**CHAPTER 1**

**INTRODUCTION**

This chapter provided the overview of network security challenges, Machine Learning for Network Traffic Classification, Blockchain Technology for Encryption Key Management, Cryptography Libraries for Data Encryption, problem statement, objectives, purpose and scope.

* 1. **Background**

**1.1.1 Overview of Network Security Challenges**

The landscape of network security is constantly evolving, with cyber threats becoming increasingly sophisticated and pervasive. From malware attacks to social engineering tactics, organizations face a multitude of challenges that can compromise the confidentiality, integrity, and availability of their data. This section provides a detailed examination of the key challenges in network security, including the importance of threat detection, vulnerability management, and incident response strategies. By understanding these challenges, organizations can better prepare themselves to address security risks and protect their network infrastructure from potential breaches. In recent years, the rapid adoption of cloud computing technologies has revolutionized the way organizations manage and store their data. While the cloud offers numerous benefits such as scalability, cost-efficiency, and flexibility, it also introduces new security challenges. One of the most critical security threats facing cloud environments is privilege escalation attacks, where malicious actors exploit vulnerabilities to gain unauthorized access to sensitive data and resources.

Privilege escalation attacks are particularly dangerous because they allow attackers to access information and perform actions that are typically restricted to privileged users. This can include accessing sensitive data, executing malicious code, or even taking control of the entire cloud environment. Privilege escalation attacks can occur through various means, including exploiting vulnerabilities in the cloud infrastructure, leveraging insecure configurations, or compromising user credentials.

To protect against privilege escalation attacks, organizations must implement robust security measures at various levels of their cloud environment. This includes implementing strong authentication mechanisms, regularly updating and patching software, monitoring and logging all activities, and implementing least privilege principles to restrict access to sensitive resources.

One of the key challenges in mitigating privilege escalation attacks is the complexity of modern cloud environments. Cloud environments often consist of multiple layers, including infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS). Each of these layers introduces its own set of security challenges, making it difficult for organizations to secure their entire cloud environment effectively.

Cloud security is a critical aspect of modern IT infrastructure, as organizations increasingly rely on cloud services to store and process their data. Cloud security encompasses a range of practices, technologies, and policies designed to protect cloud-based systems, data, and infrastructure from threats such as unauthorized access, data breaches, and service disruptions. In this article, we will explore the key components of cloud security, the challenges it presents, and best practices for ensuring a secure cloud environment.

One of the fundamental aspects of cloud security is data protection. This includes ensuring that data is encrypted both in transit and at rest, so that even if it is intercepted or accessed by unauthorized parties, it remains secure. Encryption is a key tool in protecting sensitive data, and cloud service providers (CSPs) typically offer encryption services as part of their offerings.

Another important aspect of cloud security is identity and access management (IAM). IAM involves managing the identities of users who have access to the cloud environment, as well as the permissions and privileges they are granted. This helps to ensure that only authorized users have access to sensitive data and resources, and that they only have access to the data and resources they need to perform their jobs.

Additionally, cloud security involves securing the cloud infrastructure itself. This includes protecting against threats such as distributed denial of service (DDoS) attacks, which can disrupt cloud services and make them unavailable to users. CSPs typically have measures in place to protect against these types of attacks, but organizations should also implement their own security measures to further protect their data and resources.

One of the challenges of cloud security is the shared responsibility model. In a cloud environment, both the CSP and the customer are responsible for security. The CSP is responsible for securing the infrastructure that supports the cloud services, while the customer is responsible for securing the data and applications they run on the cloud. This can lead to confusion about who is responsible for what, so it is important for organizations to clearly understand their responsibilities and ensure that they are met.

Another challenge of cloud security is the dynamic nature of cloud environments. cloud environments are highly scalable and flexible, which means that they can change rapidly in response to demand. This can make it difficult to keep track of all the assets and data in the cloud environment, and to ensure that they are all properly secured. Additionally, the use of multiple cloud providers and services can further complicate security efforts.

**1.1.2 Machine Learning for Network Traffic Classification**

Machine Learning algorithms have revolutionized the field of cybersecurity by enabling automated analysis and classification of network traffic data. This section delves into the intricacies of Machine Learning for network traffic classification, exploring how algorithms can identify patterns, anomalies, and potential security threats in real-time. By leveraging Machine Learning models, organizations can enhance their ability to detect and respond to malicious activities within their networks, thereby strengthening their overall security posture and reducing the risk of data breaches. Machine Learning, a subset of artificial intelligence, has emerged as a powerful tool for enhancing cloud security. By leveraging the capabilities of Machine Learning algorithms, organizations can analyse vast amounts of data in real-time to identify patterns and anomalies indicative of privilege escalation attempts. This proactive approach enables security teams to respond swiftly to potential threats before they escalate into full-blown security breaches.

Machine Learning techniques have the potential to revolutionize cloud security by providing a more efficient and effective way to detect and prevent privilege escalation attacks. Traditional security measures often rely on predefined rules and signatures to identify threats, which can be ineffective against evolving attack strategies. Machine Learning, on the other hand, can adapt and learn from new data, making it more adept at detecting previously unseen threats.

One of the key advantages of Machine Learning in cloud security is its ability to analyse large volumes of data quickly and accurately. This is particularly important in cloud environments, where vast amounts of data are generated and processed every second. Machine Learning algorithms can sift through this data to identify patterns and anomalies that may indicate a privilege escalation attempt.

There are several ways in which Machine Learning can be applied to enhance cloud security. One approach is to use Machine learning algorithms to analyse user behaviour and identify deviations from normal patterns.

Another approach is to use Machine Learning for anomaly detection. Machine Learning algorithms can be trained on a dataset of normal behaviour to learn what normal activity looks like. Once trained, the algorithms can then detect deviations from this normal behaviour, which may indicate a privilege escalation attempt.

Machine Learning can also be used for threat intelligence. By analysing data from multiple sources, including threat intelligence feeds, machine Learning algorithms can identify emerging threats and proactively protect against them. This can help organizations stay one step ahead of attackers and prevent potential security breaches.

Machine Learning holds great promise for enhancing cloud security, there are also challenges and limitations that need to be addressed. One challenge is the need for high-quality data for training machine Learning algorithms. The effectiveness of machine Learning models depends heavily on the quality and quantity of the data used to train them.

Another challenge is the need for interpretability. Machine Learning algorithms can sometimes be seen as "black boxes," making it difficult for security teams to understand how they arrived at a particular conclusion. This lack of transparency can be a barrier to adoption, especially in security-critical environments where trust and accountability are paramount.

**1.1.3 Blockchain Technology for Encryption Key Management**

Blockchain technology offers a decentralized and secure solution for managing encryption keys, which are essential for protecting sensitive data from unauthorized access. This section delves into the innovative use of blockchain for encryption key management, highlighting its benefits such as transparency, immutability, and resilience to tampering. By leveraging blockchain technology, organizations can establish a trusted and tamper-proof system for storing and managing encryption keys, ensuring the confidentiality and integrity of their data assets. In today’s digital landscape, the need for secure file management solutions has become increasingly critical. The widespread adoption of cloud computing has transformed how organizations manage their data, but it has also brought about new cyber security challenges, particularly with regards to insider threats.

**1.1.4 Cryptography Libraries for Data Encryption**

Cryptography libraries play a pivotal role in securing data through encryption, ensuring that sensitive information remains confidential and protected from unauthorized access. This section delves into the use of cryptography libraries for implementing robust encryption techniques, securing communication channels, and safeguarding data at rest. By integrating cryptography into the system, organizations can establish a strong defense against data breaches, unauthorized disclosures, and other security threats, thereby enhancing the overall security of their networked environment. Malicious insiders pose a particularly serious threat to organizations, as they often have greater access and opportunities to cause significant damage. Unlike external threats, insiders already have privileged access to information and resources, making them harder to detect and mitigate. The existing system often lack robust security measures, leaving users vulnerable to data breaches and unauthorized access.

To address this challenge, the project aims to development of a secure file management system. This system will provide a user-friendly interface for uploading, sharing, and managing files which ensures the confidentiality, integrity and availability of data.

Advanced security features such as encryption, access control, and anomaly detection will be implemented to safeguard against cyber threats. Moreover, Machine learning algorithms will be used to detect and prevent privilege escalation attacks and fraudulent activities. This ensures a secure and efficient platform for file management, which enhances data security and user trust in digital file management systems.

* 1. **Problem Statement**

Privilege escalation attacks pose significant risks to organizations, including data breaches, reputational damage, and financial loss. These attacks can lead to unauthorized access to sensitive information, compromising the confidentiality and integrity of data. Moreover, successful privilege escalation attacks can damage an organization's reputation, eroding trust among customers and stakeholders. The financial implications of such attacks can be severe, including theft of sensitive information, disruption of services, and costs associated with recovering from the attack. The organizations may also face legal and regulatory consequences for failing to protect against privilege escalation attacks, such as fines, lawsuits, and damage to their brand.

**1.3 Objectives**

The main objective behind the project is to provide a robust and secure solution for file sharing and management.

**Enhancing Network Security:** By utilizing machine learning algorithms for network traffic classification and blockchain technology for encryption key management, the project aims to enhance network security by detecting and mitigating potential threats such as denial of service attacks and unauthorized access attempts. The increasing reliance on digital file management systems for storing and sharing sensitive information has raised concerns about data security and privacy. Current systems lack robust security measures, leaving users vulnerable to cyber threats such as data breaches and unauthorized access.

**Efficient File Management:** The project seeks to provide a secure and efficient platform for file management by leveraging Flask for the web framework and various libraries for cryptography and machine learning. This includes features such as secure file storage, access control mechanisms, and real-time monitoring of network traffic.Developing a secure file management system aims to provide users with a reliable platform for managing files securely. By incorporating advanced security technologies like encryption, access control, anomaly detection, the system will ensure the confidentiality, integrity and availability of data.

**Secure Encryption Key Management:** By integrating blockchain technology for securely storing and managing encryption keys, the project aims to enhance data security and confidentiality, ensuring that sensitive information remains protected from unauthorized access or tampering.

**Enhancing User trust**: The project aims to provide a user-friendly interface for interacting with the system, allowing users to easily manage files, access control settings, and monitor network traffic in a seamless and intuitive manner. The project implements best practices in data security. This will help to mitigate the risks associated with cyber threats, ultimately improving data security and privacy for individuals and organizations

* 1. **Purpose and Scope**

**1.4.1 Purpose**

Thepurpose of the system is to provide a secure and efficient platform within an organization to store, manage, and share files. The system aims to address the increasing concerns regarding data security and privacy by implementing robust encryption algorithms, access controls and authentication mechanisms. The project ensures the confidentiality, integrity, and availability of stored data, it seeks to provide users with a reliable solution for protecting their sensitive information.

**1.4.2 Scope**

The **s**cope of the project includes the development of a comprehensive platform that offers secure storage, management, and sharing of files within an organization. The system includes features such as user authentication to enhance usability and data protection. The system focuses on providing robust security features to ensure the confidentiality, integrity, and availability of stored data.

**CHAPTER 2**

**LITERATURE SURVEY OF THE PROPOSED SYSTEM**

This chapter facilitated with the various research papers related to the detecting and preventing privilege escalation attacks.

**L.Liu, O. de Vel, Q.-L.Han, J.Zhang and Y.Xiang [15] have proposed a paper on**

**“Detecting and Preventing cyber insider threats: A Survey”.**

The above paper provides a comprehensive survey of techniques and approaches for detecting and preventing cyber insider threats. Insider threats refer to attacks or breaches that occur due to individuals within an organization having unauthorized access to resources or information. The paper also provides the motivations, common attack vectors, and the impact on organizations due to these insider threats. It also discusses different types of insider threats, such as malicious insiders and unintentional insiders. The paper discusses the need for comprehensive monitoring and balance between privacy and security.

**M.Abdelsalam, R. Krishnan, Y. Huang, and R.Sandhu [24] proposed a paper on “Malware detection in cloud infrastructures using Convolutional neural networks”.**

The above paper focuses on convolutional neural networks for detecting malware in cloud infrastructures. Cloud infrastructures are susceptible to malware attacks, which can lead to data breaches, service disruptions, and other security issues. The paper also says that traditional techniques struggle with evolving nature of malware, making it challenging to detect and mitigate threats effectively. The paper also discusses the use of CNNs used in image processing, for analyzing malware samples. The paper describes how CNNs are trained on a dataset of malware samples to distinguish between malicious and normal files.

**G. Ravikumar and M.Govindarasu [16] proposed a paper on “Anomaly detection and mitigation for wide-area damping control using machine learning”.**

The above paper focuses on application of machine learning techniques for detecting and mitigating anomalies in wide area damping control systems. Wide-area damping control is an important aspect of power system stability, aimed at dampening oscillations that can occur in large interconnected power grids. The paper discusses the need for real time monitoring and control of oscillations across a wide geographic area. The paper highlights the limitations of traditional control approaches and the potential benefits of using machine learning for anomaly detection and mitigation. This system includes Supervised learning algorithms, such as Support Vector Machines or neural networks, trained on labeled data to identify normal and abnormal behavior.

**H.Touqueer, S.Zaman, R.Amin, M.Hussain, F.Al-Turjman, and M.Bilal [6] have proposed a paper on “Smart home security: Challenges, issues and solutions at different IoT layers**”**.**

The above paper provides an in depth analysis of security challenges, issues, and solutions related to smart home environments. Smart homes are characterized by the integration of IoT devices for automation and remote monitoring, which introduces security concerns. It also discusses the various layers in smart homes. It identifies security challenges and vulnerabilities at each layer. The paper discusses the approaches to enhance security in smart homes. The paper also covers the emerging technologies and standards for smart home security, such as block chain for secure data sharing and access control. The paper highlighted the importance of user awareness and education in ensuring smart home security.

**N. T. Van and T. N. Thinh, [20] ‘‘An anomaly-based network intrusion detection system using deep learning’’.**

The primary focus of the paper was to develop an intrusion detection system (IDS) that could effectively detect anomalous behavior in network traffic, which could indicate a potential intrusion or attack. Traditional IDSs often rely on signature-based detection methods, which can be limited in their ability to detect new or previously unseen threats. Anomaly-based detection, on the other hand, aims to detect deviations from normal behavior, which can be indicative of an intrusion.The authors proposed using deep learning, a subset of machine learning, to build a more robust and accurate anomaly-based IDS. Deep learning models, such as deep neural networks, are capable of learning complex patterns and relationships in data, making them well-suited for detecting subtle anomalies in network traffic.The research likely involved collecting and preprocessing network traffic data to train the deep learning model. The model was then trained to recognize patterns of normal network behavior. During the testing phase, the model was exposed to new network traffic data, and any deviations from the learned patterns were flagged as potential anomalies. The paper likely discussed the performance of the proposed IDS compared to traditional signature-based IDSs. It may have included metrics such as detection accuracy, false positive rate, and detection speed to evaluate the effectiveness of the deep learning approach.

**CHAPTER 3**

**SYSTEM ANALYSIS**

This chapter presented various aspects of the system such as the existing system, proposed system, requirement analysis, software requirements, hardware requirements.

**3.1 Existing System**

**Signature-Based Detection:** This method relies on predefined signatures or patterns of known attacks to identify insider threats. Signatures are based on characteristics such as specific sequences of commands or access patterns that indicate malicious behavior. The main struggle with the signature-based detection is it may struggle with detecting new or previously unseen attacks.

**Anomaly Detection:** Establishing a baseline of normal behavior and flagging any deviations as potential threats. This method detects unusual patterns or anomalies that may indicate insider threats. The anomaly detection systems may generate false positives if the baseline is not accurately defined or if legitimate changes in behavior occur.

**Log Analysis and Monitoring:** Some systems employ log analysis and monitoring tools to detect suspicious activities by analyzing system logs and network traffic for unusual patterns or anomalies. It requires careful monitoring and analysis of large volumes of data.

**User and Entity Behavior Analytics:** This system analyzes user and entity behavior to detect and respond to insider threats. By building profiles of normal behavior for users and entities, these systems can identify anomalies which may indicate malicious activity. It struggles to adapt to rapidly changing user behavior patterns, which can limit their effectiveness in detecting insider threats.

**3.2 Proposed System**

The proposed system is designed to keep the files safe and efficient way to access these files. The system provides robust encryption mechanism, which ensures that all files are stored on the system are encrypted using strong encryption algorithm. This encryption helps protect files from unauthorized access and ensures that sensitive information remains confidential.

The system implements strict access controls to ensure that only authorized users have access to files. This will help prevent data breaches and unauthorized access to sensitive information. User authentication will also be required to access the system.

The system maintains audit trails of all file access and modifications. This will allow owners to track and monitor user activity, ensuring that files are accessed and modified only by authorized users. The system will provide users with the ability to securely backup their files, preventing data loss in case of system failure.

Version control will also be supported, allowing users to track changes to files and revert to previous versions if necessary. This will help users manage their files more effectively and ensure that they always have access to the most up-to-date versions.

**3.3 Requirement Analysis**

The provided software and hardware requirements establish foundation for a versatile and inclusive system or software. It requires an operating system such as Windows for broad accessibility. The system requires a minimum of 4GB of RAM, for the tasks like encryption and decryption of files. It also requires 64GB of storage space to securely store the files. The system is optimized for modern processors like intel i3 or equivalent to efficiently handle encryption processes. The software stack includes Python, enabling compatibility with a wide range of libraries and tools for security and tools for security and file management. It also uses databases such as SQLite or MySQL for storing the metadata and access control information.

**3.4 Hardware Requirements**

RAM : Minimum 4GB

Storage : Minimum 64GB

Processor : Intel i3

**3.5 Software Requirements**

Operating System : Windows 7 or later

Tool : Python 3.6

Platform : Anaconda

Database : SQLite

Framework : Flask framework

**3.6 Justification of Technology**

This project uses a minimum of 4GB RAM which ensures that the system can handle the processing requirements of file encryption, decryption, and user authentication efficiently. It requires a minimum of 64GB storage to accommodate the storage of encrypted files and user data securely. The intel processor ensures the compatibility and optimal performance for handling the encryption and decryption processes

Python serves various purposes, including backend development, web development, data processing, machine learning, security enhancement, automation, and integration with external systems. In backend development, Python is used for tasks like file uploading, downloading, encryption, decryption, and access control.

Python frameworks like Flask and Django are employed for web development, providing tools for building secure web interfaces. For data processing, Python libraries such as NumPy, pandas, and OpenCV are utilized, enabling efficient handling and analysis of data related to file management.

Machine learning models are implemented using libraries like TensorFlow, Keras, and scikit-learn for tasks such as fraud detection and anomaly detection. Python is also instrumental in implementing security features such as encryption algorithms and authentication mechanisms. Additionally, Python scripts automate various tasks such as backups, synchronization, and access logging. The Anaconda platform provides a set of tools and libraries for data management and encryption, enhancing the system’s security and functionality.

SQLite is used as the database due to its lightweight nature and efficiency in storing file metadata and users information.

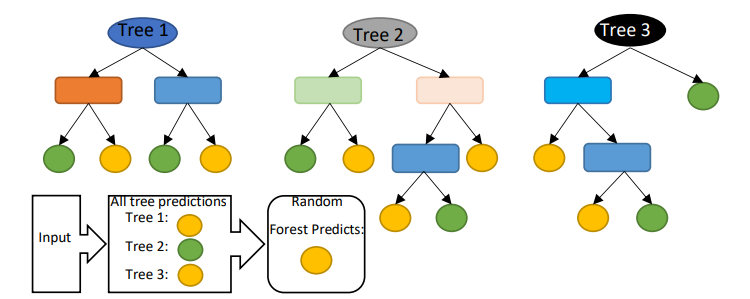
**CHAPTER 4**

**PROPOSED METHODOLOGY AND ALGORITHMS**

This chapter discusses the various algorithms that were used in the project. The use of each algorithm in this project is also discussed.

**4.1 Random Forest Algorithm**

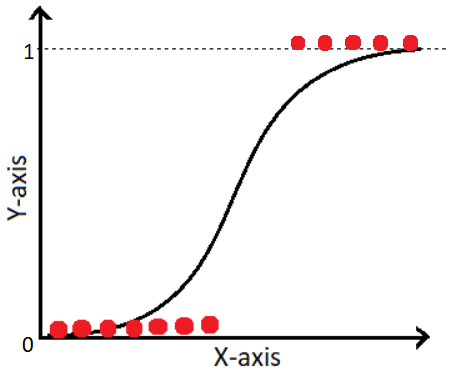
Random forest is a machine learning algorithm that leverages the strength of multiple decision trees to improve prediction accuracy. It works by constructing a forest of treesduring training, where each tree is built on a random subset of the training data and a random subset of features. This randomness helps to de correlate the trees, reducing the risk of overfitting and improving the model’s generalization ability. During prediction, the algorithm aggregates the predictions of individual trees, either through majority voting for classification tasks or averaging for regression tasks. In the project the Random Forest algorithm is utilized for the insider threat detection and classification, contributing to the system’s ability to identify various anomalous occurrences related to privilege escalation.



**Figure 4.1: Random Forest Classifier**

**4.2 Logistic Regression**

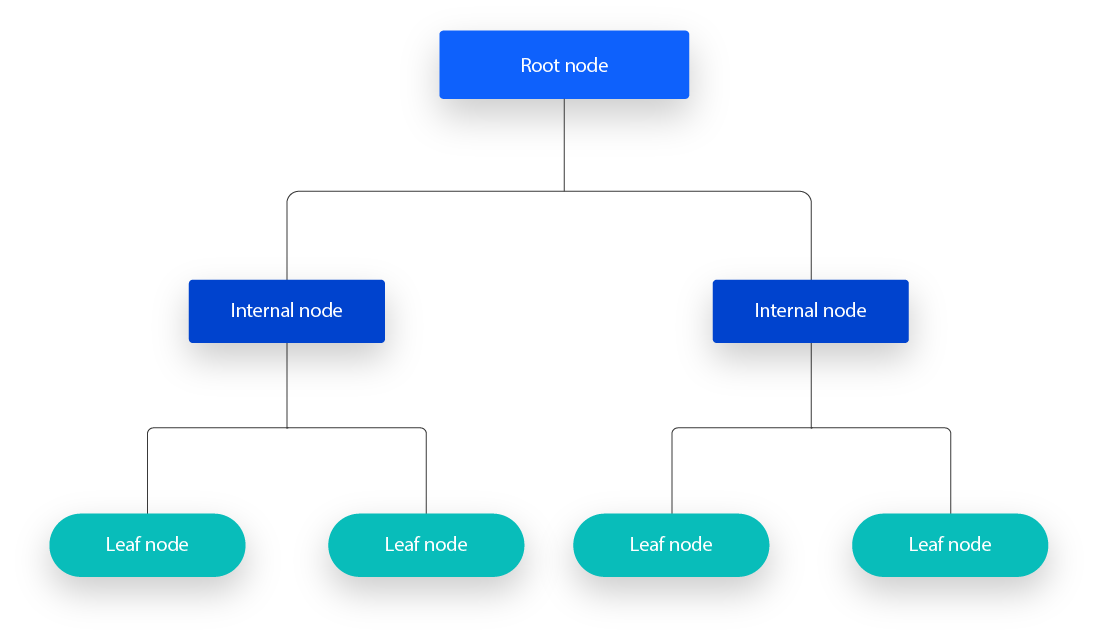
Logistic regression is a statistical model used for binary classification tasks, where the goal is to predict the probability that an instance belongs to a particular class. Logistic regression is a linear model that uses a logistic function to map the output of a linear combination of input features to a probability between 0 and 1. During training, the model adjusts its weights to minimize the logistic loss, which measures the difference between predicted probabilities and actual class labels. In this project logistic regression is used for the predictive maintenance, such as forecasting the downtime of the files based on real time predictions of potential failures.



**Figure 4.2: Logistic Regression**

**4.3 Decision Tree**

Decision tree is a machine learning model used for both classification and regression tasks. It is like tree structure where each internal node represents a test on an attribute and each leaf node represents a class label or a numerical value. The tree is built recursively by splitting the dataset into subsets based o the attribute that results I best split according to a certain criterion. In this project Decision Tree is used for the access control, where it can determine the level of access a user should have based on their attributes or role.



**Figure 4.3: Decision Tree**

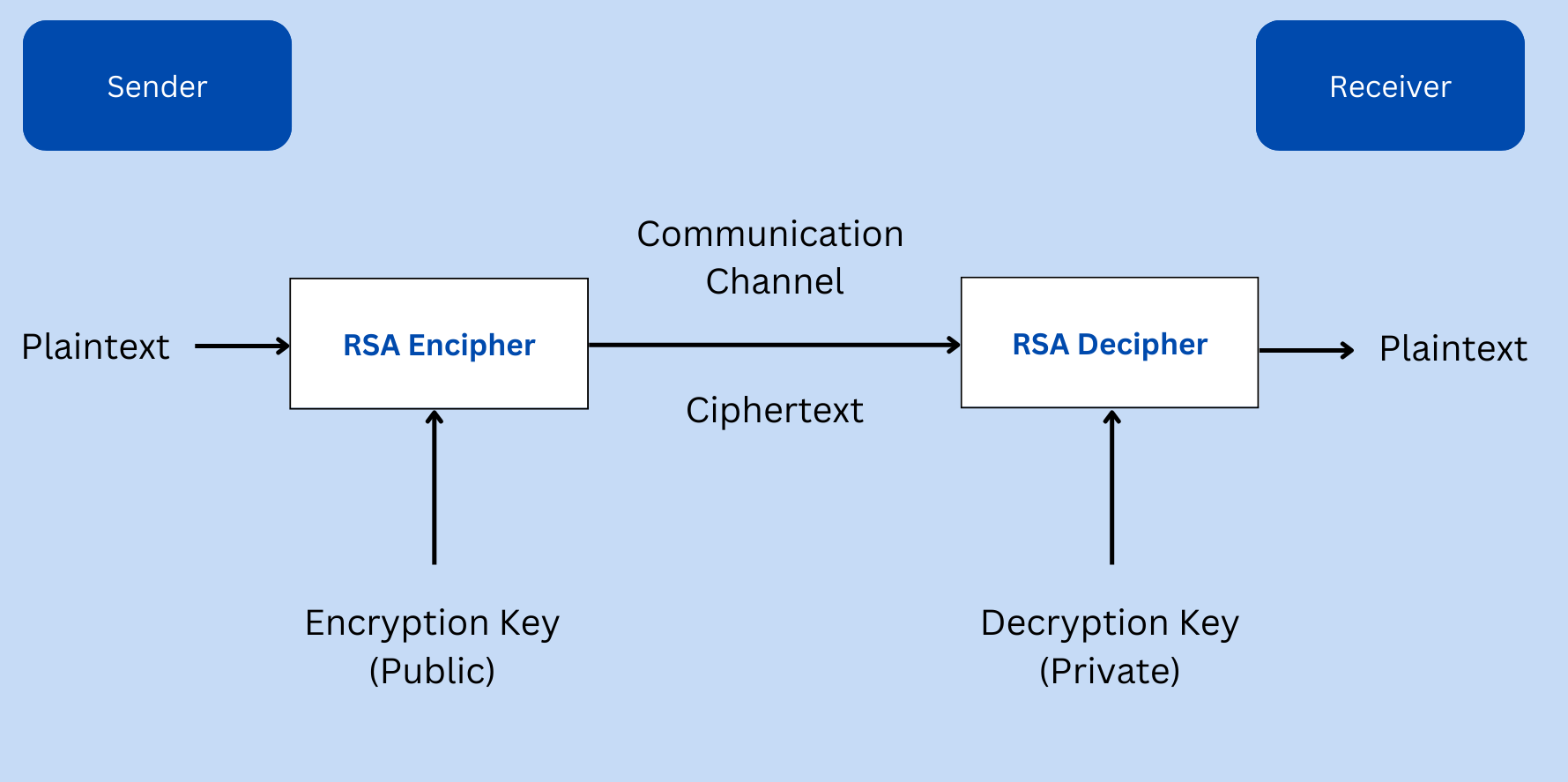
**4.4 Support Vector Machine**

Support Vector Machines are supervised learning models used for classification and regression tasks. The primary goal of SVM is to find the hyperplane that best separates different classes in the feature space while maximizing the margin, which is the distance between the hyperplane and the nearest point from each class, also known as support vectors. In this project SVM is used for anomaly detection in file access patterns. By analyzing the access patterns of users to files, the algorithm can identify unusual or suspicious access patterns that may indicate a potential security breach or insider threat.

 **Figure 4.4: Support Vector Machine**

**4.5 RSA Algorithm**

The Rivest-Shamir-Adleman algorithm is an asymmetric cryptographic algorithm that is used for secure data transmission. It involves the use of a public key for encryption and a private key for decryption. The security of RSA relies on the practical difficulty of factoring the product of two large prime numbers. The public key is made visible to anyone who wants to send encrypted data to the owner of the private key. The private key is kept secret and is used by the owner to decrypt the data encrypted with corresponding public key.



**Figure 4.5: RSA algorithm**

**CHAPTER 5**

**SYSTEM DESIGN**

This chapter discussed about the modules of the project, input design, output design, ER and UML Diagrams.

**5.1 Modules Division**

5.1.1Owner Module

The owner module helps the owners to facilitate various interactions of the owners. It begins with registration where new owners provide details such as username, password, email, contact number and password. These details are securely stored in the database. Owners can login with their username and password which are provided during the registration. These details are verified by the system from the database. After successful login of the owners, they can upload files, which are securely stored and are encrypted. Owners can also view their uploaded files and can also upload data to cloud storage, handling owner responses to user requests, and updates the database with owner responses.

5.1.2 User Module

The user module helps the users to facilitate various interactions of the user. The user module begins with the user registration where a new user provides information such as username, date of birth, email, city, contact number and password. The details are stored securely in a database. Users can login with their email and password which are provided during the registration. These details are verified by the system from the database. After successful login of the users they can view the files which were uploaded by the owners and send a request to the server to access the file. To access the file the user needs a private key which will be sent by the server to the users registered email. Then the user can view the data that is present in the file.

5.1.3 Server Module

The Server module helps the servers to facilitate various interactions of the server. The server module begins with the login process. The server is able to view the encrypted files which are uploaded by the owner of the system. The server is also responsible for the encryption of the data which is present in the files. The encryption of the file and file data is done by the RSA algorithm. The server provides a private key to user when user requests to view the file data. The private key will be sent to the users registered email.

**5.2 Input Design**

The input design for a Flask Secure File Management System is crucial for ensuring a user-friendly and secure experience. It involves creating interfaces that allow users to easily input data such as file names, descriptions, and access permissions. Validation is key to ensuring that the data entered by users is correct and that only authorized users can perform certain actions. Error handling mechanisms should be in place to manage any issues that arise during the input process, providing users with clear feedback on how to resolve them.

Simplicity is essential in the input design to help users navigate the system easily and reduce the risk of errors. Clear instructions should be provided to guide users through the process of uploading, managing, and sharing files. Feedback mechanisms, such as progress indicators or confirmation messages, can further enhance the user experience by providing reassurance that their actions have been successfully completed. By focusing on simplicity, security, and ease of use in the input design, the Flask Secure File Management System can offer users a seamless and efficient file management experience while ensuring the security and integrity of their data.

**5.3 Output design**

The output design for the system is crucial for presenting information to users in a clear and organized manner. It includes designing interfaces that allow users to view their uploaded files, access file details, and manage file permissions effectively. The system provides users with easy navigation options, allowing them to browse through their files and quickly locate the desired files.

The output design provides users with the ability to view detailed file properties, such as file name, size, upload date, and permissions. This information is presented in a user-friendly format that is easy to understand. The output design includes features for users to perform actions on their files, such as downloading, deleting, or sharing files with others.

Notification features are integrated into the output design to keep users informed about the status of their actions. This includes displaying notifications for successful file uploads, downloads, or changes to file permissions. Error messages should also be displayed when errors occur, providing users with clear instructions on how to resolve them**.**

The output design ensures the interfaces are responsive and compatible with different devices

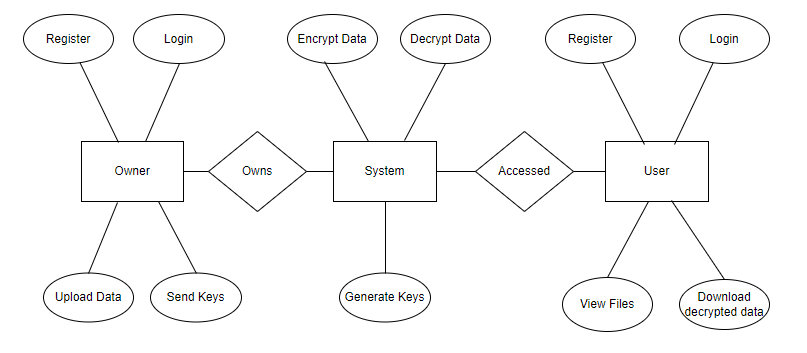
and screen sizes. This ensures a consistent user experience across various platforms, enhancing

usability and user satisfaction. Overall, the output design aims to provide users with intuitive

interfaces that facilitate efficient file management and enhance the overall user experience.

**5.4 ER Diagram**

The ER Diagram is the visual representation of the entities in a database and relationships between the entities. The entities are represented as rectangles, and relationships between the entities are represented by the rhombus and the attributes of the entities are represented as oval.

****

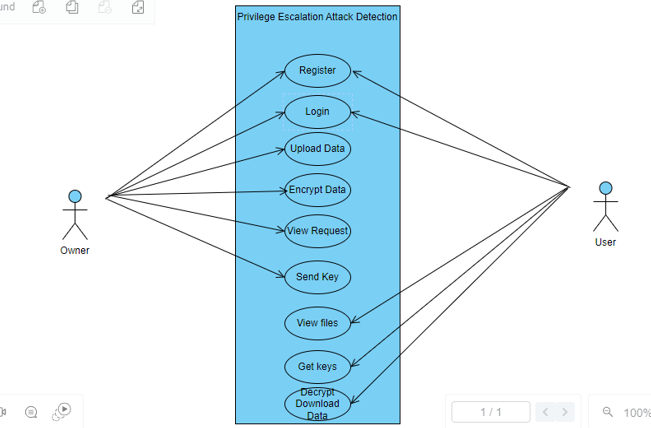
**Figure 5.1 ER Diagram**

* In the above ER diagram, the entities are Owner, User and System which are represented in rectangles.
* The owner can register, login, upload data, and send the keys to the system so these are the attributes of the owner which are represented in ovals.
* The attributes of the user are register, login view files, download the data.
* The relationship between the owner and the system is represented in a rhombus.

**5.5 UML Diagrams**

**Use case Diagram**

A Use case diagram is a visual representation of the interactions between the actors and a system, depicting the various ways in which the actors interact with the system. It contains actors, use cases, Relationships and System boundary.

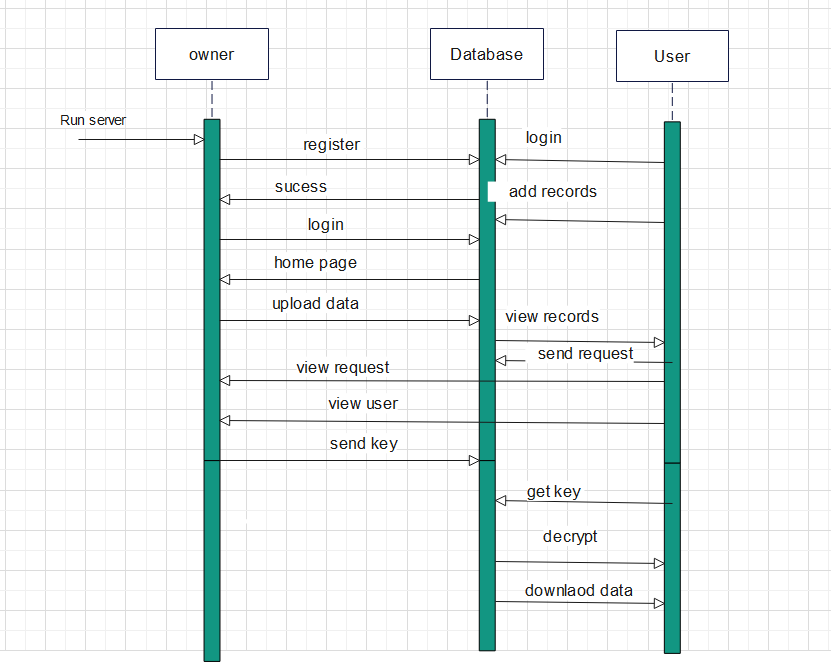
****

**Figure 5.2: Use Case Diagram**

In the above Use Case diagram, the actors are owners and the users. The use cases are register, login, upload data, encrypt data, view request, send key, view files, get key, and decrypt download data. The relationship between actors and the use cases represented as an arrow. The system boundary is the privilege escalation attack detection.

**Sequence Diagram**

A Sequence diagram is a type of interaction diagram that shows how objects interact in a particular scenario of a system operation. It depicts the sequence of messages exchanged between objects, along with lifelines of the objects involved in the interaction. It contains objects, messages, activation boxes, return messages.

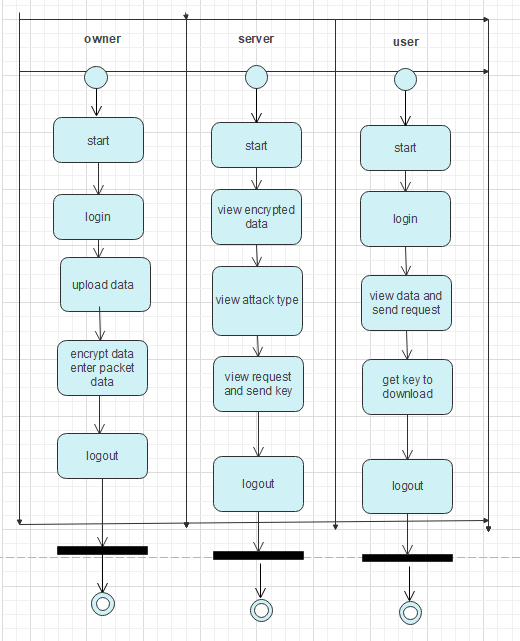


**Figure 5.3: Sequence Diagram**

The above sequence diagram the objects are owner, user, and database which are represented by rectangle. The messages are represented by an arrow between the two entities from sender to receiver. The activation box is represented by a vertical arrow which represents the lifeline of an object. The return messages are responses from the receiver object to the sender object.

**Activity** **Diagram**

An activity diagram is a visual representation within the Unified Modeling Language (UML) that depicts the workflow or sequence of activities in a system, process, or business operation. It employs various symbols such as rounded rectangles to denote activities, arrows to indicate the flow of control between activities, and diamond shapes for decision points where alternate paths may be taken based on conditions.



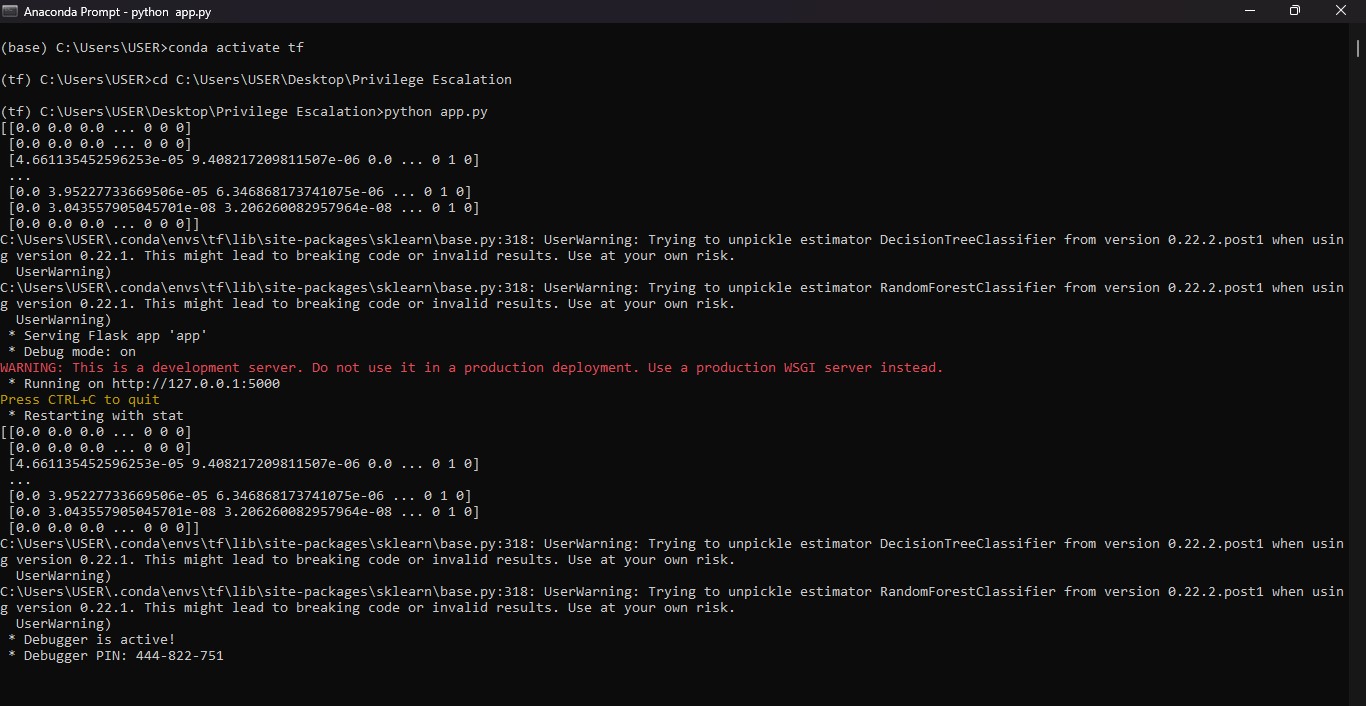
**Figure 5.4 Activity Diagram**

The above diagram represents the activity diagram, which starts with owner’s login and the sequence of activities performed by the owner. Then sequence of activities performed by the server. Finally sequence of activities performed by the user.

**CHAPTER 6**

**RESULTS AND DISCUSSIONS**

This chapter presents the outputs of the project and discuss about each output screen to understand it effectively.

****

**Figure 6.1: Execution of the Project in anaconda prompt**

The above output screen shows the execution of the project in the anaconda prompt. In this screen there will be an URL which should be copied.

****

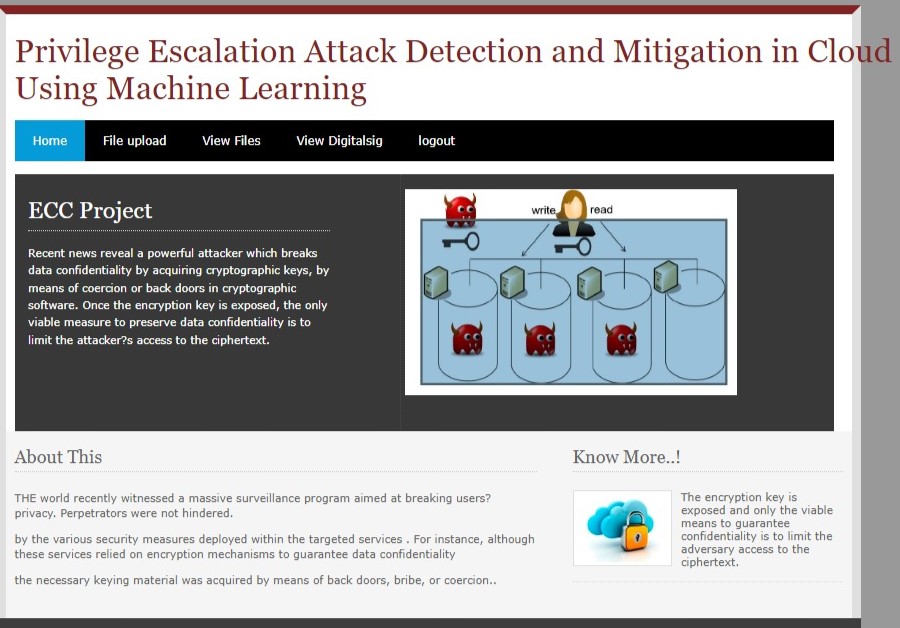
**Figure 6.2: Home page**

The above screen represents the home page of the project. It consists of information about the privilege escalation attacks.

****

**Figure 6.3: Owner Login Page**

The above screen represents the owner’s login page where each owner can login with their registered username and password.



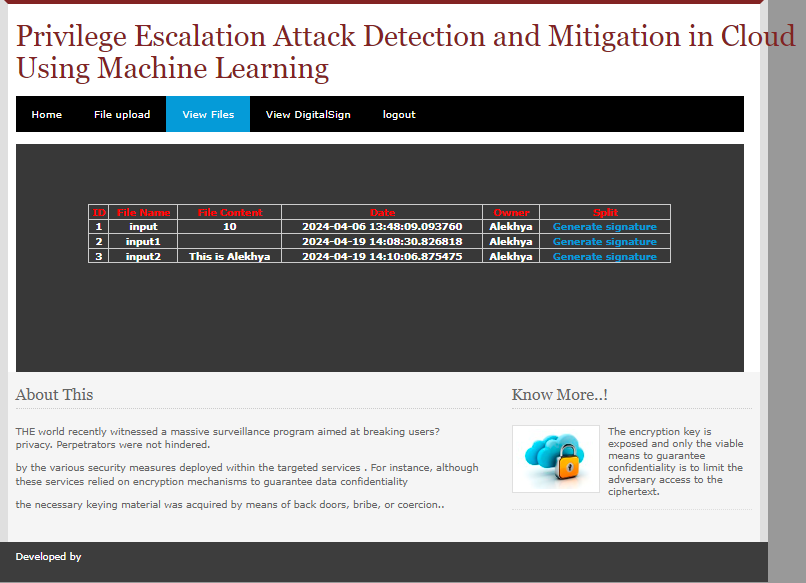
**Figure 6.4: Home page for owner**

The above screen represents the home page for the owner.



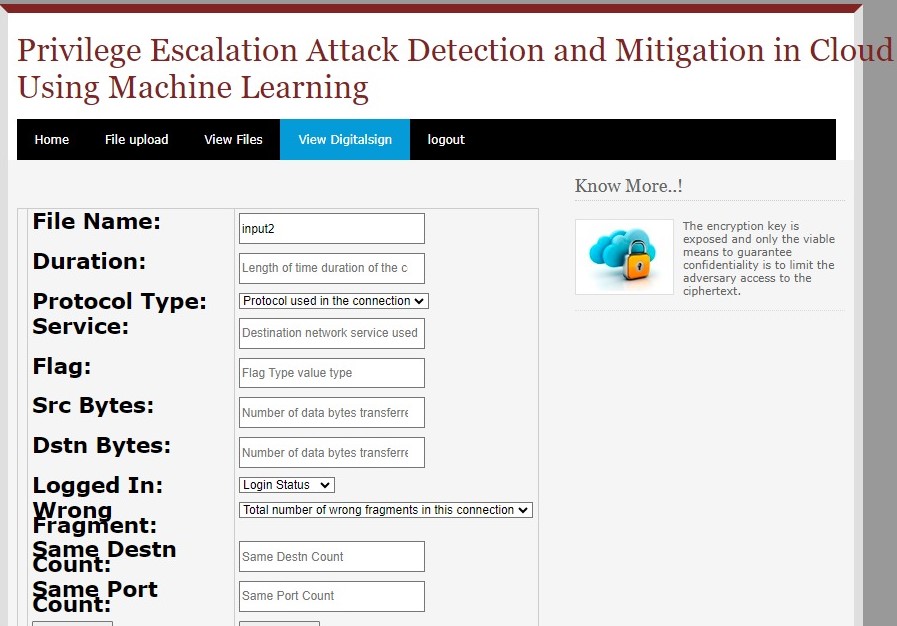
**Figure 6.5: File upload page**

The above screen represents the file upload page where an owner can upload the files of an organization with a file name.



**Figure 6.6: View files page**

The above screen shows the file which were uploaded by the owner with file name, file content, date and time, owner name and a link for the generation of the digital signature



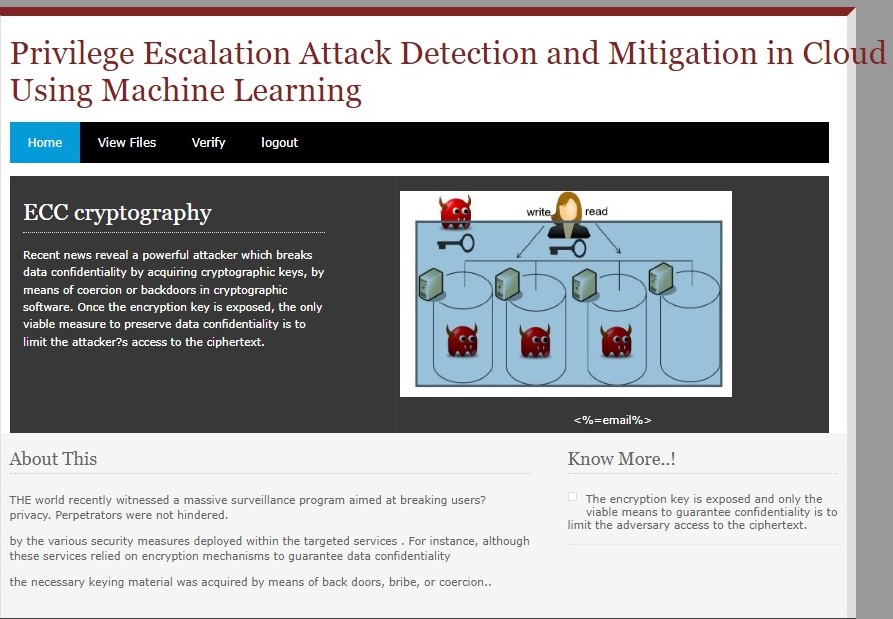
**Figure 6.7: View Digital Signature with status**

The above screen represents the digital signature to know whether the file uploaded by the owner is safe or not according to the values provided.



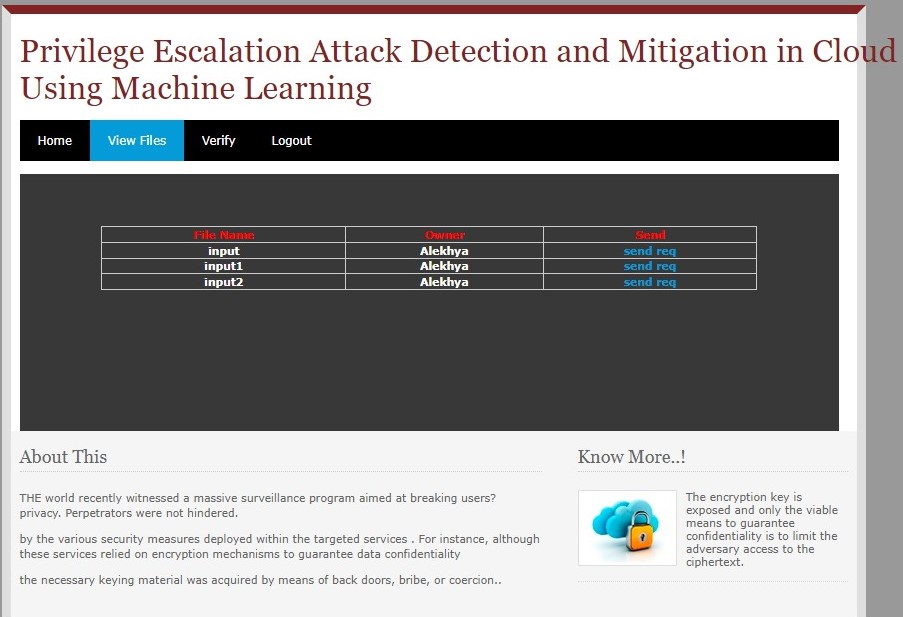
**Figure 6.8: User login page**

The above screen represents the user’s login page where user can login with an email and password.



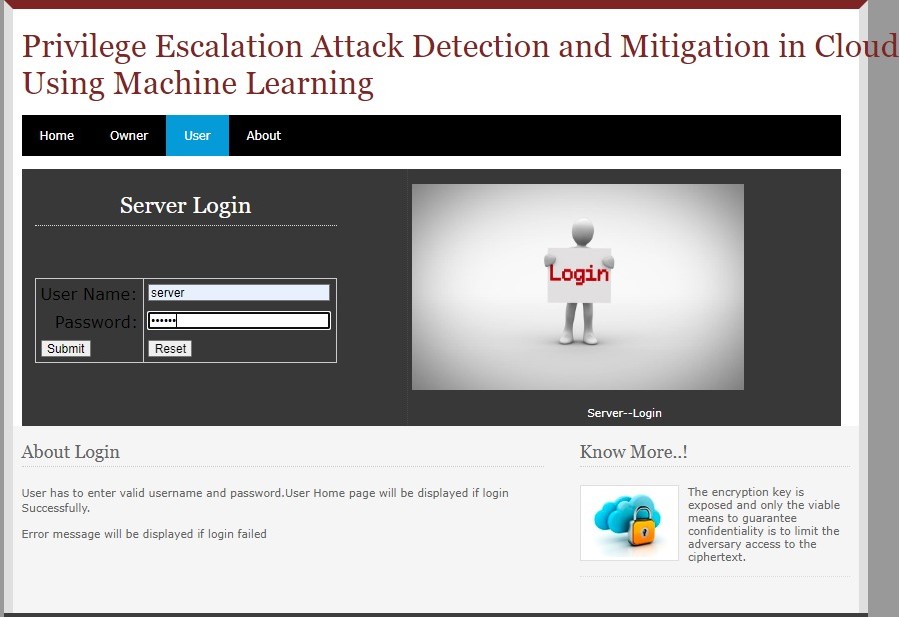
**Figure 6.9: Home page for user**

The above screen represents the home page for the user.



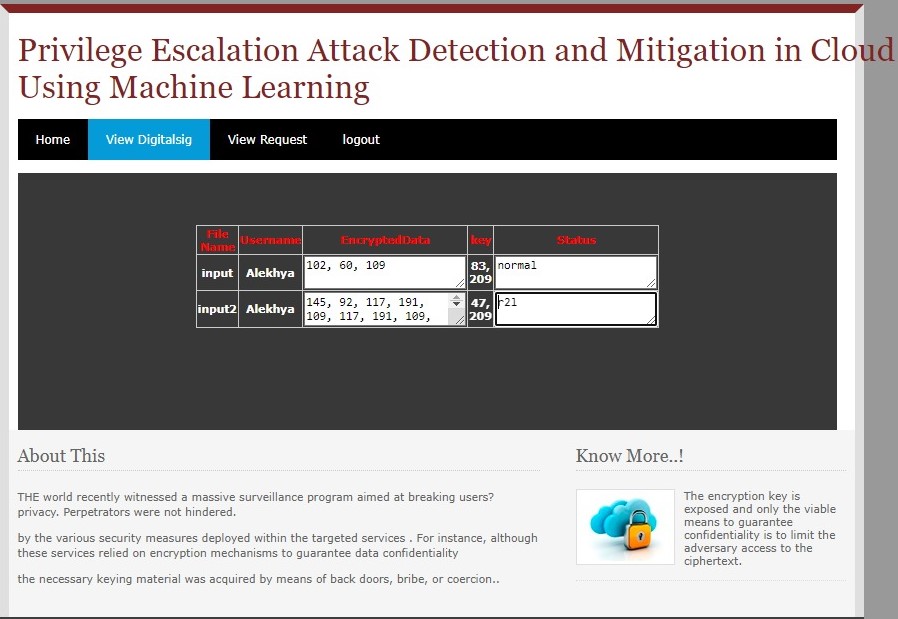
**Figure 6.10: View files for the user**

The above screen shows the various files to a user which were uploaded by the owner it contains the file name, owner name, and a requesting link to view the file.



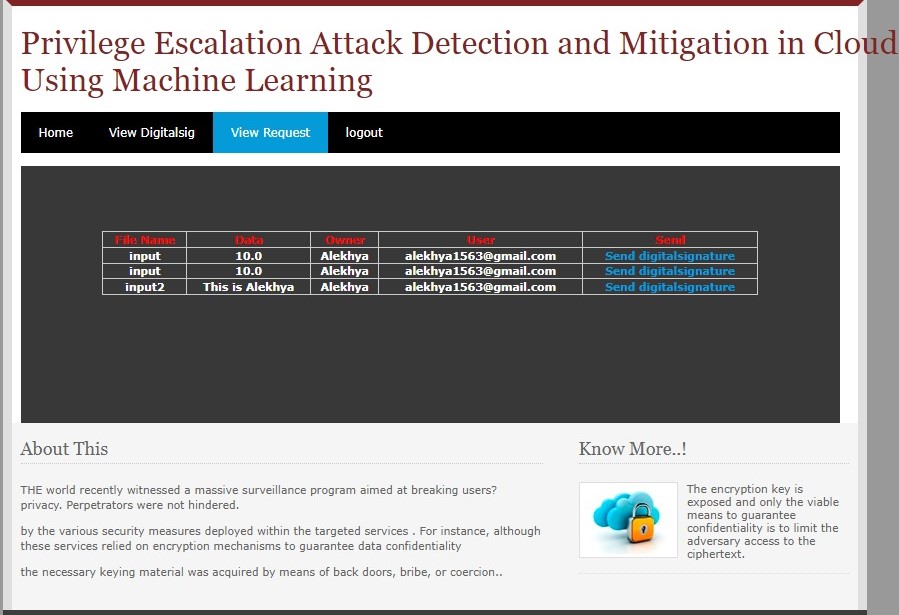
**Figure 6.11: Server Login page**

The above screen represents the server’s login page where each server can login with an username and password.



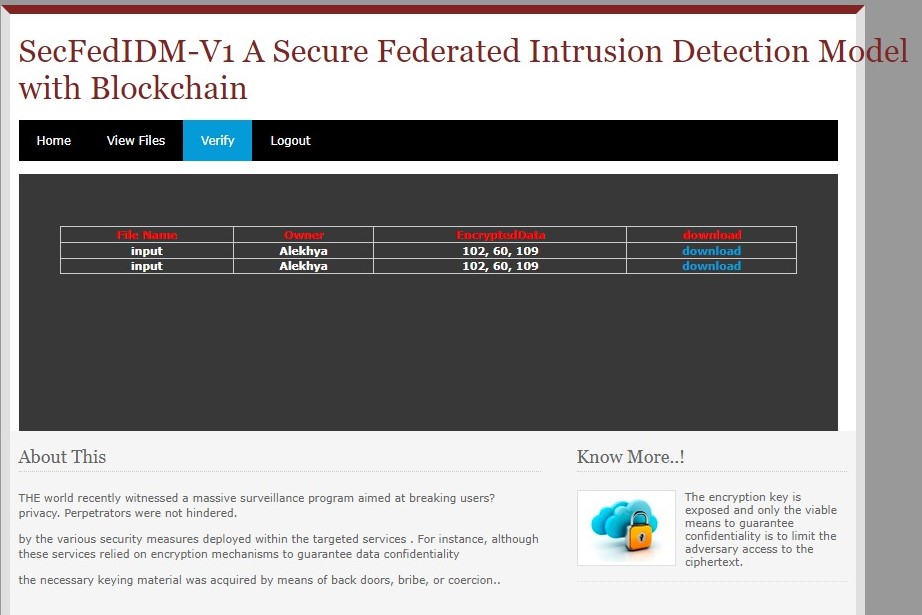
**Figure 6.12: View Digital signature page**

The above screen represents the status of each file whether it is a normal file or a malicious file.



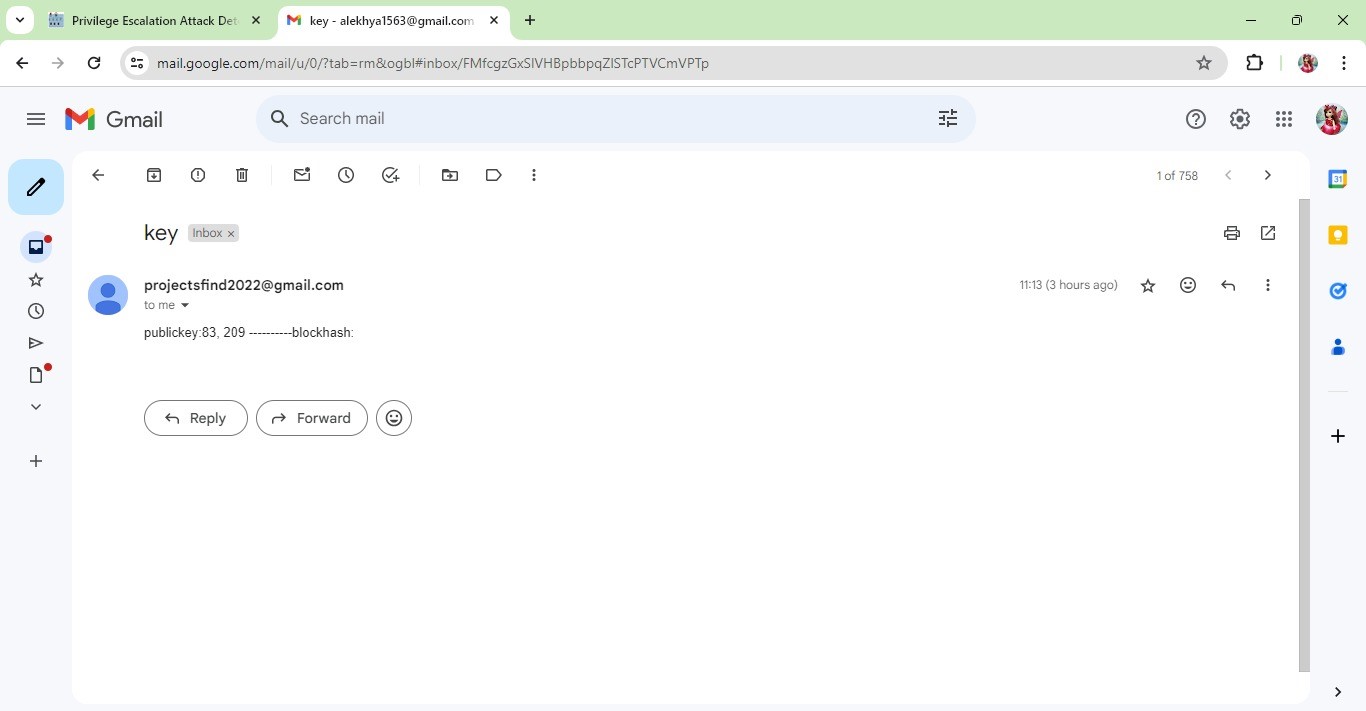
**Figure 6.13: View requests page**

The above screen shows the requests from users to view the file. It consists of the file name, the information in the file, Owner name, user’s mail id, and a link to approve the file viewing.



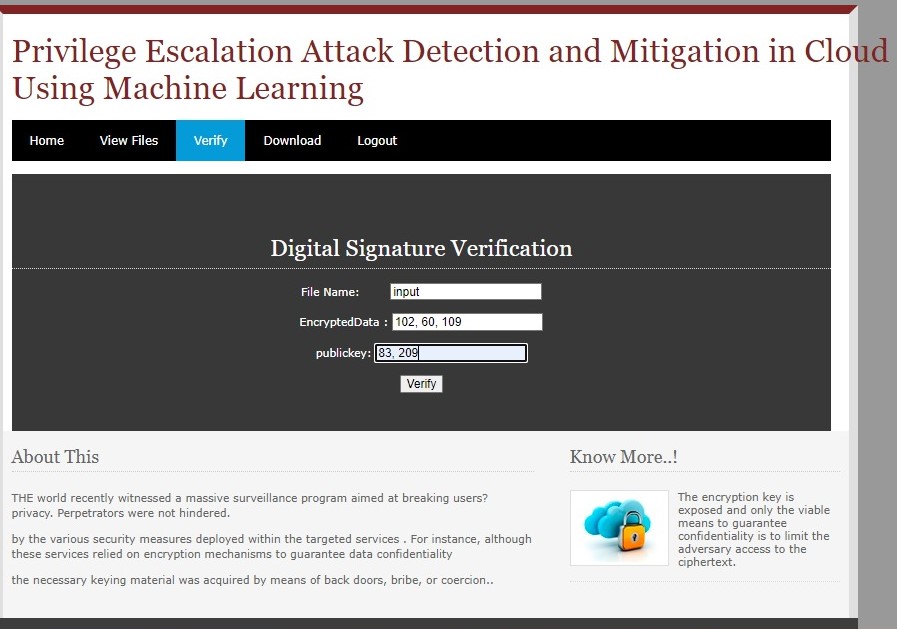
**Figure 6.14: Verifying the files page**

The above screen represents the verification of the file. It contains the file name, owner name, encrypted data and a link to verify the file.



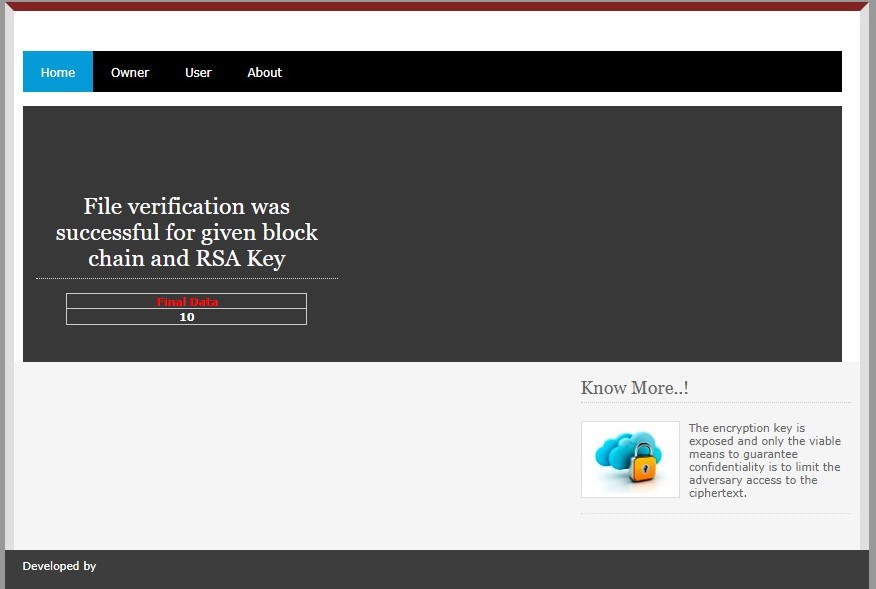
**Figure 6.15: Key generated mail**

The above screen gives the key that is generated for a file to the user’s mail.



**Figure 6.16: Verifying the Public Key**

The above screen represents the verification of the key which was sent to the user’s mail id.



**Figure 6.17: Final Output**

The above screen gives the final output to the user who wanted to access the file.

**CHAPTER 7**

**CONCLUSION AND FUTURE WORK**

This chapter provides the conclusion of the project and gives the suggestions of the extensions of the project that is the future work of the project.

**Conclusion**

In today's digital world, ensuring the security of cloud-based systems is more important than ever. With businesses and individuals increasingly relying on cloud services for storing, sharing, and managing files, the need for strong security measures is paramount. To address these concerns, a cutting-edge project has been developed to offer a secure platform that not only protects data but also provides a user-friendly experience. This project uses advanced encryption techniques to secure data stored in the cloud. Before files are uploaded, they are encrypted using state-of-the-art algorithms. This ensures that even if the cloud service provider's servers are compromised, the data remains unreadable to unauthorized users. Additionally, the project includes robust authentication mechanisms to verify users' identities, preventing unauthorized access to sensitive information.

In addition to its focus on security, the project places a high priority on user experience. The platform features an intuitive interface that makes it easy for users to upload, share, and manage their files. Clear instructions and user-friendly controls guide users through the process, making it simple for even non-technical users to securely store and share their data.This project represents a significant advancement in cloud security. The system is designed with a strong emphasis on data security and integrity. It employs advanced encryption algorithms to ensure that all files are protected during transmission and while at rest.

Key management is a crucial aspect, and it is handled using RSA encryption, a widely trusted public-key cryptosystem. RSA encryption ensures that even if data is intercepted, it remains unreadable without the corresponding private key, providing an additional layer of security against unauthorized access. The system integrates blockchain technology to further enhance security. The decentralized nature of blockchain means there is no single point of failure, making it highly resistant to tampering and fraud. Each transaction is recorded as a block linked to the previous one, creating a chain that is virtually impossible to alter without detection. Continuous monitoring and auditing are employed to detect and respond to security incidents in real-time, minimizing the impact of potential breaches. Regular security updates and patches are also applied to protect against newly discovered vulnerabilities.

**Future Work**

Implementing advanced encryption techniques, integrating multi-factor authentication, improving the user interface, expanding file type support, enhancing scalability, integrating with cloud storage services, and incorporating machine learning algorithms for security are all valuable directions for future development. These enhancements will not only improve the system's security but also enhance user experience and functionality. Enhancing logging and auditing capabilities could also be beneficial for better traceability and compliance with data protection regulations. This could involve implementing logging mechanisms to track user actions and file access, as well as integrating with auditing tools to analyze logs and identify potential security incidents.

Implementing data loss prevention (DLP) measures could also be beneficial to prevent unauthorized access, use, or transmission of sensitive data. This could involve implementing encryption, data masking, and access controls to protect sensitive information.

**REFERENCES**

1. U. A. Butt, R. Amin, H. Aldabbas, S. Mohan, B. Alouffi, and A. Ahmadian, ‘‘Cloud-based email phishing attack using machine and deep learning algorithm,’’ Complex Intell. Syst., pp. 1–28, Jun. 2022.
2. D. C. Le and A. N. Zincir-Heywood, ‘‘Machine learning based insider threat modelling and detection,’’ in Proc. IFIP/IEEE Symp. Integr. Netw. Service Manag. (IM), Apr. 2019, pp. 1–6.
3. P. Oberoi, ‘‘Survey of various security attacks in clouds based environments,’’ Int. J. Adv. Res. Comput. Sci., vol. 8, no. 9, pp. 405–410, Sep. 2017.
4. A.Ajmal, S. Ibrar, and R. Amin, ‘‘Cloud computing platform: Performance analysis of prominent cryptographic algorithms,’’ Concurrency Comput., Pract. Exper., vol. 34, no. 15, p. e6938, Jul. 2022.
5. U. A. Butt, R. Amin, M. Mehmood, H. Aldabbas, M. T. Alharbi, and N. Albaqami, ‘‘Cloud security threats and solutions: A survey,’’ Wireless Pers. Commun., vol. 128, no. 1, pp. 387–413, Jan. 2023.
6. H. Touqeer, S. Zaman, R. Amin, M. Hussain, F. Al-Turjman, and M. Bilal, ‘‘Smart home security: Challenges, issues and solutions at different IoT layers,’’ J. Supercomput., vol. 77, no. 12, pp. 14053–14089, Dec. 2021.
7. S. Zou, H. Sun, G. Xu, and R. Quan, ‘‘Ensemble strategy for insider threat detection from user activity logs,’’ Comput., Mater. Continua, vol. 65, no. 2, pp. 1321– 1334, 2021
8. G. Apruzzese, M. Colajanni, L. Ferretti, A. Guido, and M. Marchetti, ‘‘On the effectiveness of machine and deep learning for cyber security,’’ in Proc. 10th Int. Conf. Cyber Conflict (CyCon), May 2018, pp. 371–390.
9. D. C. Le, N. Zincir-Heywood, and M. I. Heywood, ‘‘Analyzing data granularity levels for insider threat detection using machine learning,’’ IEEE Trans. Netw. Service Manag., vol. 17, no. 1, pp. 30–44, Mar. 2020.
10. F. Janjua, A. Masood, H. Abbas, and I. Rashid, ‘‘Handling insider threat through supervised machine learning techniques,’’ Proc. Comput. Sci., vol. 177, pp. 64–71, Jan. 2020.
11. R. Kumar, K. Sethi, N. Prajapati, R. R. Rout, and P. Bera, ‘‘Machine learning based malware detection in cloud environment using clustering approach,’’ in Proc. 11th Int. Conf. Comput., Commun. Netw. Technol. (ICCCNT), Jul. 2020, pp. 1–7.
12. D. Tripathy, R. Gohil, and T. Halabi, ‘‘Detecting SQL injection attacks in cloud SaaS using machine learning,’’ in Proc. IEEE 6th Int. Conf. Big Data Secur. Cloud (BigDataSecurity), Int. Conf. High Perform. Smart Comput., (HPSC), IEEE Int. Conf. Intell. Data Secur. (IDS), May 2020, pp. 145–150.
13. X. Sun, Y. Wang, and Z. Shi, ‘‘Insider threat detection using an unsupervised learning method: COPOD,’’ in Proc. Int. Conf. Commun., Inf. Syst. Comput. Eng. (CISCE), May 2021, pp. 749–754.
14. J. Kim, M. Park, H. Kim, S. Cho, and P. Kang, ‘‘Insider threat detection based on user behavior modeling and anomaly detection algorithms,’’ Appl. Sci., vol. 9, no. 19, p. 4018, Sep. 2019.
15. L. Liu, O. de Vel, Q.-L. Han, J. Zhang, and Y. Xiang, ‘‘Detecting and preventing cyber insider threats: A survey,’’ IEEE Commun. Surveys Tuts., vol. 20, no. 2, pp. 1397–1417, 2nd Quart., 2018.
16. P. Chattopadhyay, L. Wang, and Y.-P. Tan, ‘‘Scenario-based insider threat detection from cyber activities,’’ IEEE Trans. Computat. Social Syst., vol. 5, no. 3, pp. 660–675, Sep. 2018.
17. G. Ravikumar and M. Govindarasu, ‘‘Anomaly detection and mitigation for wide-area damping control using machine learning,’’ IEEE Trans. Smart Grid, early access, May 18, 2020, doi: 10.1109/TSG.2020.2995313.
18. D. S. Berman, A. L. Buczak, J. S. Chavis, and C. L. Corbett, ‘‘A survey of deep learning methods for cyber security,’’ Information, vol. 10, no. 4, p. 122, 2019.
19. M. I. Tariq, N. A. Memon, S. Ahmed, S. Tayyaba, M. T. Mushtaq, N. A. Mian, M. Imran, and M. W. Ashraf, ‘‘A review of deep learning security and privacy defensive techniques,’’ Mobile Inf. Syst., vol. 2020, pp. 1–18, Apr. 2020
20. N. T. Van and T. N. Thinh, ‘‘An anomaly-based network intrusion detection system using deep learning,’’ in Proc. Int. Conf. Syst. Sci. Eng. (ICSSE), 2017, pp. 210–214.
21. G. Pang, C. Shen, L. Cao, and A. V. D. Hengel, ‘‘Deep learning for anomaly detection: A review,’’ ACM Comput. Surv., vol. 54, no. 2, pp. 1–38, Mar. 2021.
22. Arora, A. Khanna, A. Rastogi, and A. Agarwal, ‘‘Cloud security ecosystem for data security and privacy,’’ in Proc. 7th Int. Conf. Cloud Comput., Data Sci. Eng., Jan. 2017, pp. 288–292.
23. L. Coppolino, S. D’Antonio, G. Mazzeo, and L. Romano, ‘‘Cloud security: Emerging threats and current solutions,’’ Comput. Electr. Eng., vol. 59, pp. 126–140, Apr. 2017.
24. M. Abdelsalam, R. Krishnan, Y. Huang, and R. Sandhu, ‘‘Malware detection in cloud infrastructures using convolutional neural networks,’’ in Proc. IEEE 11th Int. Conf. Cloud Comput. (CLOUD), Jul. 2018, pp. 162–169.
25. F. Jaafar, G. Nicolescu, and C. Richard, ‘‘A systematic approach for privilege escalation prevention,’’ in Proc. IEEE Int. Conf. Softw. Quality, Rel. Secur. Companion (QRS) Aug. 2016, pp. 101-108.