Intrusion Detection System for Port Scanning Attacks

Prof. Bharti Bandgar[1], Kaivalya Kulkarni[2], Prasad Waghmare[3], Onkar Pomane[4], Shubham Waghmare[5]
[1]*Prof. Department of Computer Engineering, K. J. College of Engineering & Management Research, Pune, India*
[2,3,4,5]*Department of Computer Engineering, K. J. College of Engineering & Management Research, Pune, India*

***Abstract*—In today's world, cyberattacks are hitting computer networks harder than ever, and old-school defences like firewalls or basic rule-checking just aren't cutting it against sneaky, advanced intruders. Hackers often start with stealthy recon moves, like port scanning, that slip right past traditional setups.Our project introduces a smart Machine Learning-powered Intrusion Detection System (IDS) that keeps a real-time eye on network traffic to spot these malicious scans. It grabs live packets using Scapy, pulls out key network features, and sorts good traffic from bad with a Random Forest classifier. To cut down on false alarms, we added flow-based analysis that looks at patterns across multiple packets in a short time window, instead of judging each one alone. The system fires off instant alerts with clear visual cues and keeps thorough logs for digging into incidents later. Overall, this IDS boosts detection rates, slashes false positives, and delivers a lightweight, practical tool ready for real-world use.**

***Index Terms*—Intrusion Detection System, Network Security, Machine Learning, Random Forest, Port Scan Detection, Real-Time Monitoring.**

I. INTRODUCTION

As computer networks continue to expand across industries such as banking, healthcare, education, and e-commerce, the risk of cyber intrusions has grown significantly. Attackers frequently perform reconnaissance attacks such as port scanning to identify vulnerable services before launching more severe exploits. Detecting such early-stage attacks is crucial for preventing major security breaches.

Traditional network security tools like firewalls and access control lists operate based on predefined rules and lack the intelligence to analyze traffic behavior dynamically. These systems often fail to detect internal threats, zero-day attacks, or low-and-slow reconnaissance techniques. As a result, malicious activities may remain unnoticed until substantial damage has already occurred.

An Intrusion Detection System (IDS) plays a vital role by continuously monitoring network traffic and identifying suspicious behavior. Recent advancements in machine learning have enhanced IDS capabilities by enabling systems to learn traffic patterns and distinguish between normal and malicious behavior. This project focuses on developing a machine learning driven IDS that detects port scanning activities in real time with improved accuracy and reduced false positives.

That's where an Intrusion Detection System (IDS) steps in, constantly watching traffic for anything fishy. Machine learning has supercharged these systems lately, teaching them to spot normal patterns versus malicious ones. Our project builds a ML-driven IDS that catches port scanning in real time, with sharper accuracy and way fewer false alarms.

II. PROBLEM DEFINITION

With the rapid growth of computer networks, cyberattacks such as port scans, reconnaissance, and unauthorized access attempts are becoming increasingly frequent. Traditional signature-based Intrusion Detection Systems (IDS) struggle to detect novel or evolving attacks, often generating a high number of false positives.

The challenge is to design a real-time, intelligent IDS capable of analyzing live network traffic, identifying suspicious activities, and generating timely alerts. The system should automatically differentiate between normal traffic and potential attacks by learning traffic patterns from historical data.

III. OBJECTIVE

Develop a machine learning-based IDS that captures live TCP/IP traffic, extracts meaningful features, and detects potential intrusions—especially port scans—while minimizing false alarms, providing network administrators with accurate and actionable alerts.

- Develop a Real-Time Intrusion Detection System (IDS) capable of monitoring live network traffic continuously.
- Detect reconnaissance attacks, such as port scans, by analyzing traffic patterns in real time.
- Apply machine learning techniques, specifically Random Forest classification, to differentiate between normal and malicious network behavior.
- Extract and engineer meaningful features from raw network packets (e.g., TCP flags, packet length, inter-arrival time) for accurate detection.
- Implement flow-based correlation to reduce false positives by considering multiple packets from the same source within a short time window.
- Generate real-time alerts and maintain logs with clear source and destination details for forensic and administrative purposes.

IV. LITERATURE REVIEW

Research in Intrusion Detection System has primarily focused on two major approaches: signature-based and anomaly-based detection. Signature-based systems, such as Snort, identify threats by comparing network traffic against known attack patterns. These systems are efficient for detecting previously identified attacks but require continuous updates and are ineffective against new or unknown threats. On the other hand, anomaly-based systems analyze deviations from normal behavior and often incorporate techniques from Machine Learning. While they can detect novel attacks, they tend to suffer from higher false-positive rates and require extensive training data.

Advanced IDS platforms like Suricata have improved detection capabilities by supporting multi-threading and deep packet inspection. These systems provide better performance and scalability compared to earlier tools but come with increased complexity and resource requirements. Additionally, tools such as Wireshark are widely used for packet-level traffic analysis, allowing detailed inspection of network data. However, they are mainly designed for manual monitoring and lack automated intrusion detection features, making them less suitable for real-time security enforcement.

Recent studies have explored lightweight and flexible IDS solutions using programming-based approaches. Scapy has gained attention as a powerful tool for packet sniffing and custom rule implementation in Python. It enables real-time traffic analysis and allows researchers to design tailored detection mechanisms for specific attack scenarios such as port scanning and ICMP flooding. Although Scapy-based systems are easy to implement and resource-efficient, they are generally limited in scalability and lack the advanced detection capabilities of enterprise-level IDS solutions, highlighting the need for further research in this area.

V. PROPOSED SYSTEM

The proposed system is a real-time machine learning based Intrusion Detection System (IDS) designed to monitor live network traffic and identify potential attacks such as port scans. It captures packets using Scapy, extracts relevant features like TCP flags, packet length, and timing, and applies a Random Forest classifier to determine whether the traffic is normal or suspicious. To enhance accuracy and reduce false positives, the system incorporates flow-based correlation, analyzing multiple packets from the same source over a defined time window. Alerts are generated in real time with clear visualization and logged for future investigation. The system is lightweight, efficient, and suitable for deployment on practical environments like Kali Linux, ensuring timely detection and minimal performance overhead.

5.1 System Architecture

Packet Capture Module:

The system captures live network packets in real time using Scapy. It continuously monitors TCP/IP traffic across the network interface and forwards packets to the feature extraction module for further analysis. This ensures no network activity goes unnoticed.

Feature Extraction Module:

Extracted packets are processed to generate numerical features such as TCP flags (SYN, ACK), packet length, inter-arrival times, and connection frequency.

These features form structured input suitable for machine learning analysis.

Machine Learning Module:
The pre-processed features are fed into a pre-trained Random Forest classifier. The classifier predicts whether the traffic is normal or malicious, providing a probabilistic score for detection.

Flow Correlation Module:
To reduce false positives, this module analyzes traffic behavior over a short time window. It tracks multiple packets from the same source IP, counting unique destination ports and repeated patterns to validate potential scanning activity.

Alert and Logging Module:
When malicious behavior is detected, real-time alerts are generated with colored visualization, including source IP, destination IP, and port details. All events are logged for forensic analysis and audit purposes, enabling network administrators to respond promptly.

System Control Module:
Integrates all modules and ensures smooth data flow from packet capture to alert generation. It manages timing, packet buffering, and coordination between detection modules to maintain efficient real-time performance.

5.2 System Workflow
The operation of the proposed Intrusion Detection System is carried out through a clear stepwise procedure, ensuring accurate identification of malicious traffic while minimizing false positives.
1. Packet Capture:
The IDS continuously monitors network traffic in real time using Scapy, capturing all TCP/IP packets from the active network interface.

2. Feature Extraction:
Each captured packet is processed to extract relevant numerical features, including TCP flags (SYN, ACK), packet length, inter-arrival time, and connection patterns.

3. Data Preprocessing:
Extracted features are scaled and normalized to ensure consistency with the training dataset, preparing them for accurate model predictions.

4. Machine Learning Prediction:
The pre-trained Random Forest classifier evaluates the scaled features and predicts whether the traffic is normal or suspicious.

5. Flow-Based Correlation:
To reduce false positives, traffic from the same source IP is analyzed over a short time window. Unique destination ports and repeated patterns are tracked to confirm potential scanning activity.

6. Alert Generation and Logging:
When malicious behavior is detected, real-time alerts are generated with colored visualization, including source and destination information, while all events are securely logged for further analysis.

7. System Control:
Ensures smooth coordination between modules, manages packet flow, and maintains real-time detection efficiency across the network.
This workflow ensures that only suspicious or malicious traffic is flagged, thereby providing a robust and automated network security monitoring system.

5.3 Advantages of the Proposed System
- Real-Time Detection: Continuously monitors network traffic and detects malicious activity immediately.
- High Accuracy: Combines Random Forest classification with flow-based correlation to reduce false positives.
- Lightweight and Efficient: Runs on Kali Linux without heavy computational overhead.
- Automated Alerts and Logging: Generates real-time colored alerts and maintains detailed logs for forensic analysis.
- Scalable and Modular: Architecture allows integration of additional attack detection features or advanced ML models in the future.

VI. METHODOLOGY AND IMPLEMENTATION

The proposed Intrusion Detection System (IDS) is implemented as a real-time network monitoring tool using a combination of packet capture, feature extraction, machine learning classification, and flow-

based correlation. The system follows a structured methodology:

1.Packet Capture
Live network traffic is captured using the Scapy library on a Kali Linux environment. TCP/IP packets are collected in real time from the active network interface. This module ensures continuous monitoring of the network without dropping packets and supports multiple network protocols for comprehensive analysis.

2. Feature Extraction
Relevant features are extracted from each packet, including TCP flags (SYN, ACK), packet length, inter-arrival time, and connection frequency. These features form the numerical input required for the machine learning model. Feature engineering also considers behavioral patterns, such as burst traffic or multiple port access from a single source.

3.Data Preprocessing
The extracted features are normalized using a StandardScaler to maintain consistency with the training dataset, ensuring accurate predictions by the model. Preprocessing also handles missing values or anomalies to maintain the integrity of input data for reliable detection.

4. Machine Learning Classification
A pre-trained Random Forest classifier evaluates the scaled features to classify network traffic as normal or suspicious. Probabilistic predictions are used to quantify the likelihood of malicious activity, allowing the system to tune alert thresholds based on network conditions.

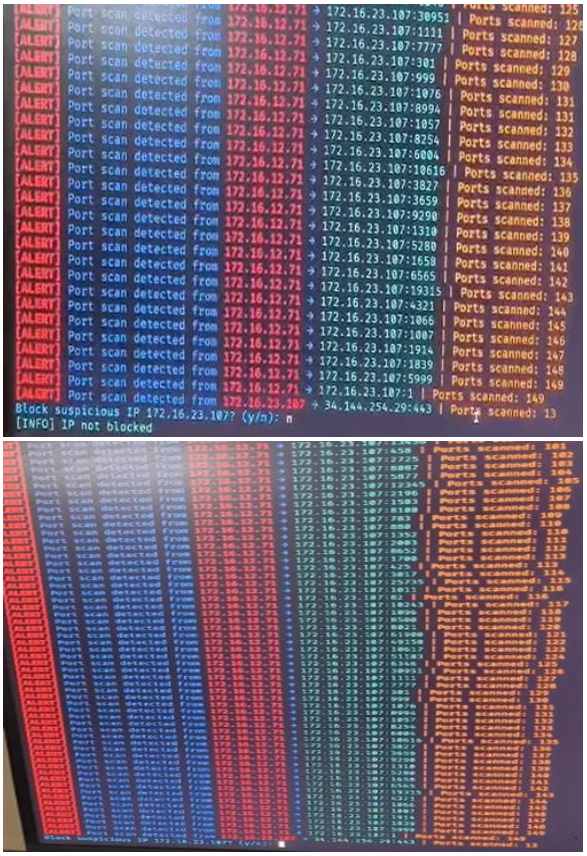
5. Flow-Based Correlation
To reduce false positives, multiple packets from the same source IP are tracked within a defined time window. Unique destination ports and repeated access patterns are analyzed to confirm potential scanning activity. This module allows the system to detect low-and-slow reconnaissance attacks that might be missed by single-packet analysis.

6. Alert Generation and Logging
When the system detects malicious traffic, it generates real-time alerts with colored visualization and logs detailed information for further forensic investigation. The alerts include source and destination IPs, targeted ports, and the number of suspicious packets, which helps administrators prioritize responses.

7. Implementation Details
- Programming Language: Python 3.x
- Packet Capture Library: Scapy
- Machine Learning Library: scikit-learn (Random Forest)
- Execution Environment: Kali Linux (Live monitoring)

The implementation is modular, allowing independent updates to the packet capture, feature extraction, or classification modules for easy future enhancements and scalability.

VII. RESULTS AND DISCUSSION



The performance of the proposed real-time intrusion detection system was evaluated under practical network conditions. The outcomes obtained from the experimental analysis are discussed below.

1. Intrusion Detection Accuracy

The proposed intrusion detection system demonstrated a high level of accuracy in identifying malicious network activities, particularly port scanning behavior. The use of a machine learning-based classifier significantly improved the system's ability to distinguish between normal and suspicious traffic patterns.

2. Flow-Based Detection Effectiveness

The effectiveness of the flow correlation mechanism was analyzed. It was observed that the system successfully identified scanning behavior by monitoring multiple connection attempts from a single source within a defined time window. This approach helped in reducing false positives and ensured more reliable detection compared to single-packet analysis.

3. Real-Time Performance

The system exhibited low processing latency during live packet capture and analysis. The detection and alert generation were performed in near real-time, demonstrating that the system meets the requirements for continuous network monitoring.

4. Error Analysis (False Positives and False Negatives)

The evaluation of false positive and false negative rates indicated that the system maintains a balanced trade-off between detection sensitivity and accuracy. The integration of machine learning with flow-based filtering reduced unnecessary alerts while still detecting genuine intrusion attempts effectively.

5. Robustness Under Network Variations

The system was tested under different network conditions, including normal browsing, secure connections, and simulated attack scenarios. The IDS maintained stable performance across varying traffic patterns, confirming its robustness in practical environments.

6. Comparative Performance

The proposed system was compared with traditional rule-based intrusion detection approaches. The results showed that the machine learning-based IDS provided improved detection capability, especially for unknown or dynamic attack patterns. Additionally, the inclusion of behavioral analysis made the system more adaptable and reliable than conventional methods.

VIII. CONCLUSION

This project successfully implements a Machine Learning based Intrusion Detection System capable of detecting port scanning attacks in real time. By combining Random Forest classification with flow-based traffic correlation, the system achieves high detection accuracy while minimizing false positives. The lightweight design makes it suitable for deployment in real-world environments without requiring extensive computational resources. Future enhancements may include intrusion prevention mechanisms, multi-class attack detection, and integration with centralized security monitoring platforms.

REFERENCES

[1] "A new intrusion detection method using ensemble classification and CNN–RF hybrid," 2025.
[2] "Intrusion detection system for IoT based on modified random forest," 2025.
[3] "A cost-effective machine learning-based network intrusion detection," 2025.
[4] "Hybrid improved binary GWO–PSO with random forest for intrusion detection systems," 2026.
[5] "Enhanced intrusion detection system using SVM and random forest on UNSW-NB15 dataset," 2025.
[6] "Machine learning model for flow-based network threats," 2023.
[7] "Advanced random forest for real-time intrusion detection," 2024.
[8] "Machine learning-based multi-stage intrusion detection system with random forest," 2025.
[9] "Reproducing random forest efficacy in detecting port scanning," 2023.
[10] "Intrusion detection using hybrid random forest for network flow monitoring," 2025.