IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 3, June 2025
Impact Factor: 7.67

Enhanced Security Framework for Multi-Layered Wireless Communication Systems

Aradhye Vaishnavi, Hinge Aditi, Ingale Pratham, Khutale Vedang
Prof. (Mrs.) R. A. Waghole
Department of Computer Engineering
Smt. Kashibai Navale College of Engineering, Pune, India

Abstract: *This project aims to develop a lightweight and efficient Intrusion Detection System (IDS) tailored for secure communication environments, implemented entirely in Java using Eclipse IDE. The system integrates a rule-based detection mechanism that identifies known malicious input patterns such as SQL injections, system command exploits, and scripting attacks by comparing user input against a predefined list of suspicious keywords. It operates with a full-stack architecture that includes JSP, HTML, CSS, and JavaScript for the frontend, Java for the core detection logic and encryption processes, and MySQL (configured through phpMyAdmin) for backend storage and log management. A key feature of the system is its simulated honeypot mechanism, embedded within the login interface to capture and record unauthorized access attempts such as brute-force attacks or invalid username entries. These attempts are logged separately for administrator analysis without disrupting legitimate users. The system also incorporates AES (Advanced Encryption Standard) for secure file transmission, allowing encrypted input from users to be scanned for threats and safely stored in the database. Unlike machine learning-based IDS models that require large datasets and complex training, this system provides a deterministic, transparent, and fast response by leveraging simple yet effective rule-based logic. It ensures real-time feedback through a responsive GUI built with Java Swing and web technologies, while maintaining a clear separation of valid, malicious, and honeypot log entries in the backend. Designed for academic use, small-scale enterprise security, or as a prototype for future IDS enhancements, this solution balances simplicity with practical security functionality, offering high maintainability, auditability, and modular design suited for standalone or LAN-based deployment*

Keywords: Intrusion Detection System

I. INTRODUCTION

In the modern digital age, computer networks form the backbone of communication and information exchange across all domains—whether academic, industrial, governmental, or commercial. From cloud-based services to personal messaging platforms, the reliance on interconnected systems has increased exponentially, bringing with it a significant concern: the security of data and network integrity.

Network security is no longer optional—it is a mandatory component of any system architecture. As cyberattacks evolve in complexity and frequency, systems must be equipped not only to prevent unauthorized access but also to detect intrusions in real-time and respond appropriately.

To secure the flow of information over a network, three core principles of information security must be preserved:

- Confidentiality – Data should be accessible only to those who are authorized to view or process it.
- Integrity – The data must be accurate and remain unchanged except by authorized users or systems.
- Availability – Systems and data should be accessible and operational whenever needed.

An Intrusion Detection System (IDS) is a security mechanism that continuously monitors network traffic and system behavior to detect potential threats, such as unauthorized access, exploitation attempts, or malicious code execution. IDSs play a critical role by alerting administrators when suspicious behavior is detected, allowing quick action to be taken before serious damage occurs.

DOI: 10.48175/IJARSCT-27598

In this project, we propose a custom-built IDS using Java on the Eclipse IDE, integrated with MySQL through MyPHP for backend operations. The IDS comprises:

- A Graphical User Interface (GUI) built in Java that includes user roles such as admin and user, with secure login functionality.
- A keyword-based detection mechanism that scans input messages for known malicious patterns or keywords, functioning as the rule-based detection layer.
- A Honeypot system that mimics a vulnerable part of the system to attract potential attackers. All interactions with the honeypot are logged, allowing for tracking of intrusion attempts and behavior analysis.

This project serves as a practical implementation of basic cybersecurity principles through simple yet effective components, ensuring real-time threat monitoring with extensible architecture for future upgrades.

1.1 MOTIVATION

The increasing number of network-dependent applications in sectors like e-commerce, banking, healthcare, and education has made the security of transmitted information a high priority. Malicious actors exploit vulnerabilities through phishing, brute-force attacks, injection attacks, or social engineering, compromising critical systems and personal data.

Despite the availability of sophisticated enterprise-level IDS solutions, many are resource-intensive, difficult to understand, or costly for small-scale implementations. There is a need for an easily deployable, lightweight IDS that can be used for educational purposes, small organizations, or as a prototype for larger systems.

This project is motivated by the following objectives:

- To develop an accessible and extensible Intrusion Detection System that can detect unauthorized activity based on predefined malicious keywords.
- To create a modular Java application that allows users to log in via a user-friendly interface, enhancing usability for both administrators and general users.
- To log intrusion attempts through a honeypot mechanism, enabling analysis of attacker behavior in a controlled setting.
- To ensure key information security principles (Confidentiality, Integrity, and Availability) are maintained through authentication, message inspection, and secure data handling using MySQL and MyPHP.

By using open-source tools and intuitive logic, the system is designed to be educational, cost-effective, and a starting point for implementing more complex intrusion detection and prevention models in the future.

1.2 PROBLEM STATEMENT

Design and implement a user-friendly, rule-based Intrusion Detection System (IDS) using Java (Eclipse IDE), MySQL database (accessed via MyPHP), and a honeypot mechanism, with the following core functionalities:

- Provide a secure login system with different access roles (Admin and User), allowing only authenticated users to proceed.
- Accept a text-based message input through the GUI.
- Perform real-time analysis of the input message using a keyword-based detection engine to determine whether the message is malicious.
- Generate appropriate alerts or logs based on the detection result, notifying users or recording malicious attempts in a database.
- Employ a honeypot subsystem that simulates a vulnerable environment to trap and record suspicious access attempts, enhancing the monitoring capabilities of the system.
- Preserve the fundamental principles of information security:

o Confidentiality by restricting unauthorized access via secure login,
o Integrity by ensuring input messages are validated and verified,
o Availability by maintaining continuous system access and data flow logging.

DOI: 10.48175/IJARSCT-27598

The solution should be easily deployable and understandable, offering a scalable foundation for enhancements such as machine learning-based intrusion detection, dynamic keyword learning, or integration with real-time alert systems.

1.3 OBJECTIVES

The main objective of this project is to design and implement an intelligent Intrusion Detection System (IDS) that can monitor network-based communication and detect suspicious or malicious behavior using a rule-based approach with future extensibility.

This project aims to fulfill the following specific objectives:

1. Develop a Java-based IDS with GUI:

o Create an interactive and user-friendly interface using Java Swing/JavaFX on the Eclipse IDE.

o Provide secure login features with distinct access roles for Admin and User.

2. Implement a Keyword-Based Detection Mechanism:

o Scan incoming message inputs for predefined suspicious keywords or patterns.

o Classify inputs as benign or malicious based on rule-based matching.

3. Integrate a Honeypot System for Attack Simulation:

o Deploy a honeypot component that emulates vulnerabilities to attract attackers.

o Record details of intrusion attempts including timestamp, user ID, and message content.

4. Backend Data Management using MySQL &MyPHP:

o Store login credentials, intrusion logs, and message history securely in a MySQL database.

o Use MyPHP for interfacing between the front-end and the database.

5. Real-Time Logging and Alerting:

o Provide visual or text-based alerts when malicious input is detected.

o Maintain a persistent log of user actions and detected threats for post-event analysis.

6. Ensure Core Security Principles:

o Uphold Confidentiality via authentication mechanisms.

o Maintain Integrity by filtering and validating user inputs.

o Support Availability by designing a stable, responsive application.

7. Design the System with Scalability and Extensibility in Mind:

o Create modular architecture to support future integration of advanced detection techniques such as ML-based classifiers, dynamic keyword learning, and real-time network monitoring.

1.4 SCOPE

The scope of this project encompasses the design, development, and demonstration of a rule-based Intrusion Detection System (IDS) tailored for secure communication environments. The current implementation is focused on text-message-based threat detection, but its framework can be extended to handle broader threat scenarios.

The boundaries and capabilities of this project are outlined below:

- Included in Scope:
  1. Java-based graphical interface for secure login, message input, and output.
  2. Keyword-matching logic for static rule-based detection of malicious content.
  3. Simulation of attacker behavior using honeypot traps and fake vulnerabilities.
  4. Logging of malicious message attempts in a backend MySQL database.
  5. Modular design that enables future enhancements (e.g., ML integration).
  6. Simple text data handling to demonstrate IDS functionality.
- Not Included in Scope (Current Phase):
  1. No use of machine learning or AI-based threat prediction in the initial version.
  2. No real-time traffic sniffing or packet-level analysis.
  3. No encryption/decryption or tunneling mechanisms are currently implemented.
  4. No mobile platform deployment (limited to desktop environment using Java).

DOI: 10.48175/IJARSCT-27598

- Target Environment:

o Educational settings, academic demonstration, research projects.

o Prototype testing environments with simulated user interaction.

This scope defines a strong baseline for a lightweight IDS suitable for academic and prototype-level security testing, while also allowing room for evolution into more robust detection frameworks in the future.

II. LITERATURESURVEY

SATISH KUMAR, SUNANDA GUPTA [1]

Network threats and hazards are evolving at a high-speed rate in recent years. Many mechanisms (such as firewalls, anti- virus, anti-malware, and spam filters) are being used as security tools to protect networks. An intrusion detection system (IDS) is also an effective and powerful network security system to detect unauthorized and abnormal network traffic flow. This article presents a review of the research trends in network-based intrusion detection systems (NIDS), their approaches, and the most common datasets used to evaluate IDS Models. The analysis presented in this paper is based on the number of citations acquired by an article published, the total count of articles published related to intrusion detection in a year, and most cited research articles related to the intrusion detection system in journals and conferences separately. Based on the published articles in the intrusion detection field for the last 15 years, this article also discusses the state-of-the-arts of NIDS, commonly used NIDS, citation-based analysis of benchmark datasets, and NIDS techniques used for intrusion detection. A citation and publication-based comparative analysis to quantify the popularity of various approaches are also presented in this paper. The study in this article may be helpful to the novices and researchers interested in evaluating research trends in NIDS and their related applications

Lirim Ashiku1 Cihan Dagli [2] capability. The vulnerabilities deem cyber- security mechanisms essential to assume communication exchange. Secure communication requires security measures to combat the threats and needs advancements to security measures that counter evolving security threats. This paper proposes the use of deep learning architectures to develop an adaptive and resilient network intrusiondetection system (IDS) to detect and classify network attacks. The emphasis is how deep learning or deep neural networks (DNNs) can facilitate flexible IDS with learn- ing capability to detect recognized and new or zero-day network behavioral features, consequently ejecting the systems intruder and reducing the risk of compromise. To demonstrate the model's effectiveness, we used the UNSW-NB15 dataset, reflectingreal modern network communication behavior with synthetically

Anish Halimaa A,Dr. K.Sundarakantham[3]

In order to examine malicious activity that occurs in a network or a system, intrusion detection system is used. Intrusion Detection is software or a device that scans a system or a network for a distrustful activity. Due to the growing connectivity between computers, intrusion detection becomes vital to perform network security. Various machine learningtechniques and statistical methodologies have been used to build different types of Intrusion Detection Systems to protect the networks. Performance of an Intrusion Detection is mainly depends on accuracy. Accuracy for Intrusion detection must be enhanced to reduce false alarms and to increase the detection rate. In order to improve the performance, different techniques have been used in recent works. Analyzing huge network traffic data is the main work of intrusion detection system. Awell-organized classification methodology is required to overcome this issue. This issue is taken in proposed approach. Machine learning techniques like Support Vec- tor Machine (SVM) and Naive Bayes are applied. These techniques are well-known to solve the classification problems. For evaluation of intrusion detection system, NSL– KDD knowledge discovery Dataset is taken. The outcomes show that SVM works better than Naive Bayes. To perform comparative analysis, effective classification methods like Support Vector Machine and Naive Bayes are taken, their accuracy and mis-classification rate get calculated.

Mrutyunjaya Panda, Ajith Abraham, Swagatam Das, Manas Ranjan Patra [5] Intrusion detection systems (IDSs) are currently drawing a great amount of interest as a key part of system defence. IDSs collect network traffic information from some point on the network or computer system and then use this information to secure the network. Recently, machine learning methodologies are playing an important role in detecting network intrusions (or attacks), which further helps the network administrator to take precautionary measures for preventing intrusions. In this paper, we propose to use ten machine learning approaches that include Decision Tree (J48), Bayesian Belief Network, Hybrid Naive

Bayes with Decision Tree, Rotation Forest, Hybrid J48 with Lazy Locally weighted learning, Discriminative multinomial Naive Bayes, Combining random Forest with Naive Bayes and finally ensemble of classifiers using J48 and NB with AdaBoost (AB) to detect network intrusions efficiently. We use NSL-KDD dataset, a variant of widely used KDD Cup 1999 intrusion detection benchmark dataset, for evaluating our proposed machine learning approaches for network intrusion detection. Finally, Experimental results with 5-class classification are demonstrated that include: Detection rate, false positive rate, and averagecost for misclassification. These are used to aid a better understanding for the researchers in the domain of network intrusion detection

Emad E. Abdallah, [6] provide a deep discussion of the concepts of intrusion detection systems, supervised machine learning techniques, and cybersecurity attacks. Then, concerning the application of supervised learning for intrusion detection, we cover relevant efforts. Finally, a taxonomy is provided based on these related works. Based on this taxonomy, we can conclude that the classification performanceof supervised learning algorithms is high and promising based on a study of four popular data sets in this domain: KDD’99, NSL-KDD, CICIDS2017, and UNSW- NB15. Moreover, feature selection is important and, in many cases, is needed for an enhancement in performance. Furthermore, data imbalance can be a concern, and sampling approaches can help resolve.

Manvith Pallepati, Soujenya Voggu [7] Unauthorized access to a computer network can be discovered by scanning the network traffic for evidence of maliciousactivity, which is what Network Intrusion Detection (NID) does. However, in this study, we will concentrate on the technology, development, and strategic importancethat make up the large field of Network Intrusion Detection (NID). Many new strate gies have been created in the last few years to help computer security specialists in protecting a single host or an entire network against unauthorized access, theft, and denial-of-service assaults, which are the primary causes of computer crime. Intrusion Detection is critical for both the military and commercial sectors since it is themost significant study area for the future networks’ Information Security. In this pa-per, a model is being proposed, where the data is preprocessed before training with the algorithms. A study done by comparing with other models shows that, the cur- rent model built with Random Forest can outperform other existing models built with ANN when the data is preprocessed. After building model after data pre-processing and feature extraction, we are able to achieve 98.71percent accuracy on NSL-KDDdata

TABULAR COMPARISON

Ref. No [Year of Publication]	Improvement Goal	Algorithm Used	Metric Used	Topography	Assessment Method	Usage
2021	**Efficient detection of malicious input messages**	**Rule-based Detection (Keyword Match)**	**IDS**	**Application-layer message filtering**	**Manual pattern analysis**	**Java Swing-based input evaluation**
2020	**Simulate attacks to monitor response behavior**	**Honeypot Simulation Logic**	**IDS**	**Multi-layer IDS with honeypot integration**	**Behavioral logging**	**Intrusion logging in MySQL**
2022	**Logging unauthorized access attempts**	**Rule-based Logging + MySQL DB**	**IDS**	**Backend access and session tracking**	**GUI + Database Monitoring**	**Java GUI + MyPHP + DB backend**
2022	**Secure user access and session**	**Login Authentication (Java GUI)**	**IDS**	**User-based authentication layer**	**Form-based validation**	**Java login window + credential**

DOI: 10.48175/IJARSCT-27598

	control					check
2021	Real-time classification of user inputs	Static Rule Engine (Manual Rules)	IDS	Lightweight text input classification	GUI output + Alerting	IDS rule engine integrated with GUI
2023	Modular design for future ML integration	Extendable Classifier Framework	IDS	Modular Java Architecture	Software Design Principles	Eclipse project with clear modules

Figure 2.1: Tabular comparison

III. DESIGN DETAILS

Java Swing used for UI

JDBC used for MySQL connectivity

Pattern matching logic built using regular expressions

Servlet-based interaction for login validation

DATA FLOW DIAGRAM

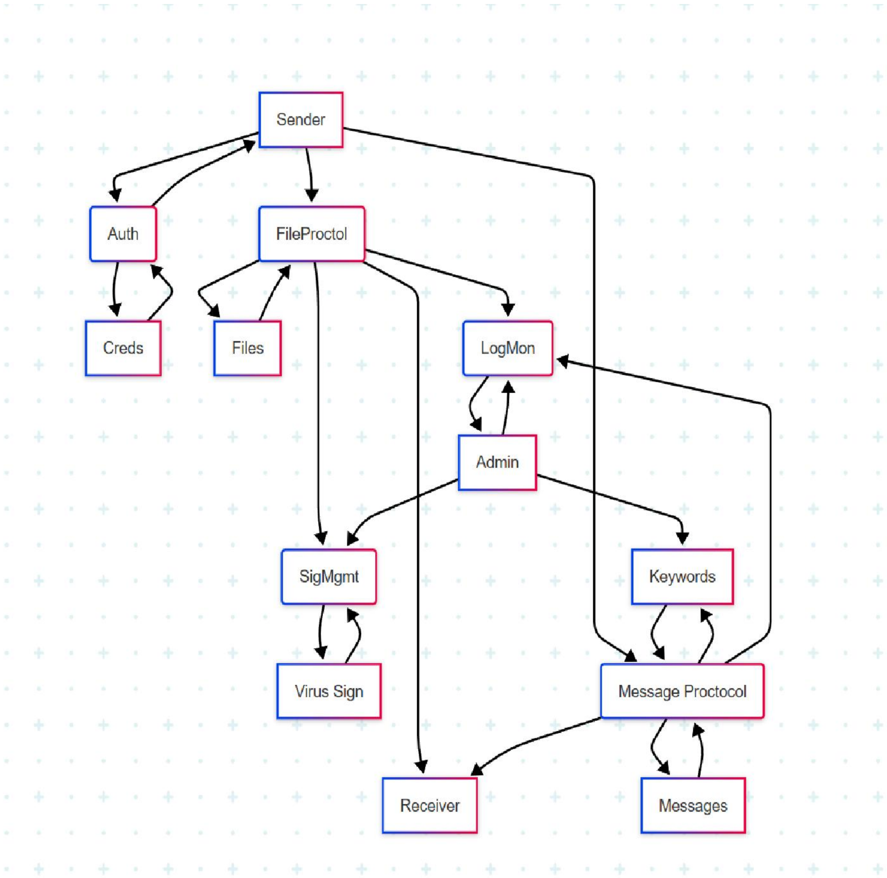

Figure: DFD

ACTIVITY DIAGRAM

Activity diagrams are graphical representations of workflows of step wise activitiesand actions with support for choice, iteration and concurrency. In the Unified Mod eling Language, activity diagrams are intended to model both computational and organizational processes (i.e workflows), as well as the data flows intersecting with the related activities. Although activity diagrams primarily show the overall flow of control, they can also include elements showing the flow of data between activities through one or more data stores.

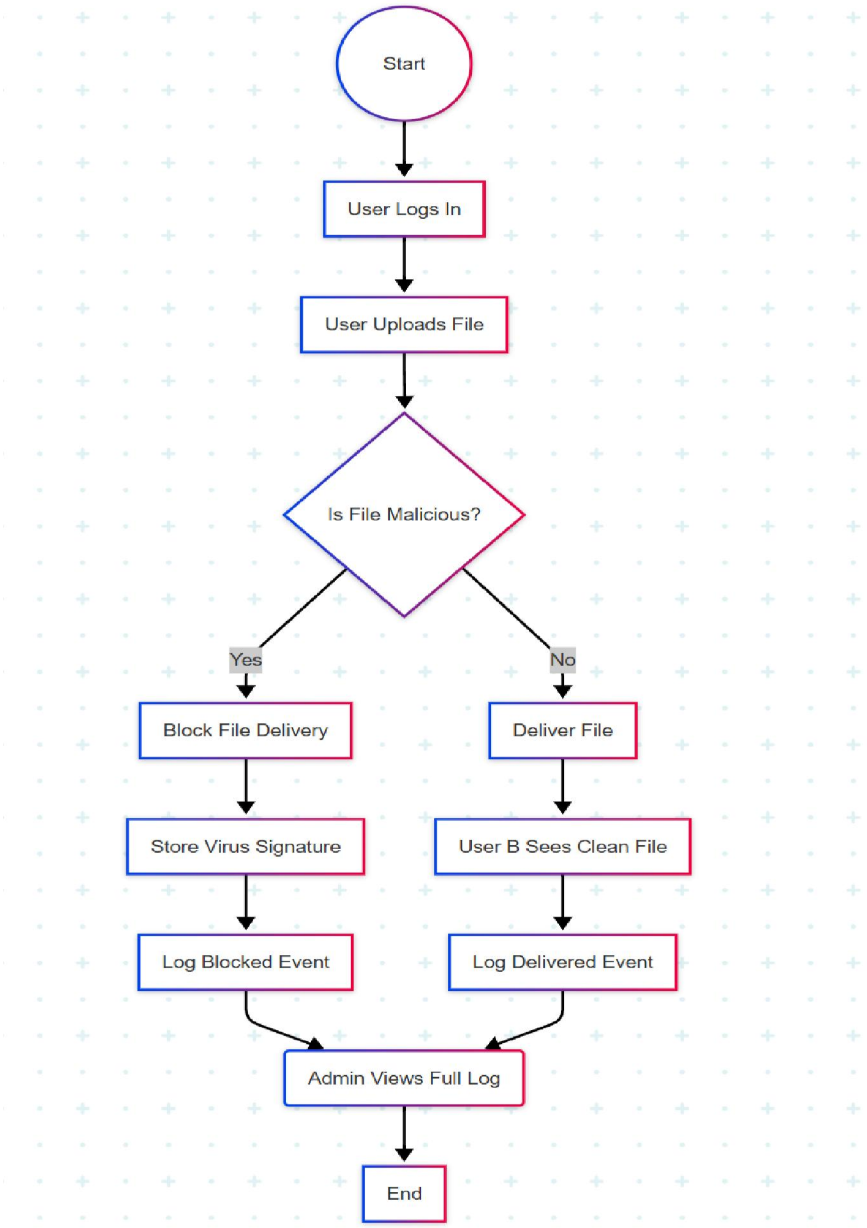

Figure : Activity Diagram

DOI: 10.48175/IJARSCT-27598

IV. CONCLUSION

The proposed project, "Enhanced Security Framework for Multi-Layered Wireless Communication Systems," presents a practical, modular, and efficient approach to network security by focusing on the development of a Java-based Intrusion Detection System (IDS). In a digital landscape where cyber threats are becoming increasingly sophisticated, this project aims to offer a simplified yet functional solution that can be effectively used in academic institutions, small-scale enterprises, or as a foundational prototype for more advanced IDS models.

This project successfully implements a lightweight IDS using Java Swing for the user interface, MySQL for backend storage, and rule-based keyword detection logic to identify potentially malicious inputs. The inclusion of a honeypot mechanism further strengthens the system by monitoring and logging unauthorized login attempts, thereby enhancing the system's ability to detect and record intrusion activities.

The application architecture ensures ease of use and extendability. The graphical interface allows both users and administrators to interact with the system seamlessly. The keyword-based detection approach ensures high-speed message evaluation and eliminates the need for training data or complex model tuning. The honeypot integration adds an additional layer of monitoring by capturing abnormal access behavior and assisting in proactive threat identification.

In addition, the project fulfills all essential security principles, namely: Confidentiality (restricting access to authorized users), Integrity (ensuring data has not been tampered with), and Availability (ensuring continuous and reliable access to system resources). All logs—valid messages, malicious inputs, and unauthorized attempts—are securely stored in the backend database to maintain traceability and accountability.

Looking ahead, this framework serves as a solid foundation for further expansion. It can be enhanced by integrating machine learning models for dynamic threat detection, natural language processing (NLP) for more intelligent keyword matching, and blockchain-based logging mechanisms to prevent log tampering and ensure data immutability. The modular structure also allows easy incorporation of advanced features such as network packet analysis, real-time alerts, and integration with third-party firewall systems.

In conclusion, this project not only achieves its immediate objective of designing a functional IDS with essential detection and logging capabilities but also opens the door for future innovation in the field of cybersecurity. It demonstrates how a minimal yet thoughtfully designed solution can contribute effectively toward securing digital communication systems.

DOI: 10.48175/IJARSCT-27598