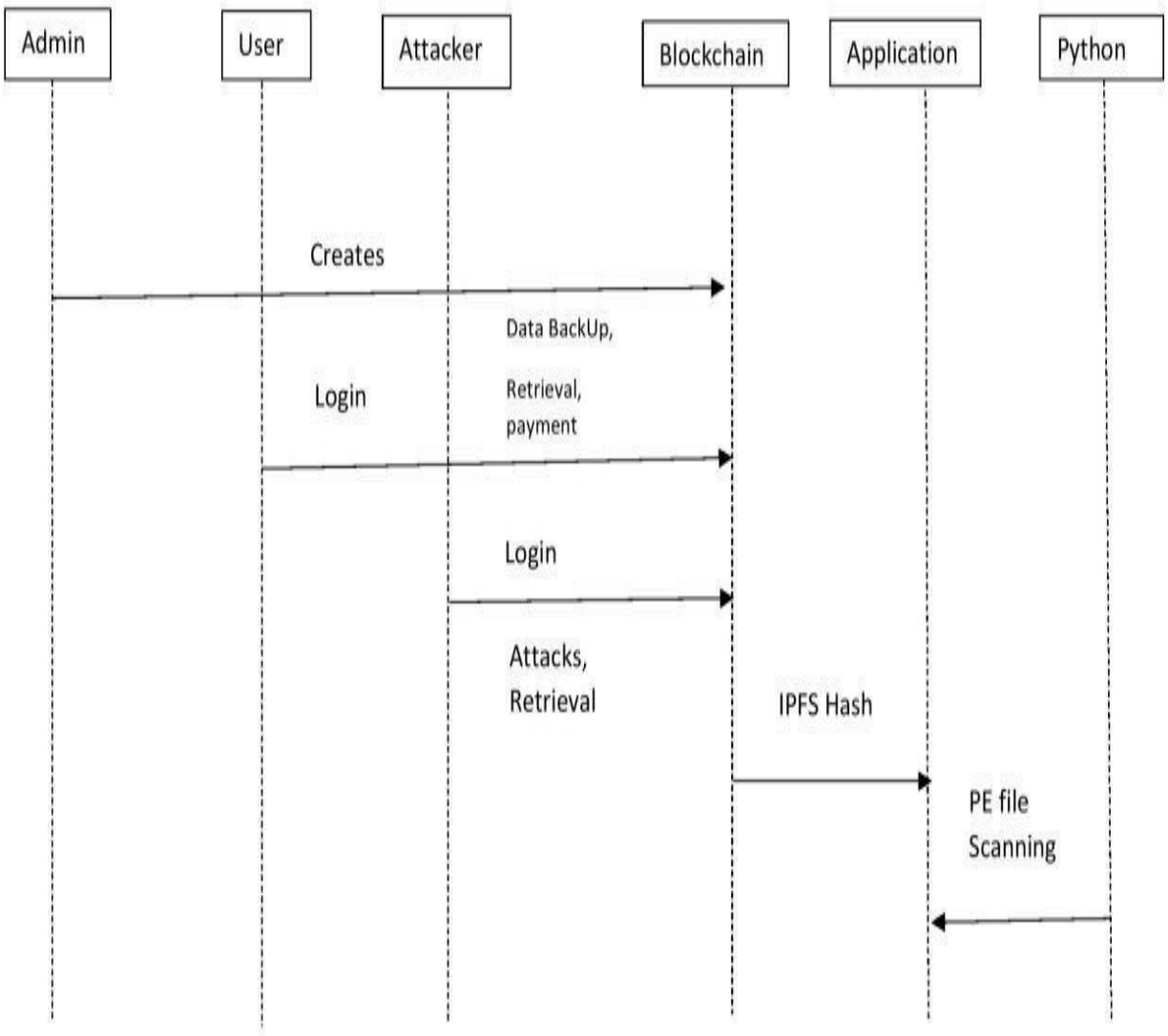

Fig 6.2.1 sequence diagram

6.3 Usecase Diagram

Unified Modeling Language (UML) is a standardized general-purpose modeling language in the field of software engineering. The standard is managed and was created by the Object Management Group. UML includes a set of graphic notation techniques to create visual models of software intensive systems. This language is used to specify, visualize, modify, construct and document the artifacts of an object oriented software intensive system under development.

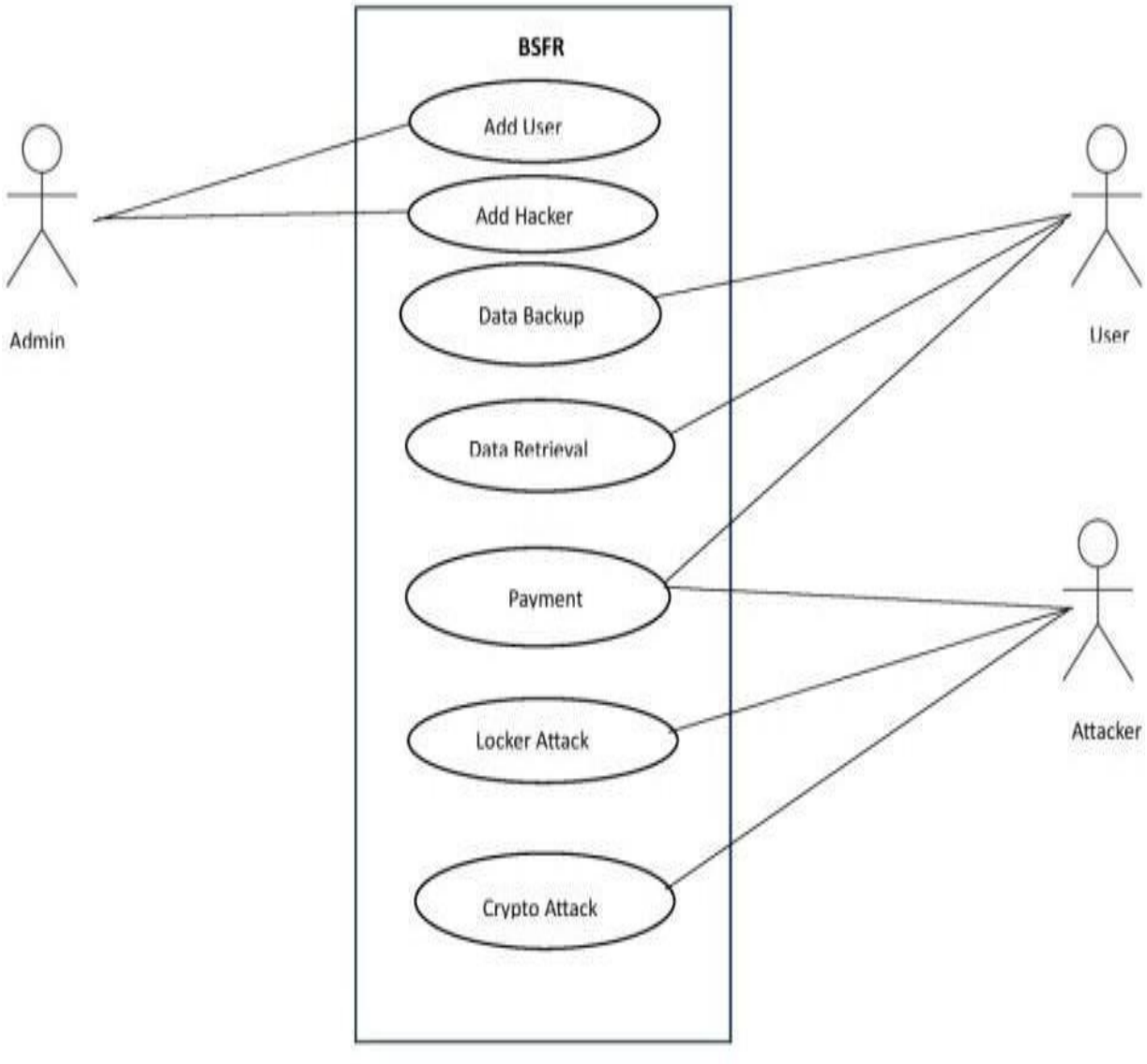

Fig 6.3.1 use case diagram

6.4 Activity Diagram

Activity diagram is a graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. An activity diagram shows the overall flow of control.

The most important shape types:

- Rounded rectangles represent activities.
- Diamonds represent decisions.
- Bars represent the start or end of concurrent activities.
- A black circle represents the start of the workflow.
- An encircled circle represents the end of the workflow.

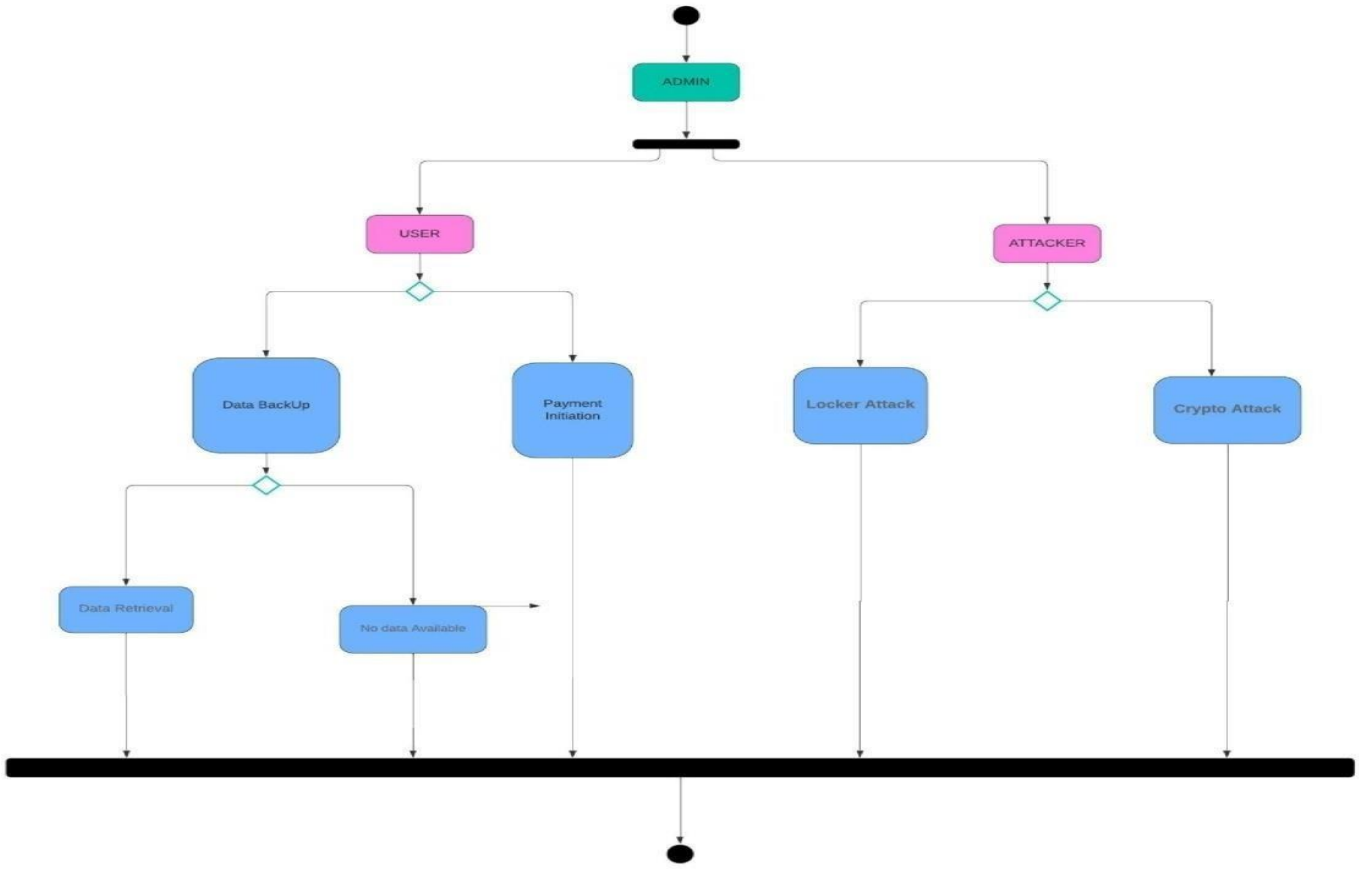

Fig 6.4.1 activity diagram

6.5 Collaboration Diagram

UML Collaboration Diagrams illustrate the relationship and interaction between software objects. They require use cases, system operation contracts and domain model to already exist. The collaboration diagram illustrates messages being sent between classes and objects.

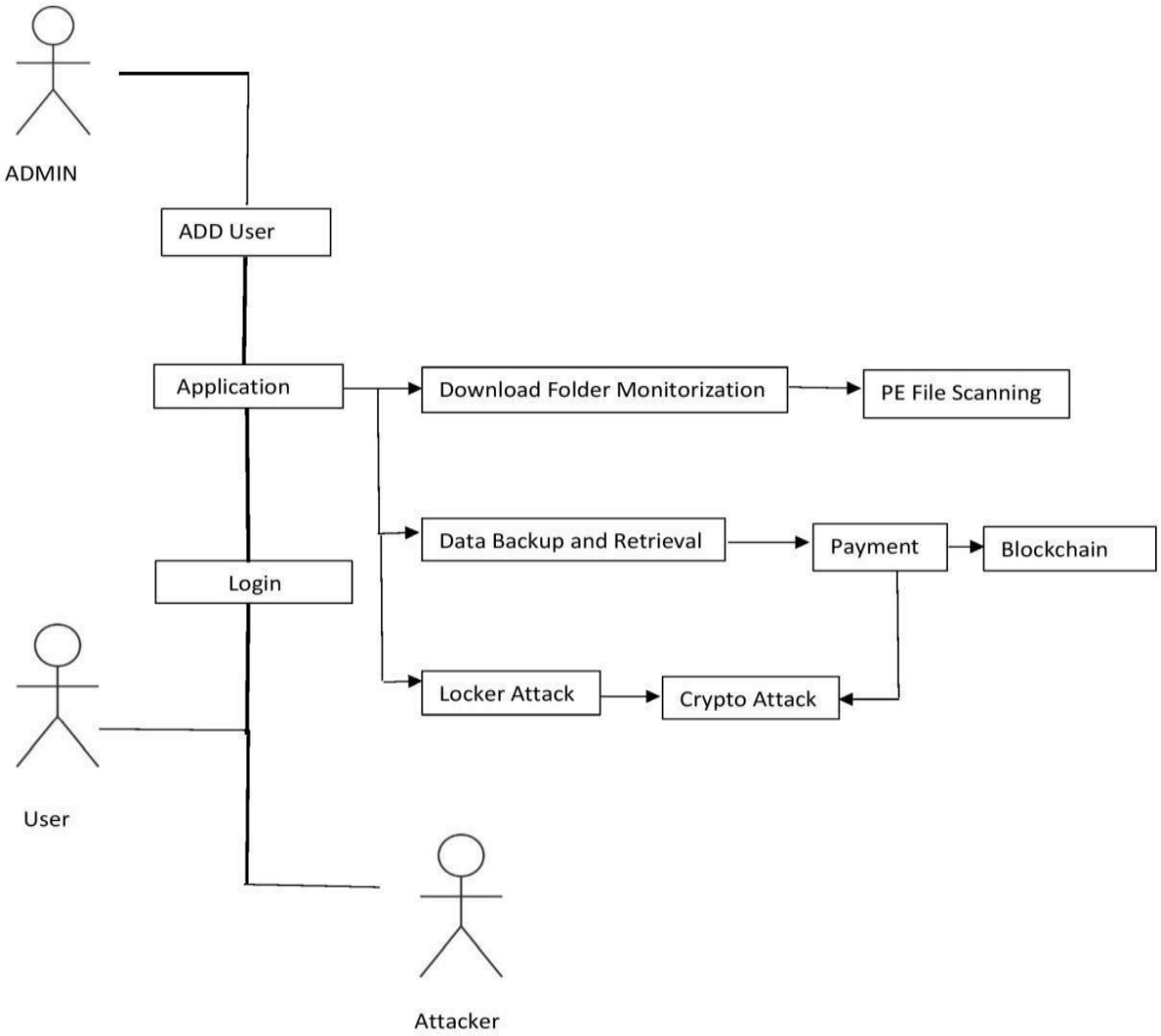

Fig 6.5.1 Collaboration Diagram

6.6 Data Flow Diagram

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its aspects. It is a preliminary step used to create an overview of the system which can later be elaborated DFDs can also be used for visualization of data processing.

Level 0

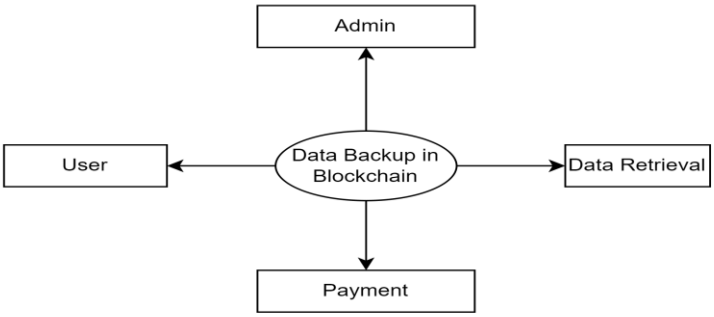

Fig 6.6.1 level 0 data flow diagram

Level 1

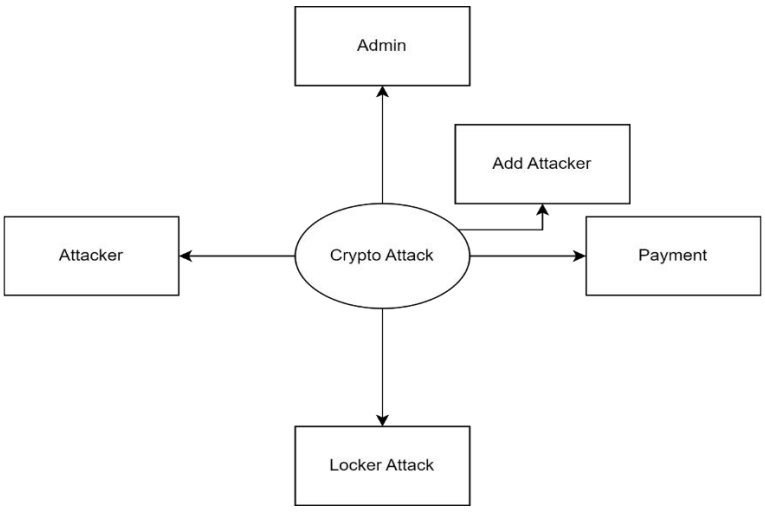

Fig 6.6.2 level 1 data flow diagram

Level 2

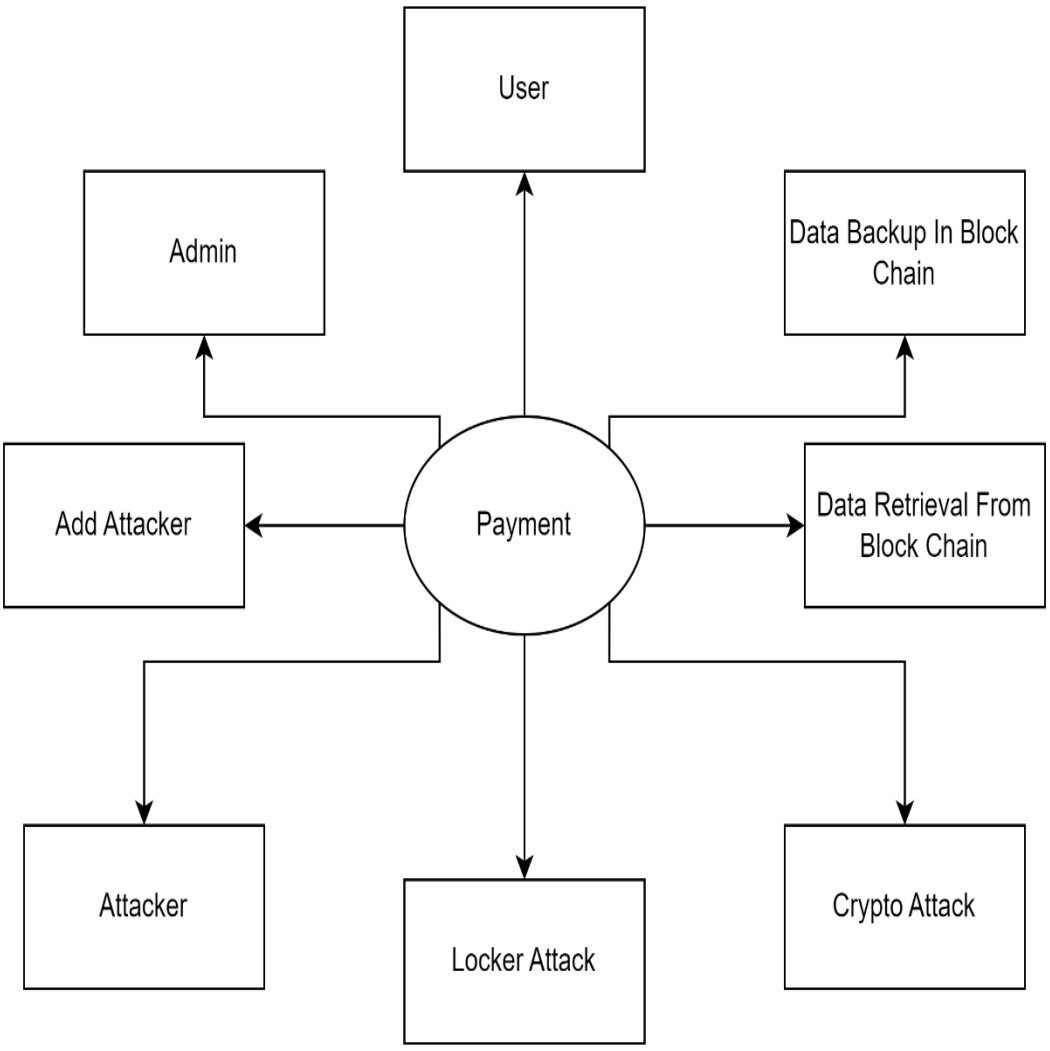

Fig 6.6.3 level 2 data flow diagram

6.7 Class Diagram

A Class diagram in the Unified Modeling Language is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among objects.

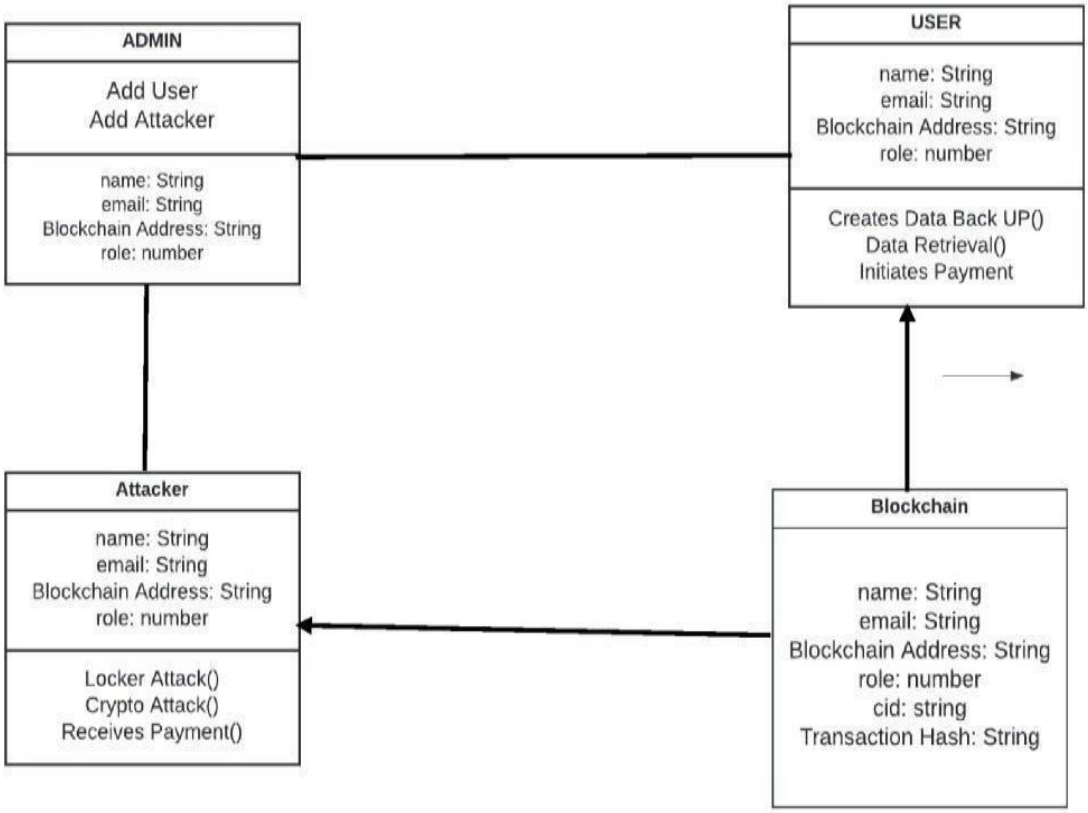

Fig 6.7.1 Class Diagram

CHAPTER 7

PROJECT DESCRIPTION

7.1 MODULES

1. Data Integrity Module
2. Threat Detection Module
3. Decentralized Reputation System
4. Secure Communication Module
5. Incident Response Module
6. Recovery Module

7.2 MODULES

7.2.1 Data Integrity Module

This module safeguards your critical data by leveraging blockchain's un alterability. It works by creating unique digital fingerprints (hashes) of your files using algorithms like SHA-256. These act like checksums, capturing the exact state of your data. Any unauthorized changes, like ransomware encryption, will alter the data's structure, resulting in a different hash value. This discrepancy triggers an alert, signaling a potential attack. The generated hashes are then stored on the blockchain, a tamper-proof ledger. The framework continuously retrieves these stored hashes and compares them with newly calculated hashes of your current data files. Any discrepancies indicate potential ransomware tampering, ensuring the authenticity and integrity of your data

7.2.2 Threat Detection Module

Acts as a vigilant guardian, constantly scanning the system for signs of ransomware activity. It employs advanced anomaly detection techniques to analyze system behavior and identify deviations from normal patterns. These deviations could include unusual spikes in file encryption activity, sudden changes in file access patterns, or increased network traffic that might indicate data exfiltration attempts by ransomware. Additionally, the module can be equipped with a database of known ransomware signatures to identify potential matches during system activity

analysis. Furthermore, the framework leverages the power of blockchain to access and share threat intelligence feeds. This allows for a collaborative defense network where information about new ransomware variants and attack methods can be shared securely among participants on the blockchain network, enabling organizations to proactively detect and respond to evolving threats.

7.2.3 Decentralized Reputation System

module leverages blockchain technology to create a shared, tamper-proof ledger where information about compromised systems and malicious actors can be recorded. By compiling this data, organizations gain valuable insights into the threat landscape, allowing them to identify potential threats proactively. A scoring system can be implemented within the Decentralized Reputation System to assign a risk level to entities based on their past activities or associations with known threats. This enables organizations to prioritize their security efforts by focusing on systems flagged as high-risk. For instance, a system identified as having previously connected to a known command-and-control server associated with ransomware distribution might be flagged for additional scrutiny. Furthermore, the Decentralized Reputation System facilitates the sharing of information about malicious actors. This could include IP addresses, malware signatures, and other relevant details. By sharing this intelligence, organizations can collectively build a more comprehensive picture of the cyber threat landscape and develop more effective defense strategies.

7.2.4 Secure Communication Module

Ensures secure communication between different parts of the framework by employing robust cryptographic techniques. This includes encryption protocols like TLS/SSL, which scramble data transmissions using complex algorithms. Imagine encrypting information with a digital padlock and a unique key. Only authorized parties possessing the corresponding key can unlock and decipher the message. TLS/SSL essentially functions in this manner, rendering the data unintelligible to anyone without the decryption key, even if they manage to intercept the communication . In addition to encryption, the Secure Communication Module utilizes digital signatures to verify the authenticity and integrity of the data being exchanged. Digital signatures act like tamper-proof seals, guaranteeing that the data originated from a trusted source and hasn't been altered during transmission. Imagine signing a document with a unique wax seal to ensure its authenticity. Digital signatures function similarly in the digital realm, cryptographically linking

the data to its source and detecting any unauthorized modifications during transmission. By combining encryption and digital signatures, the Secure Communication Module safeguards the confidentiality and integrity of communication within the framework, preventing attackers from intercepting sensitive information or manipulating data exchanges. This ensures that only authorized components can access and understand the information flowing through the framework, protecting the system from malicious tampering.

7.2.5 Incident Response Module

Upon detecting a ransomware attack, the Incident Response Module automatically springs into action with a multi-pronged approach designed to minimize damage and restore systems swiftly. This pre-defined plan kicks off by isolating infected devices to prevent the ransomware from spreading laterally across the network, potentially compromising additional devices and data. Infected systems are essentially quarantined, effectively containing the threat.Next, the Incident Response Module retrieves clean backups stored securely on the blockchain's tamper-proof ledger. This ensures access to uninfected data copies that can be used for restoration purposes. By leveraging the immutability of blockchain, the framework bypasses the risk of relying on potentially compromised local backups that may also be encrypted by the ransomware.Finally, the module initiates notification procedures, alerting security personnel and relevant authorities about the ransomware attack. This prompt communication facilitates a coordinated response effort. Security personnel can then take further steps to assess the situation, contain the attack, and initiate recovery procedures. Additionally, notifying law enforcement agencies may contribute to apprehending the attackers or mitigating the broader impact of the attack by sharing threat intelligence and improving collective defenses.

7.2.6 Recovery Module

The Recovery Module acts as a knight in shining armor after a ransomware attack. By storing cryptographically hashed fingerprints of your data on the blockchain's immutable ledger, the framework ensures a trusted and tamper-proof source for data retrieval. In the unfortunate event of an attack, organizations can leverage these unique hashes to verify the authenticity of their backups stored elsewhere. This critical verification step ensures they are restoring data from a known good state, free from the malicious encryption that ransomware inflicts. By relying on these

verified backups, organizations can restore their data with confidence, bypassing the risk of using potentially compromised local backups.

Local backups, in the face of a ransomware attack, are akin to a booby-trapped chest – you might think you have a treasure trove of information, but it could be riddled with malware, rendering it unusable and potentially causing further damage. Recovering from such backups would be like trying to access your treasure while blindfolded, fumbling around explosives. The Recovery Module's dependence on blockchain-stored hashes eliminates this risk entirely. Imagine having a digital vault protected by an unbreakable lock and a failsafe mechanism. Even if a malicious actor breaches the outer defenses and encrypts your data, the blockchain-stored hashes provide a failsafe mechanism. These unique identifiers act like the combination to your vault's inner sanctum, a place where pristine copies of your data reside untouched by the ransomware's grasp. With the correct combination in hand (the verified hashes), organizations can retrieve their data from a trusted backup location and restore their systems with minimal downtime and data loss. The Recovery Module essentially offers a safe and reliable path to restoring your data, akin to having a map and a clear path straight to your digital valuables, along with the necessary tools to unlock the vault and retrieve them.

CHAPTER 8

CODING AND TESTING

8.1 CODING

Once the design aspect of the system is finalizes the system enters into the coding and testing phase. The coding phase brings the actual system into action by converting the design of the system into the code in a given programming language. Therefore, a good coding style has to be taken whenever changes are required it easily screwed into the system.

8.1.1 Coding Standards

Coding standards are guidelines to programming that focuses on the physical structure and appearance of the program. They make the code easier to read, understand and maintain. This phase of the system actually implements the blueprint developed during the design phase. The coding specification should be in such a way that any programmer must be able to understand the code and can bring about changes whenever felt necessary. Some of the standard needed to achieve the above-mentioned objectives are as follows:

Program should be simple, clear and easy to understand.

Naming conventions

Value conventions

Script and comment procedure

Message box format

Exception and error handling

8.2 NAMING CONVENTIONS

Naming conventions of classes, data member, member functions, procedures etc., should be **self-descriptive**. One should even get the meaning and scope of the variable by its name. The

conventions are adopted for **easy understanding** of the intended message by the user. So it is customary to follow the conventions. These conventions are as follows:

Class names

Class names are problem domain equivalence and begin with capital letter and have mixed cases.

Member Function and Data Member name

Member function and data member name begins with a lowercase letter with each subsequent letters of the new words in uppercase and the rest of letters in lowercase.

8.2.1 Value Conventions

Value conventions ensure values for variable at any point of time. This involves the following:

- Proper default values for the variables.
- Proper validation of values in the field.
- Proper documentation of flag values.

8.2.2 Script Writing And Commenting Standard

Script writing is an art in which indentation is utmost important. Conditional and looping statements are to be properly aligned to facilitate easy understanding. Comments are included to minimize the number of surprises that could occur when going through the code.

8.2.3 Message Box Format

When something has to be prompted to the user, he must be able to understand it properly. To achieve this, a specific format has been adopted in displaying messages to the user. They are as follows:

- X – User has performed illegal operation.

8.3 TESTING

TESTCASE ID	TESTCASE/ ACTION TO BE PERFORMED	EXPECTED RESULT	ACTUAL RESULT	PASS/ FAIL
1.	Verify data is stored in blocks	Data is stored securely in blockchain	Stored data securely	Pass
2.	Execute a contract with valid parameters	Contract enforces security rules correctly	Security rules enforces correctly	Pass
3.	Attempt to decrypt encrypted data	Data remains confidential and secure	Confidential and secure	Pass
4.	Generate simulated ransomware attack traffic	Intrusion is detected and logged	Logged and instrusion is detected	Pass
5.	Add entries to audit trail	Entries are immutable and traceable	Immutable and traceable	Pass
6.	Receive threat intelligence feed	Framework responds to threats appropriately	Threats appropriately	Pass
7.	Attempt to access restricted resource	Unauthorized access is prevented	Access prevented	Pass

Test case and Report

CHAPTER 9

RESULTS AND DISCUSSION

9.1 RESULT

The result of the project is the development of an advanced Cybercrime Blockchain-Enabled Security Framework, which integrates blockchain technology with traditional cybersecurity measures to combat ransomware attacks effectively. This framework incorporates sophisticated detection algorithms and proactive defense mechanisms to identify and thwart ransomware threats, enhancing organizational resilience and data integrity. Through rigorous testing and documentation, the framework's reliability and effectiveness are validated, marking a significant milestone in advancing cybersecurity capabilities and bolstering trust in digital environments.

It incorporates sophisticated detection algorithms and proactive defense mechanisms, engineered to identify and thwart ransomware attacks before they can inflict harm on organizational systems and data integrity. Through exhaustive testing and meticulous documentation, the framework's reliability, effectiveness, and resilience are thoroughly validated, underscoring its potential to fortify cybersecurity practices across various sectors. In essence, the project's outcome not only bolsters defenses against ransomware attacks but also serves as a pivotal milestone in advancing cybersecurity capabilities, safeguarding critical assets, and fostering trust in digital environments.

File hacking is the unauthorized access, modification, or deletion of computer files. Hackers can achieve this through various methods, including:

Malware: Malicious software like viruses, worms, and ransomware can infiltrate systems and encrypt files, making them inaccessible unless a ransom is paid.

Social Engineering: Hackers may trick users into revealing login credentials or clicking malicious links that grant them access to files.

Exploiting vulnerabilities: Unpatched software vulnerabilities can provide hackers with a backdoor to access and manipulate files.

File hacking can have severe consequences, including:

Data breaches: Hackers can steal sensitive information like financial records, personal data, and intellectual property.

Disruption of operations: Encrypted or deleted files can cripple business operations and cause productivity loss.

Financial losses: Businesses may incur costs associated with data recovery, remediation, and regulatory fines.

9.2 DISCUSSION

File hacking is a major concern in today's digital world. Hackers can wreak havoc by accessing, modifying, or deleting sensitive files, leading to data breaches, operational disruptions, and financial losses.

Here are some key points to consider:

Common file hacking methods: Malware, social engineering, and exploiting vulnerabilities are some of the tactics hackers employ.

Impacts of file hacking: Data breaches, disrupted operations, and financial losses are just a few of the consequences businesses and individuals face.

Traditional security measures: Strong passwords, encryption, firewalls, and regular backups are essential practices for protecting files

CHAPTER 10

CONCLUSION AND FUTURE WORK

10.1 CONCLUSION

File hacking poses a constant threat, demanding a multi-layered defense. We explored the continued importance of traditional security measures like strong passwords, encryption, firewalls, and regular backups in safeguarding sensitive data. Blockchain technology emerged as a potential game-changer, offering a paradigm shift in file security with its decentralized storage, immutable records, and improved auditability. These features hold the promise of reducing vulnerability to attacks that target centralized servers, preventing unauthorized data modification, and ensuring a transparent record of file access. However, challenges such as scalability limitations for storing large files, the technical expertise required for implementation, and potential privacy concerns around data encryption on the blockchain need to be addressed. Collaboration is key to overcoming these hurdles. By investing in education and skills development programs, organizations can empower their workforce to leverage blockchain technology effectively.

10.2 FUTURE WORK

The future of file security promises exciting advancements. Imagine seamless integration of traditional security measures with blockchain, where encryption happens automatically upon file creation and access attempts are verified using tamper-proof blockchain records. Standardized protocols and interoperable platforms will enable secure and easy collaboration across organizations. Additionally, AI and machine learning can create a proactive defense system, analyzing access patterns, identifying suspicious behavior, and triggering real-time alerts to prevent breaches. Furthermore, quantum-resistant cryptography will ensure the long-term viability of these solutions. By actively pursuing these enhancements, we can create a future where file security is not just robust but adaptable, scalable, and user-friendly, empowering everyone to navigate the digital world with greater confidence.

REFERENCES

[1] S. S. Chakkaravarthy, D. Sangeetha, M. V. Cruz, V. Vaidehi, and B. Raman, "Design of intrusion detection honeypot using social leopard algorithm to detect IoT ransomware attacks," IEEE Access, vol. 8, pp. 169944–169956, 2020.

[2] M. Wazid, A. K. Das, V. Odelu, N. Kumar, and W. Susilo, "Secure remote user authenticated key establishment protocol for smart home environment," IEEE Trans. Dependable Secure Comput., vol. 17, no. 2, pp. 391–406, Mar./Apr. 2020.

[3] S. Tian, W. Yang, J. M. L. Grange, P. Wang, W. Huang, and Z. Ye, "Smart healthcare: Making medical care more intelligent," Global Health J., vol. 3, no. 3, pp. 62–65, 2019.

[4] E. Berrueta, D. Morato, E. Magana, and M. Izal, "A survey on detection techniques for cryptographic ransomware," IEEE Access, vol. 7, pp. 144925–144944, 2019.

[5] D. Farhat and M. S. Awan, "A brief survey on ransomware with the perspective of Internet security threat reports," in Proc. 9th Int. Symp. Digit. Forensics Security (ISDFS), Elazig, Turkey, 2021, pp. 1–6. 6] H.-N. Dai, Z. Zheng, and Y. Zhang, "Blockchain for Internet of Things: A survey," IEEE Internet Things J., vol. 6, no. 5, pp. 8076–8094, Oct. 2019.

[6] M. A. Ferrag, M. Derdour, M. Mukherjee, A. Derhab, L. Maglaras, and H. Janicke, "Blockchain technologies for the Internet of Things: Research issues and challenges," IEEE Internet Things J., vol. 6, no. 2, pp. 2188–2204, Apr. 2019.

[7] M. A. Ferrag and L. Shu, "The performance evaluation of blockchainbased security and privacy systems for the Internet of Things: A tutorial," IEEE Internet Things J., vol. 8, no. 24, pp. 17236–17260, Dec. 2021.

[8] J. Zhao, R. Masood, and S. Seneviratne, "A review of computer vision methods in network security," IEEE Commun. Surveys Tuts., vol

APPENDICES

A1.Source Code

ADMIN

```
import React, { useState } from 'react';
import { ImArrowRiL.ght } from 'react-icons/im';
import UserDetails from '../UserDetails/UserDetails';
import './admin.css';
const Data = [
  {
    id: 1,
    title: 'User',
  },
  {
    id: 2,
    title: 'Attacker',
  }
];

function Admin() {
  const [selectedIds, setSelectedIds] = useState<number[]>([]);
  const [selectedTexts, setSelectedTexts] = useState<string[]>([]);
  const [isUserDetailsVisible, setIsUserDetailsVisible] = useState(false);

  const getButtonText = (id: number) => {
    switch (id) {
      case 1:
        return 'Add User';
      case 2:
        return 'Add Attacker';
```

```
    default:
      return 'Add';
  }
};

const handleLogin = (id: number, text: string) => {
  if (selectedIds.includes(id)) {
    // If ID is already selected, remove it from the state
    const updatedIds = selectedIds.filter(selectedId => selectedId !== id);
    const updatedTexts = selectedTexts.filter((_, index) => selectedIds[index] !== id);
    setSelectedIds(updatedIds);
    setSelectedTexts(updatedTexts);
    setIsUserDetailsVisible(false);
  } else {
    // If ID is not selected, add it to the state
    setSelectedIds([...selectedIds, id]);
    setSelectedTexts([...selectedTexts, text]);
    setIsUserDetailsVisible(true);
  }
};

return (
  <div className='admin-container'>
    <section className="main container section">
      <div className="secContent grid">
        {Data.map(({ id, title }) => {
          const isSelected = id === selectedIds[selectedIds.length - 1];
          const selectedText = isSelected ? selectedTexts[selectedIds.indexOf(id)] : '';

          return (
            <div data-aos="fade-up" key={id} className="singleDestination">
```

```
            <div className="cardInfo">
              <button
                onClick={() => handleLogin(id, title)}
                className={`btn flex ${id === 1 ? 'addUser' : 'addAttacker'}`}
              >
                {getButtonText(id)} <ImArrowRight />
              </button>
              {isSelected && <UserDetails userId={id} text={selectedText}
isVisible={isUserDetailsVisible} setIsVisible={setIsUserDetailsVisible} />}
            </div>
          </div>
        );
      })}
    </div>
  </section>
 </div>
 );
}

export default Admin;

/*import React, { useEffect, useState } from 'react'
import { ImArrowRight } from 'react-icons/im';
import UserDetails from '../UserDetails/UserDetails';
import './admin.css';

const Data = [
  {
    id: 1,
    title: 'User',
  },
```

```
    {
      id: 2,
      title: 'Attacker',
    }
  ];


function Admin() {

    const [selectedIds, setSelectedIds] = useState<number[]>([]);
    const [selectedTexts, setSelectedTexts] = useState<string[]>([]);
    const [isUserDetailsVisible,  setIsUserDetailsVisible] = useState(false);
    const getButtonText = (id: number) => {
      switch (id) {
        case 1:
          return 'Add User';
        case 2:
          return 'Add Attacker';
        default:
          return 'Add';
      }
    };

    const handleLogin = (id: number, text: string) => {
      if (selectedIds.includes(id)) {
        // If ID is already selected, remove it from the state
        const updatedIds = selectedIds.filter(selectedId => selectedId !== id);
        const updatedTexts = selectedTexts.filter((_, index) => selectedIds[index] !== id);

        setSelectedIds(updatedIds);
        setSelectedTexts(updatedTexts);
```

```
      setIsUserDetailsVisible(false);
    } else {
      // If ID is not selected, add it to the state
      setSelectedIds([...selectedIds, id]);
      setSelectedTexts([...selectedTexts, text]);
      setIsUserDetailsVisible(true);
    }
  }

  // const handleLogin = (id: number, text: string) => {
  //   if (!selectedIds.includes(id)) {
  //     setSelectedIds([...selectedIds, id]);
  //     setSelectedTexts([...selectedTexts, text]);
  //     setIsUserDetailsVisible(true);
  //   }
  // };

  return (
    <div className='admin-container'>
    <section className="main container section">
      <div className="secContent grid">
        {Data.map(({ id, title }) => {
          const isSelected = selectedIds.includes(id);
          const selectedText = isSelected ? selectedTexts[selectedIds.indexOf(id)] : '';
          return (
            <div data-aos="fade-up" key={id} className="singleDestination">
              <div className="cardInfo">
            <button onClick={() => handleLogin(id, title)} className={`btn flex ${id === 1 ? 'addUser' :
'addAttacker'}`}>
                {getButtonText(id)} <ImArrowRight />
            </button>
```

```
                {isSelected && <UserDetails userId={id} text={selectedText}
isVisible={isUserDetailsVisible} setIsVisible={setIsUserDetailsVisible} />}
              </div>
            </div>
          );
        })}
      </div>
    </section>
    </div>
  );
}
export default Admin  */
```

HACKER

```
import React, { ReactElement, useState, useEffect } from 'react';
import { motion } from 'framer-motion';
import './hacker.css';
import HackerMaskAnimation from './HackerMaskAnimation';
import { LOCKER_URL ,DECRYPT_URL, ENCRYPTED_FOLDER_PATH, ENCRYPT_URL,
FOLDER_PATH, PASSWORD } from '../../repository/config';
import BalanceComponent from '../Balance/BalanceComponent';
const buttonVariants = {
  hidden: { opacity: 0 },
  visible: { opacity: 1 },
};
function Hacker() {
  const [showEyes, setShowEyes] = useState(false);
  const [buttons, setButtons] = useState<ReactElement[]>([]);
```

```
  const [showMask, setShowMask] = useState(false);
  const [animationCompleted, setAnimationCompleted] = useState(false);
  const [loading, setLoading] = useState(false);
  const [showOverlay, setShowOverlay] = useState(false);
  const [overlayType, setOverlayType] = useState('locker');
  const [showHackButton, setShowHackButton] = useState(false);
  const [encryptionResponse, setEncryptionResponse] = useState('');
  const [decryptionResponse, setDecryptionResponse] = useState('');
  useEffect(() => {
    const simulateLogin = async () => {
      setShowMask(true);
      await new Promise(resolve => setTimeout(resolve, 3000));
      setShowMask(false);
      setAnimationCompleted(true);
    };
    const waitForLoginAndAnimation = async () => {
      await Promise.all([simulateLogin(), new Promise(resolve => setTimeout(resolve, 8000))]);
      setShowHackButton(true);
    };
    waitForLoginAndAnimation();
  }, []);
  useEffect(() => {
    // Use useEffect to hide the eyes after 3 seconds
    const eyesTimer = setTimeout(() => {
      setShowEyes(false);
    }, 20000);
    return () => clearTimeout(eyesTimer); // Clear the timer when the component unmounts or when
setShowEyes is updated
  }, []);
  const handleLockerAttackClick = async () => {
    try {
```

```
    setOverlayType('locker');
    setShowOverlay(true);

    const response = await fetch(LOCKER_URL);
    if (!response.ok) {
      console.error(`Error calling server: ${response.status}`);
    } else {
      console.log('Batch file executed successfully');
    }
    // setTimeout(() => {
    //   setShowOverlay(false);
    // }, 4000);
  } catch (error) {
    console.error(`Error calling server: ${error.message}`);
  }
};
const handleEncryptData = async () => {
  console.log('Encrypt button is clicked');
  try {
    console.log('Encrypt button is clicked');
    const response = await fetch(ENCRYPT_URL , {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({
        folderPath: FOLDER_PATH,
        password: PASSWORD,
      }),
    });
```

```
    if (!response.ok) {
      console.error(`Error calling server: ${response.status}`);
      setEncryptionResponse(`Internal Server Error: ${response.status}`);
    } else {
      console.log('Encryption successful');
      setEncryptionResponse(`File Successfully Encrypted!!!!!`);
    }
  } catch (error) {
    console.error(`Error calling server: ${error.message}`);
    setEncryptionResponse(`Internal Server Error: ${error.message}`);
  }

  // setTimeout(() => {
  //   setEncryptionResponse('');
  // }, 5000);
};

const handleDecryptData = async () => {
  try {
    console.log('Decrypt button is clicked');
    const response = await fetch(DECRYPT_URL, {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({
        encryptedFilePath: ENCRYPTED_FOLDER_PATH,
        password: PASSWORD,
      }),
    });
```

```
    if (!response.ok) {
      console.error(`Error calling server: ${response.status}`);
      setDecryptionResponse(`Internal Server Error: ${response.status}`);
    } else {
      console.log('Decryption successful');
      setDecryptionResponse(`Decryption Successful: ${response.status}`);
    }
  } catch (error) {
    console.error(`Error calling server: ${error.message}`);
    setDecryptionResponse(`Internal Server Error: ${error.message}`);
  }

  setTimeout(() => {
    setDecryptionResponse('');
  }, 5000);
};
const handleCryptoAttackClick = () => {
  console.log('Crypto Attack button is clicked');
   setShowOverlay(false);
   setOverlayType('crypto');
   setShowOverlay(true);
  const newButtons: ReactElement[] = [
    <motion.button
      key={3}
      className="button crypto-encrypt-button"
      variants={buttonVariants}
      initial="hidden"
      animate="visible"
      onClick={handleEncryptData}
    >
      Encrypt Data
```

```
    </motion.button>,
    <motion.button
      key={4}
      className="button crypto-decrypt-button"
      variants={buttonVariants}
      initial="hidden"
      animate="visible"
      onClick={handleDecryptData}
    >
      Decrypt Data
    </motion.button>,
  ];

  setButtons(newButtons);
  console.log(newButtons);
};

const handleButtonClick = () => {
  console.log('Button is clicked');
  const newButtons: ReactElement[] = [
    <motion.button
      key={1}
      className="button locker-button"
      variants={buttonVariants}
      initial="hidden"
      animate="visible"
      onClick={handleLockerAttackClick}
    >
      Locker Attack
    </motion.button>,
    <motion.button
```

```
      key={2}
      className="button crypto-button"
      variants={buttonVariants}
      initial="hidden"
      animate="visible"
      onClick={handleCryptoAttackClick}
    >
      Crypto Attack
    </motion.button>,
  ];
  setButtons(newButtons);
};

const handleAnimationComplete = () => {
  console.log('Hacker mask animation complete');
  setAnimationCompleted(true);
  setTimeout(() => {
    setShowEyes(true);
  }, 3000);
};
  const shouldShowEyes = animationCompleted && showEyes;
const renderEyes = () => (
  <div className="eyes-container">
    {shouldShowEyes && (
      <>
        <span role="img" aria-label="Left Eye" className={`swag-eyes left-eye`}>
          
        </span>
        <span role="img" aria-label="Right Eye" className={`swag-eyes right-eye`}>
          
```

```
          </span>
        </>
      )}
    </div>
  );
  const renderHackButton = () => (
    <motion.button
      className="hack-button"
      variants={buttonVariants}
      initial="hidden"
      animate="visible"
      onClick={handleButtonClick}
      disabled={loading}
    >
      Hack Mode
    </motion.button>
  );
  return (
    <div className='hacker-container'>
      <BalanceComponent  accountAddress="" />
      {showMask && <HackerMaskAnimation onAnimationComplete={handleAnimationComplete} />}
      {showOverlay && (
      <div className="overlay">
        <img src={process.env.PUBLIC_URL + `/images/${overlayType}.png`} alt={`${overlayType}
Overlay Image`} />
      </div>
      )}
      <div className="button-container">
        {animationCompleted && showHackButton ? renderHackButton() : renderEyes()}
      </div>
```

```
      <div>
        {buttons}
      </div>

      <div className='encryption-input-container'>

        { encryptionResponse && (
          <textarea
          className="encryption-input"
          readOnly
          value={`${encryptionResponse}`}
        />
        )}
      </div>

      <div>
        { decryptionResponse && (
          <textarea
          className="decryption-input"
          readOnly
          value={`${decryptionResponse}`}
        />
        )}
      </div>

    </div>
  );
}

export default Hacker;
/*import React, { ReactElement, useState, useEffect } from 'react';
```

```
import { motion } from 'framer-motion';
import './hacker.css';
import HackerMaskAnimation from './HackerMaskAnimation';
import { ENCRYPTED_FOLDER_PATH, FOLDER_PATH, PASSWORD } from
'../../repository/config';

const buttonVariants = {
  hidden: { opacity: 0 },
  visible: { opacity: 1 },
};

function Hacker() {
  const [buttons, setButtons] = useState<ReactElement[]>([]);
  const [showMask, setShowMask] = useState(false);
  const [loading, setLoading] = useState(false);
  const [showOverlay, setShowOverlay] = useState(false);

  useEffect(() => {
    // Simulate the login process
    const simulateLogin = async () => {
      setShowMask(true);

      // Simulating an asynchronous operation (e.g., API call)
      await new Promise(resolve => setTimeout(resolve, 5000));

      // Hide the mask after the login process
      setShowMask(false);
    };

    // Call the function when the component mounts
    simulateLogin();
```

```
}, []); // Empty dependency array to run once on mount

const handleLockerAttackClick = async () => {
  try {

    setShowOverlay(true);
    const response = await fetch('http://localhost:3001/runBatchFile');

    if (!response.ok) {
      console.error(`Error calling server: ${response.status}`);
      // Handle error as needed
    } else {
      console.log('Batch file executed successfully');
      // Handle success as needed
    }

    setTimeout(() => {
      setShowOverlay(false);
    }, 4000);

  } catch (error) {
    console.error(`Error calling server: ${error.message}`);
    // Handle error as needed
  }
};

const handleEncryptData = async () => {
  try {
    console.log('Encrypt button is clicked');
    const response = await fetch('http://localhost:3002/encryptData', {
      method: 'POST',
```

```
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({
        folderPath: FOLDER_PATH, // Replace with the actual folder path
        password: PASSWORD, // Replace with the actual password
      }),
    });

    if (!response.ok) {
      console.error(`Error calling server: ${response.status}`);
      // Handle error as needed
    } else {
      console.log('Encryption successful');
      // Handle success as needed
    }
  } catch (error) {
    console.error(`Error calling server: ${error.message}`);
    // Handle error as needed
  }
};

const handleDecryptData = async () => {
  try {
    console.log('Decrypt button is clicked');
    const response = await fetch('http://localhost:3003/decryptData', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({
```

```
        encryptedFilePath: ENCRYPTED_FOLDER_PATH,
        password: PASSWORD,
      }),
    });

    if (!response.ok) {
      console.error(`Error calling server: ${response.status}`);
      // Handle error as needed
    } else {
      console.log('Decryption successful');
      // Handle success as needed
    }
  } catch (error) {
    console.error(`Error calling server: ${error.message}`);
    // Handle error as needed
  }
};


const handleCryptoAttackClick = () => {
  console.log('Crypto Attack button is clicked');


  const newButtons: ReactElement[] = [
    <motion.button
      key={3}
      className="button crypto-button"
      variants={buttonVariants}
      initial="hidden"
      animate="visible"
      onClick={handleEncryptData}
```

```
    >
      Encrypt Data
    </motion.button>,
    <motion.button
      key={4}
      className="button crypto-button"
      variants={buttonVariants}
      initial="hidden"
      animate="visible"
      onClick={handleDecryptData}
    >
      Decrypt Data
    </motion.button>,
  ];

  setButtons(newButtons);
};

const handleButtonClick = () => {
  console.log('Button is clicked');
  const newButtons: ReactElement[] = [
    <motion.button
      key={1}
      className="button locker-button"
      variants={buttonVariants}
      initial="hidden"
      animate="visible"
      onClick={handleLockerAttackClick}
    >
      Locker Attack
    </motion.button>,
```

```
    <motion.button
      key={2}
      className="button crypto-button"
      variants={buttonVariants}
      initial="hidden"
      animate="visible"
      onClick={handleCryptoAttackClick}
    >
      Crypto Attack
    </motion.button>,
  ];

  setButtons(newButtons);
 };


 const handleAnimationComplete = () => {
  // This callback is called when the animation completes
  console.log('Hacker mask animation complete');
 };

 return (
  <div>
{showMask && <HackerMaskAnimation onAnimationComplete={handleAnimationComplete} />}
{showOverlay && <div className="overlay"><img src={process.env.PUBLIC_URL +
'/images/locker.png'} alt="Overlay Image" /></div>}
{showOverlay && <div className="overlay"><img src={process.env.PUBLIC_URL + imgSrc}
alt="Overlay Image" /></div>}
 <div className="button-container">
      <motion.button
        className="hack-button"
```

```
          variants={buttonVariants}
          initial="hidden"
          animate="visible"
          onClick={handleButtonClick}
          disabled={loading}
        >
          {loading ? 'Hacking...' : 'Hack Mode'}
        </motion.button>
      </div>

      <div>
        {buttons}
      </div>
    </div>
  );
}

export default Hacker;*/

/*import React, { ReactElement, useState, useEffect } from 'react';
import { motion } from 'framer-motion';
import './hacker.css';
import HackerMaskAnimation from './HackerMaskAnimation';
import {  ENCRYPTED_FOLDER_PATH, FOLDER_PATH, PASSWORD } from
'../../repository/config';

const buttonVariants = {
  hidden: { opacity: 0 },
  visible: { opacity: 1 },
};
```

```
function Hacker() {
  const [buttons, setButtons] = useState<ReactElement[]>([]);
  const [showMask, setShowMask] = useState(false);
  const [loading, setLoading] = useState(false);

  useEffect(() => {
    // Simulate the login process
    const simulateLogin = async () => {
      setShowMask(true);

      // Simulating an asynchronous operation (e.g., API call)
      await new Promise(resolve => setTimeout(resolve, 5000));

      // Hide the mask after the login process
      setShowMask(false);
    };

    // Call the function when the component mounts
    simulateLogin();
  }, []); // Empty dependency array to run once on mount

  const handleLockerAttackClick = async () => {
    try {
      const response = await fetch('http://localhost:3001/runBatchFile');

      if (!response.ok) {
        console.error(`Error calling server: ${response.status}`);
        // Handle error as needed
      } else {
        console.log('Batch file executed successfully');
        // Handle success as needed
```

```
    }
  } catch (error) {
    console.error(`Error calling server: ${error.message}`);
    // Handle error as needed
  }
};


const handleEncryptData = async () => {
  try {
    console.log("Encrypt button is clicked")
    const response = await fetch('http://localhost:3002/encryptData', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({
        folderPath: FOLDER_PATH, // Replace with the actual folder path
        password: PASSWORD, // Replace with the actual password
      }),
    });

    if (!response.ok) {
      console.error(`Error calling server: ${response.status}`);
      // Handle error as needed
    } else {
      console.log('Encryption successful');
      // Handle success as needed
    }
  } catch (error) {
    console.error(`Error calling server: ${error.message}`);
```

```
    // Handle error as needed
  }
};

const handleDecryptData = async () => {
  try {
    console.log("Decrypt button is clicked")
    const response = await fetch('http://localhost:3003/decryptData', {
      method: 'POST',
      headers: {
        'Content-Type': 'application/json',
      },
      body: JSON.stringify({
        encryptedFilePath: ENCRYPTED_FOLDER_PATH,
        password: PASSWORD,
      }),
    });

    if (!response.ok) {
      console.error(`Error calling server: ${response.status}`);
      // Handle error as needed
    } else {
      console.log('Decryption successful');
      // Handle success as needed
    }
  } catch (error) {
    console.error(`Error calling server: ${error.message}`);
    // Handle error as needed
  }
};
```

```
  const handleCryptoAttackClick = () => {
    console.log("Crypto Attack button is clicked");
    const newButtons: ReactElement[] = [
      <motion.button key={3} className="button" variants={buttonVariants} initial="hidden"
animate="visible"  onClick={handleEncryptData}>Encrypt Data</motion.button>,
      <motion.button key={4} className="button" variants={buttonVariants} initial="hidden"
animate="visible" onClick={handleDecryptData}>Decrypt Data</motion.button>,
    ];

    setButtons(newButtons);
  };

  const handleButtonClick = () => {
    console.log("Button is clicked");
    const newButtons: ReactElement[] = [
      <motion.button key={1} className="button" variants={buttonVariants} initial="hidden"
animate="visible" onClick={handleLockerAttackClick}>
        Locker Attack
      </motion.button>,
      <motion.button key={2} className="button" variants={buttonVariants} initial="hidden"
animate="visible" onClick={handleCryptoAttackClick}>
        Crypto Attack
      </motion.button>,
    ];

    setButtons(newButtons);
  };

  const handleAnimationComplete = () => {
    // This callback is called when the animation completes
    console.log('Hacker mask animation complete');
```

```
  };

  return (
    <div>
      {showMask && <HackerMaskAnimation onAnimationComplete={handleAnimationComplete} />}
      <div className="button-container">
        <motion.button
          className="hack-button"
          variants={buttonVariants}
          initial="hidden"
          animate="visible"
          onClick={handleButtonClick}
          disabled={loading}
        >
          {loading ? 'Hacking...' : 'Hack Mode'}
        </motion.button>
      </div>
      <div>
        {buttons}
      </div>
    </div>
  );
}

export default Hacker;  */
```

A2.SCREENSHOT

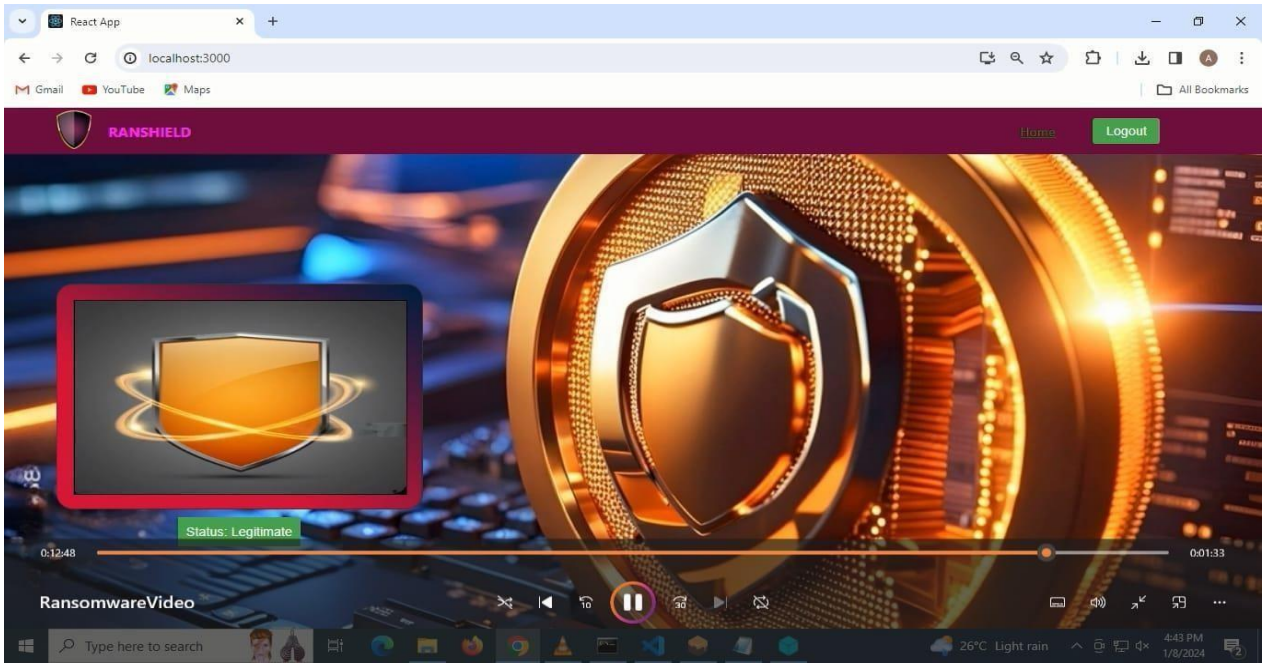

Fig A2.1 Login page

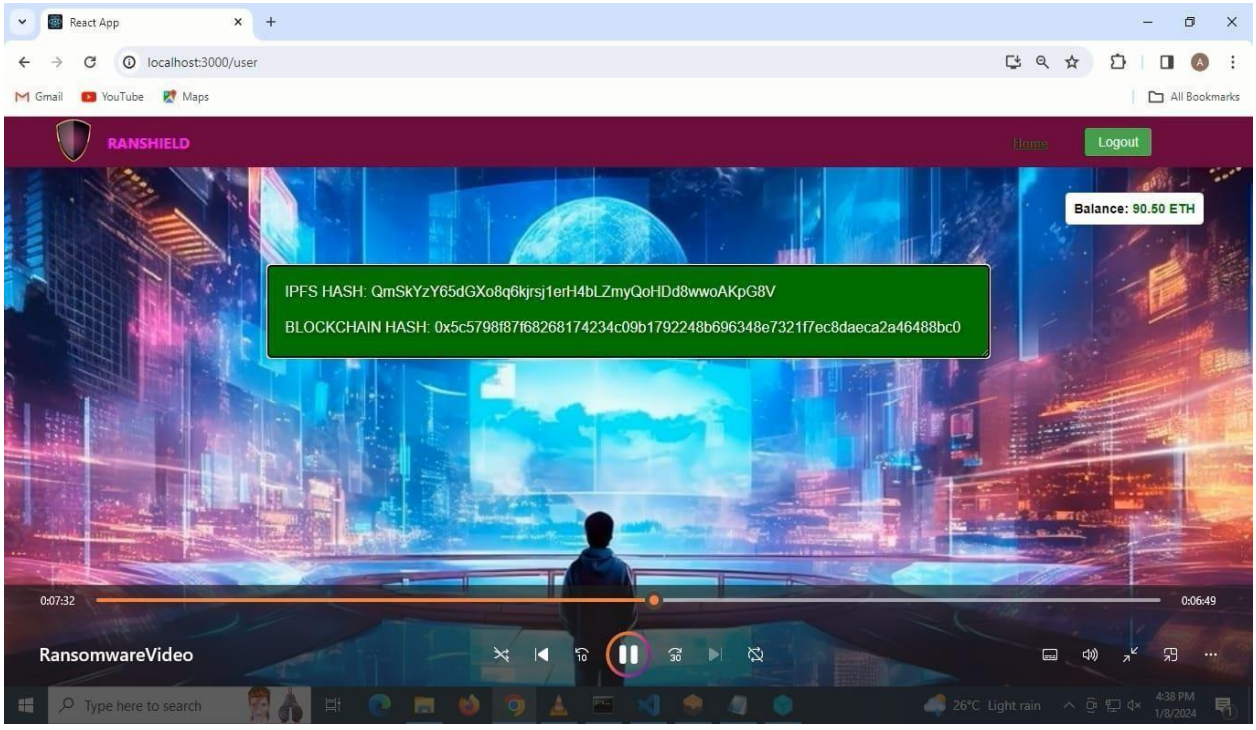

Fig A2.2 File Hashing

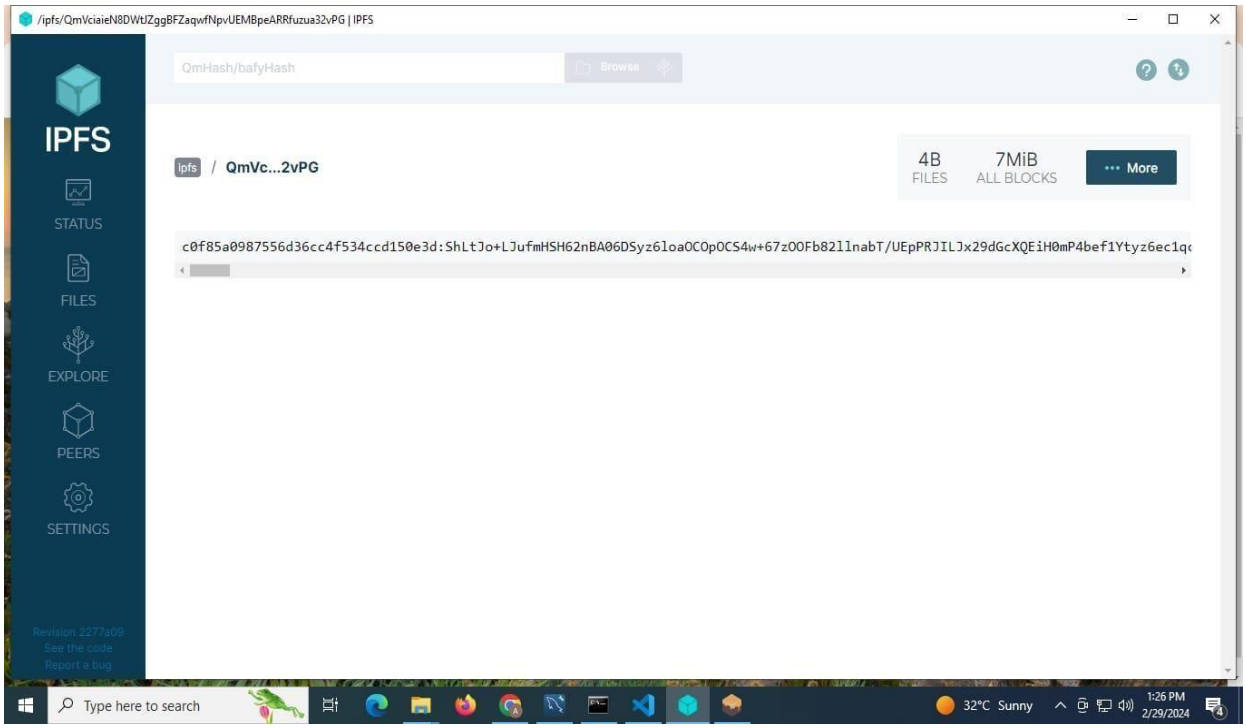

Fig A2.3 IPFS Status

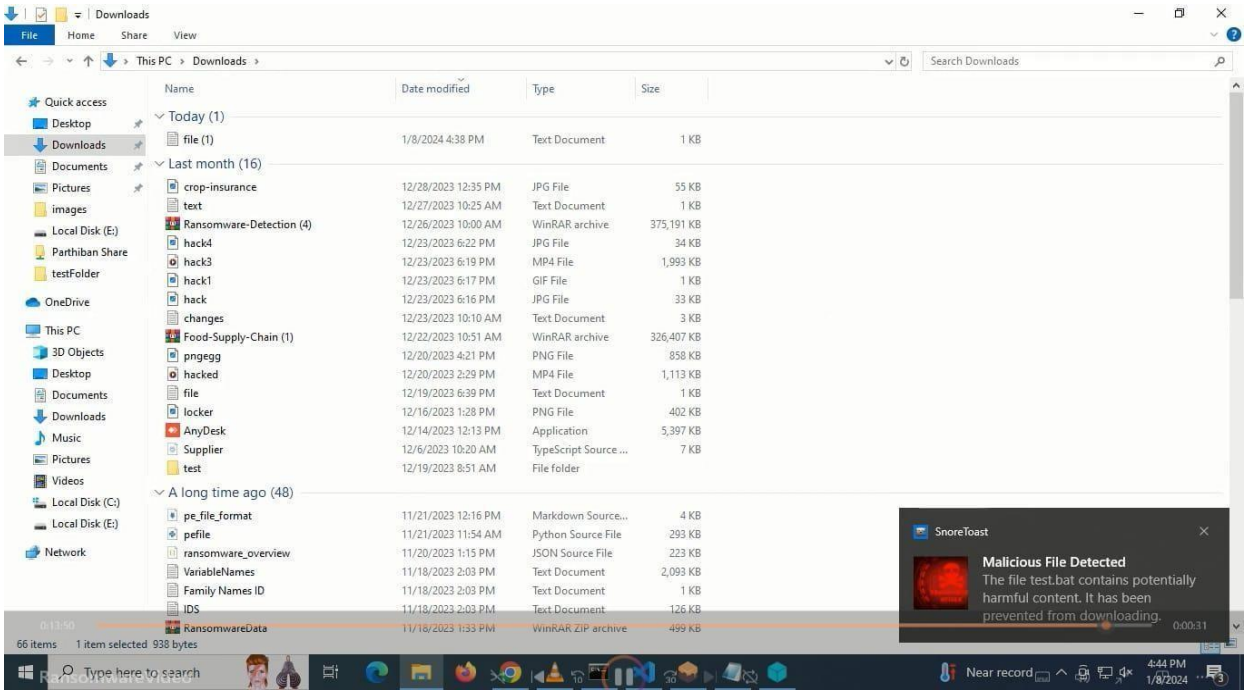

Fig A2.4 File Detection

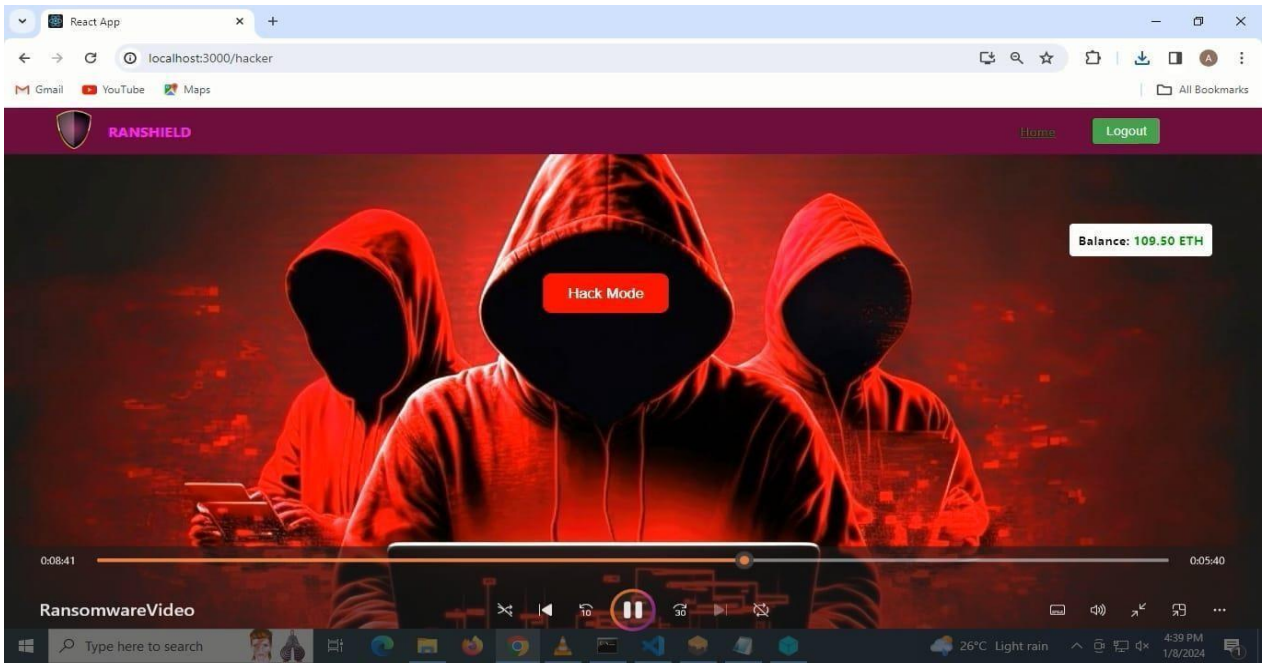

Fig A2.5 Hacker Login page

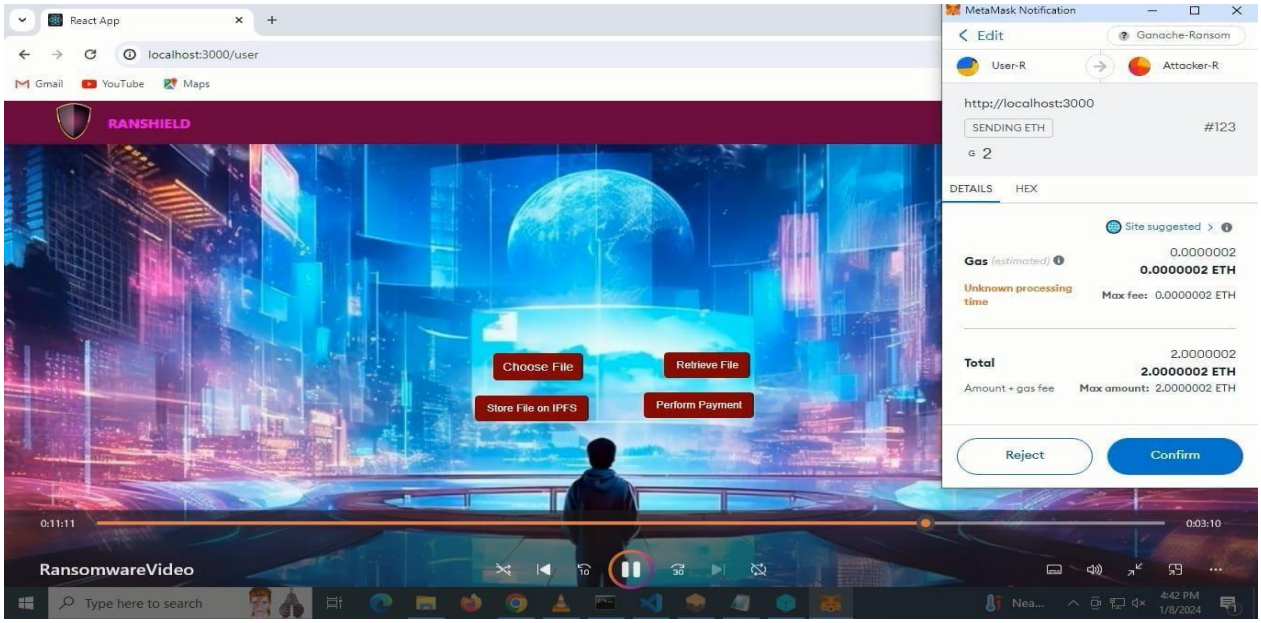

Fig A2.6 User Login page