Intrusion Detection System using Machine Learning

Karedla Rupa Sri[1], Neelapu Vasanthi[2], Jagarapu Susmitha Laxmi[3], Gandi Venkata Satya Sai Varshith[4]
[1,2,3,4]*Department of Computer Science & Engineering, Avanthi Institute of Engineering and Technology*

***Abstract*—With the rapid growth of networked systems and internet-based services, cyber threats such as Distributed Denial of Service (DDoS), brute force attacks, and data exfiltration have become increasingly sophisticated. Traditional signature-based Intrusion Detection Systems (IDS) are ineffective against unknown and zero-day attacks. This paper presents a real-time Intrusion Detection System using Machine Learning trained on the CICIDS2017 dataset. The proposed system captures live network traffic using Scapy, extracts flow-based statistical features, and classifies traffic using a Random Forest classifier. The system provides real-time attack detection with confidence scores and supports offline training and evaluation. Experimental results demonstrate high detection accuracy and efficient real-time performance.**

***Index Terms*—Intrusion Detection System, Machine Learning, Random Forest, Network Security, CICIDS2017**

I. INTRODUCTION

The increasing dependence on computer networks and cloud-based services has led to a surge in cyber-attacks targeting organizations and individuals. Intrusion Detection Systems (IDS) are essential for monitoring network traffic and identifying malicious activities. Traditional IDS solutions, such as Snort, rely on signature-based detection, which is ineffective against previously unseen attacks.

Machine learning offers an adaptive approach to intrusion detection by learning patterns from network traffic data and identifying anomalies. The objective of this research is to design and implement a real-time IDS that combines flow-based feature extraction with machine learning classification to detect multiple attack types in live network environments.

II. LITERATURE REVIEW

Several studies have explored the use of machine learning in intrusion detection systems.
Paper 1: Studies using the KDD Cup 1999 dataset demonstrated that machine learning algorithms can outperform rule-based systems but suffer from outdated attack patterns.

Paper 2: The introduction of the NSL-KDD dataset addressed redundancy issues and improved model generalization.

Paper 3: Recent research using the CICIDS2017 dataset provides realistic network traffic with up-to-date attack scenarios, making it suitable for evaluating modern IDS systems.

These studies highlight the effectiveness of ensemble learning techniques, particularly Random Forest, in handling high-dimensional network traffic data.

III. PROBLEM STATEMENT

Traditional intrusion detection systems face several limitations:

- Dependence on predefined attack signatures
- High false-positive rates
- Inability to detect zero-day attacks
- Lack of real-time detection capability

Furthermore, many existing machine learning IDS models operate only in offline environments and cannot process live network traffic. Therefore, there is a need for a real-time intrusion detection system that monitors network packets, extracts meaningful features, and classifies traffic with minimal latency.

IV. PROPOSED SYSTEM

The proposed system is a real-time intrusion detection framework that integrates packet capture, flow-based feature extraction, and machine learning classification. The system operates in two phases:
Training Phase: Historical network traffic data is used to train a Random Forest classifier.
Detection Phase: Live network packets are captured and processed to detect intrusions in real time.

The system is designed to detect various attack categories, including DDoS, brute-force, and port-scanning attacks, while maintaining high throughput and low computational overhead. This architecture enables the system to adapt to dynamic network environments and identify both known and previously unseen attack patterns effectively.

V. SYSTEM ARCHITECTURE

The architecture of the proposed IDS consists of the following layers:

- Packet Capture Layer: Captures live packets from the network interface.
- Flow Generation Layer: Groups packets into flows based on the 5-tuple (source IP, destination IP, source port, destination port, protocol).
- Feature Extraction Layer: Computes statistical features such as flow duration, packet count, and inter-arrival time.
- Classification Layer: Uses a trained Random Forest model to classify traffic as benign or malicious.
- Alert Layer: Displays real-time alerts for detected attacks.

This layered architecture ensures modularity, scalability, and efficient processing of network traffic.

VI. METHODOLOGY

The methodology adopted in this project consists of the following steps:

- Collection of network traffic dataset
- Data preprocessing and cleaning
- Feature selection and scaling
- Model training and evaluation
- Real-time deployment and testing

The training dataset is cleaned by removing duplicate records, handling missing values, and encoding categorical labels. Feature selection is performed to retain only high-variance features to improve training efficiency.

VII. IMPLEMENTATION

The system is implemented using Python and open-source libraries.

- Dataset Processing: Implemented using pandas
- Model Training: Implemented using scikit-learn
- Packet Capture: Implemented using Scapy

The IDS captures packets in real time, constructs flow, extracts features, and feeds them into the trained machine learning model for classification.

VIII. MODULES

The system is divided into the following functional modules:

- Data Preprocessing Module: Cleans and prepares the dataset
- Model Training Module: Trains and evaluates the machine learning classifier
- Prediction Module: Performs offline prediction on CSV traffic logs
- Live Detection Module: Captures and analyzes real-time network traffic

Each module is implemented as a standalone Python script to maintain modularity and reusability.

IX. TESTING

The system is tested using both offline datasets and live network traffic.

- Unit Testing: Individual modules were tested independently
- Integration Testing: Ensured correct interaction between data preprocessing, model training, and detection modules
- Performance Testing: Evaluated detection latency and throughput during live traffic monitoring

The model achieved high classification accuracy and maintained stable performance under continuous packet capture.

X. RESULTS

Experimental results show that the Random Forest classifier achieved high accuracy and recall on the CICIDS2017 dataset. The real-time IDS successfully detected simulated attack traffic including port scans and abnormal traffic bursts.

The system demonstrated low latency in packet processing and provided real-time alerts without significant packet loss.

XI. APPLICATIONS

The proposed IDS can be deployed in various environments:

- Enterprise networks
- Cloud data centers
- Educational institutions
- IoT network monitoring systems

It is particularly useful in detecting previously unseen attacks due to its machine learning-based detection mechanism.

XII. LIMITATIONS

Despite its effectiveness, the system has certain limitations:

- Limited feature set compared to full CICIDS feature space
- Requires administrative privileges for packet capture
- Model performance depends on training data quality

XIII. FUTURE WORK

Future enhancements to the system may include:

- Integration with Security Information and Event Management (SIEM) tools
- Deep learning-based anomaly detection
- Web-based monitoring dashboard
- Distributed deployment for large-scale network