INTERNATIONAL JOURNAL OF NOVEL RESEARCH AND DEVELOPMENT (IJNRD)
© 2026 IJNRD | Volume 11, Issue 4, April 2026 | ISSN: 2456-4184 | IJNRD.ORG

ABNORMAL NETWORK DATA TRAFFIC DETETCTION

A Rule-Based Network Traffic Analysis System for Detecting Abnormal Activities Using PCAP Files

Aditi Anil Botke , Atharva Hemant Chavan,
Msc Cyber Security Students
Department Of Information And Cyber Security,
Guru Nanak Khalsa College, Matunga, Mumbai
Guide: Prof. Randeep Singh Ghai, Jasbir Kaur

ABSTRACT:
Network traffic monitoring is essential for identifying abnormal activities such as denial-of-service attacks, port scanning, and unauthorized access. While many modern intrusion detection systems rely on machine learning techniques, they often require large datasets and high computational resources. This paper proposes a **rule-based backend system** for analyzing network traffic using PCAP files.

The system extracts packet-level features such as source IP, destination port, protocol, and packet frequency, and applies predefined threshold rules to classify traffic as normal or abnormal. Alerts are generated for suspicious behavior based on rule violations. The proposed approach is lightweight, transparent, and suitable for educational environments and small-scale networks, offering an effective alternative to complex machine learning-based solutions.

KEYWORDS
Network Security, PCAP Analysis, Rule-Based Detection, Intrusion Detection System, Abnormal Traffic, Python

INTRODUCTION
Modern networks face continuous security threats due to increasing internet usage and connected devices. Network traffic monitoring plays a vital role in identifying malicious activities before they cause serious damage. Packet Capture (PCAP) files store detailed information about network communication and are widely used in traffic analysis and forensic investigations.

Most recent intrusion detection systems use machine learning algorithms to classify network traffic. While effective, these methods require extensive datasets, computational power, and expertise, making them unsuitable for beginners or small organizations. This research proposes a **rule-based backend system** that performs traffic classification using simple logical rules derived from known network behavior patterns.

The objective of this project is to design and implement a backend system that can differentiate between normal and abnormal network traffic using PCAP files without relying on machine learning techniques.

NEED OF THE STUDY
With the rapid growth of digital systems, network security has become a major concern. The increase in cyberattacks such as unauthorized access, DDoS attacks, and data breaches has highlighted the need for detecting abnormal network traffic.

Abnormal traffic refers to unusual patterns in network data that may indicate malicious activity. Network administrators and security analysts come into contact with such traffic during monitoring, analysis, and response processes.

Proper knowledge of abnormal traffic detection is important to improve network security. Traffic requiring attention includes unusual spikes, suspicious IP behavior, unauthorized access attempts, and protocol anomalies. Detecting such activities helps in preventing cyber threats and protecting network systems.

3.1 Population and Sample
The population of the study consists of the complete set of network traffic data generated within a network environment. This includes all types of data packets, communication patterns, and network activities such as normal traffic and abnormal traffic.

From this population, a sample dataset is selected for analysis. The sample includes specific network traffic data containing both normal behavior and abnormal patterns such as intrusion attempts, suspicious activities, or unusual traffic spikes.

The selected sample is based on data availability and relevance, and it represents real-world network scenarios for detecting abnormal traffic behavior

3.2 Data and Sources of Data

This study uses secondary data collected from publicly available datasets and network monitoring tools.

The data includes features such as IP addresses, protocols, packet size, timestamps, and traffic volume. It represents both normal and abnormal traffic behavior over a defined period.

3.3 Theoretical Framework

The study includes dependent and independent variables.
The dependent variable is **network traffic classification** (normal or abnormal). Independent variables include packet size, traffic volume, IP addresses, and protocol types.

Abnormal traffic is identified based on deviation from normal patterns, helping in detecting potential threats in the network.

LITERATURE REVIEW

Roesch [1] introduced Snort, a lightweight rule-based intrusion detection system that performs packet-level inspection using predefined signatures. Sommer and Paxson [2] highlighted the challenges of applying machine learning techniques to real-world network intrusion detection, emphasizing issues of data quality and model generalization. Axelsson [3] provided a comprehensive taxonomy of intrusion detection systems, categorizing them into signature-based and anomaly-based approaches.

1. Many studies use **machine learning models** like Random Forest, SVM, and Neural Networks for intrusion detection.
2. These systems achieve high accuracy but suffer from **lack of explainability**.
3. Some rule-based systems exist but are often integrated into large IDS tools like Snort.
4. Limited research focuses on **simplified backend implementations for educational purposes**.

Gap: **Simple, transparent, beginner-friendly backend systems are underexplored.**

CONCLUSION OF RESEARCH PAPERS (Literature Conclusion)

From the literature survey, it is observed that while machine learning-based intrusion detection systems dominate current research, they introduce challenges such as complexity, data dependency, and lack of interpretability. This motivates the need for a simpler, rule-based approach that can provide effective detection with lower resource requirements.

OBJECTIVES OF THE STUDY

- To analyze network traffic using PCAP files
- To classify traffic as normal or abnormal using rule-based logic
- To generate alerts for suspicious activities
- To design a simple and explainable backend system
- To avoid the complexity of machine learning techniques

HYPOTHESES

- H1: Rule-based traffic analysis can effectively detect abnormal network behavior.
- H2: A non-ML approach can reduce system complexity and computational overhead.
- H3: Simple threshold-based rules are sufficient for detecting common network attacks.

RESEARCH METHODOLOGY

PCAP (Packet Capture) files are collected from publicly available cybersecurity datasets. These files contain real or simulated network traffic, including both normal and malicious activities, which are used as input for analysis.
The PCAP files are processed using tools like Scapy or PyShark to extract useful information such as source and destination IP addresses, ports, protocols, packet size, and timestamps.
Rules are defined based on common network behavior patterns. For example, excessive requests from a single IP, unusual port access, or abnormal traffic spikes are considered indicators of suspicious activity.

Based on the defined rules, the network traffic is classified into normal and abnormal categories. Each packet or flow is analyzed and labeled accordingly.
When abnormal traffic is detected, alerts are generated. These alerts may include details such as suspicious IP address, type of anomaly, and timestamp for further investigation.
The results are analyzed to evaluate the effectiveness of the detection system. This includes checking how accurately abnormal traffic is identified and assessing system performance.

TECHNOLOGIES USED

Python: Used as the primary programming language for implementing the detection logic due to its simplicity and strong library support.
Scapy / PyShark: Used for packet parsing and analysis. These tools help in reading PCAP files and extracting detailed network information.
PCAP files: Serve as the main data source containing captured network traffic for analysis.
JSON for output: Used to store and structure the output data, making it easy to read, share, and integrate with other systems.
Flask (optional for frontend integration): Used to create a simple web interface for displaying alerts and analysis results.

DATA ANALYSIS

Packet count per IP : Measures the number of packets sent by each IP address to identify unusual activity or flooding.
Protocol distribution : Analyzes the usage of different protocols (TCP, UDP, ICMP) to detect abnormal protocol behavior.
Port access frequency : Monitors how frequently specific ports are accessed, helping to identify port scanning or unauthorized access attempts.
Traffic rate analysis: Examines the rate of data transfer over time to detect sudden spikes or irregular patterns.
Traffic exceeding predefined thresholds is marked as abnormal, indicating potential threats.

CHALLENGES

Large PCAP file size, handling large datasets can slow down processing and require more memory and storage.
Noise in network traffic, normal traffic may sometimes appear suspicious, making it difficult to distinguish between genuine and malicious activity. Selecting appropriate thresholds, choosing correct threshold values is critical, as too high or too low values can affect detection accuracy. Avoiding false positives, ensuring that normal traffic is not incorrectly classified as abnormal is a key challenge. Manual rule tuning, rules need to be adjusted and refined manually, which can be time-consuming.

FINDINGS

Rule-based detection is effective for known attack patterns, it works well in identifying predefined threats such as flooding or port scanning.
System is lightweight and fast, the approach does not require heavy computation, making it efficient.
Easy to understand and explain, rule-based logic is simple and transparent compared to complex models.
Suitable for educational and small network environments, it is ideal for learning purposes and small-scale implementations.

Mini Algorithm

1. Load PCAP file
2. Extract packet features
3. Apply detection rules
4. Classify traffic (Normal/Abnormal)
5. Generate alert if abnormal
6. Store results in JSON

Sample Output (alerts.json)

```
[
  {
    "type": "High Traffic from IP",
    "ip": "192.168.1.5",
    "packet_count": 150
  },
  {
    "type": "Suspicious Port Activity",
    "port": 80,
    "access_count": 70
```

FUTURE SCOPE

Integration of machine learning models, advanced models can be used to detect unknown or complex attack patterns.
Real-time packet capture, instead of using stored pcap files, live traffic can be analyzed for instant detection.
Dashboard visualization, interactive dashboards can be developed to display traffic patterns and alerts visually.
Hybrid rule + ml system, combining rule-based and machine learning approaches can improve accuracy.
Cloud-based deployment, the system can be deployed on cloud platforms for scalability and remote access.

REFERENCES

[1] Rule-Based Intrusion Detection (Very Important)
M. Roesch,
"Snort: Lightweight Intrusion Detection for Networks,"
Proceedings of the 13th USENIX Conference on System Administration, 1999.

[2] Signature & Rule-Based Network Detection
R. Sommer and V. Paxson,
"Outside the Closed World: On Using Machine Learning for Network Intrusion Detection,"
IEEE Symposium on Security and Privacy, 2010.

[3] Network Traffic Analysis Using PCAP
S. Axelsson,
"Intrusion Detection Systems: A Survey and Taxonomy,"
Technical Report, Chalmers University of Technology, 2000.

[4] Packet-Based Anomaly Detection
T. Limmer and F. Dressler,
"Survey of Graph-Based Network Anomaly Detection,"
Computer Communications, Elsevier, 2014.

[5] PCAP and Traffic Monitoring
G. Varghese and C. Estan,
"New Directions in Traffic Measurement and Accounting,"
ACM SIGCOMM, 2004.

[6] Rule-Based vs ML IDS
N. Moustafa and J. Slay,
"The Evaluation of Network Anomaly Detection Systems,"
IEEE International Conference on Big Data, 2015.

[7] Offline Traffic Analysis Using PCAP
Wireshark Foundation,
"Wireshark User's Guide,"
https://www.wireshark.org/docs/

[18] Python-Based Packet Analysis
K. Scapy,
"Scapy: Packet Manipulation Tool,"
https://scapy.net/