NETWORK SECURITY SIMULATION

A CASE STUDY REPORT

Submitted by

KUMARI ANANYA[RA2211026010441] ADITYA KUMAR SINGH[RA2211026010418] SHREYANSH CHOUBEY[RA2211026010429]

Brijesh J [RA2211026010443]

for the course

21CSC302J – COMPUTER NETWORKS

in partial fulfillment of the requirements for the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF COMPUTING

TECHNOLOGIES SCHOOL OF COMPUTING

FACULTY OF ENGINEERING AND TECHNOLOGY

SRM INSTITUTE OF SCIENCE AND

TECHNOLOGY KATTANKULATHUR - 603 203.

SRMINSTITUTEOFSCIENCEANDTECHNOLOGY
KATTANKULATHUR–603203

BONAFIDE CERTIFICATE

Certified that Computer Network A Case Study Report titled "**network security intrusion detection** " is the bonafide work of **KUMARI ANANYA[RA2211026010441] ADITYA KUMAR SINGH[RA2211026010418] SHREYANSH CHOUBEY[RA2211026010429]**

BRIJESH.J[RA2211026010443]

who carried out the case study under my supervision.

Certified further, that to the best of my knowledge the work reported herein does not form any other work

SIGNATURE

Dr. B. Hariharan

Course Faculty

Associate Professor

Department of Computational Intelligence

SRM Institute of Science and Technology
Kattankulathur

ABSTRACT

This case study delves into network security using a simulated Intrusion Detection System (IDS) to monitor, detect, and respond to cyber threats within a controlled network environment. The rapid digitalization across industries has amplified the need for robust network security, as cyber threats are evolving both in frequency and sophistication. Traditional security measures, such as firewalls and antivirus software, are limited in their ability to detect and respond to complex, multi-faceted attacks. As such, Intrusion Detection Systems have become critical in identifying unusual patterns that could indicate unauthorized access, data breaches, or potential attacks.

This study is structured around the design and implementation of an IDS within a simulated network environment. The objective is to assess the IDS's effectiveness in identifying common types of attacks, such as port scans, SQL injections, and Distributed Denial of Service (DDoS) attacks. Our approach involves constructing a network with simulated legitimate traffic alongside various forms of malicious traffic. The IDS framework is designed to monitor and analyze this traffic, detect potential threats, and log relevant data for further investigation.

The IDS system employs both signature-based and anomaly-based detection methods. Signature-based detection involves comparing network traffic against a database of known attack signatures to detect threats with a high degree of accuracy for familiar attack types. However, as signature-based systems may fail to identify new or modified attacks, anomaly based detection complements this approach by flagging deviations from established patterns of normal network behavior. The integration of these two detection methods provides a balanced solution, aiming to detect a wide range of threats while minimizing false positives. Through a systematic simulation, the study evaluates the IDS's detection rate, response time, and impact on network performance. Key metrics include the percentage of correctly identified attacks (true positives), the incidence of false positives (normal traffic flagged as malicious), and resource utilization (CPU and memory usage). Results show that while the IDS successfully identifies known threats and certain anomalies, it also generates a significant number of false positives, particularly under complex traffic conditions. Additionally, the IDS's impact on network performance highlights the trade-offs between security and operational efficiency.

TABLEOFCONTENTS

TABLE OF FIGURES

2. Introduction

The introduction provides a foundation for the case study by explaining the importance of network security and IDS technology. Cover the following points:

- **Overview of Network Security**: Discuss the role of network security in protecting organizational data and infrastructure from unauthorized access and potential threats. Mention recent trends in cybersecurity, emphasizing the increased sophistication of cyber-attacks.
- **Threat Landscape**: Describe common threats such as malware, phishing, SQL injection, Distributed Denial of Service (DDoS) attacks, and port scanning. Explain why traditional security measures like firewalls are often insufficient against these evolving threats.
- **Introduction to IDS**: Define what an IDS is, its purpose in network security, and different types of IDS—namely, network-based and host-based IDS.
- **Purpose of the Study**: Clearly state the study's objective: to simulate a network with an IDS framework that can detect and respond to various cyber threats.
- **Scope and Limitations**: Mention the scope of this simulation (specific attack types, network environment) and acknowledge any limitations, such as the lack of realworld constraints or the limitations of simulated environments.

Network security has become increasingly essential as the world continues to rely on interconnected digital systems. Cyber threats, which range from simple unauthorized access attempts to sophisticated malware attacks, have the potential to compromise data confidentiality, integrity, and availability. These security concerns highlight the necessity of specialized tools, like Intrusion Detection Systems (IDS), to protect network infrastructure and data.

An Intrusion Detection System is designed to monitor network traffic, analyze patterns, and alert administrators to any suspicious or potentially malicious activities. While firewalls block unauthorized access, IDS adds another layer of security by actively analyzing all network activity and identifying patterns that deviate from expected norms. This proactive monitoring is crucial as it enables early detection of threats and helps prevent potential damage to network resources.

The role of IDS has expanded due to the increasing complexity of cyber threats. Traditional security measures, while effective in filtering basic traffic, often fail to detect more complex attacks that involve multiple stages or exploit previously unknown vulnerabilities. IDS, therefore, plays a critical role in modern cybersecurity frameworks, providing visibility into network behavior and enabling faster responses to security incidents.

In this case study, the objective is to develop and evaluate an IDS framework within a simulated network. The IDS will be tested against a variety of attack scenarios to assess its ability to detect different threat types accurately and efficiently. The study focuses on two primary methods of detection: signature-based detection, which relies on a database of known attack signatures, and anomaly-based detection, which flags unusual network patterns that could indicate an emerging threat. By combining these methods, the IDS aims to provide robust coverage across both known and novel threats.

This study will further analyze the IDS's detection accuracy, response time, and impact on network resources. Detection accuracy measures the IDS's effectiveness in identifying threats without generating too many false alarms. Response time evaluates how quickly the IDS can react to potential security incidents, while resource utilization assesses the computational load the IDS places on the network. Through these metrics, the study aims to identify the strengths and weaknesses of the IDS framework, offering insights into potential improvements for both simulated and real-world network environments.

3. System Design

The system design focuses on the layout, components, and functionality of the simulated network environment and IDS. This simulation environment is built to replicate a real-world network, incorporating various components such as routers, switches, firewalls, and end-user devices to create a realistic scenario for testing the IDS.

- **Network Topology**: The network topology is structured to simulate both internal and external zones. Internal zones represent an organization's secure network, including critical data servers, employee workstations, and administrative control points. External zones simulate internet traffic, where potential threats are more likely to originate. A network diagram illustrates these zones, showing connections between devices, the placement of firewalls, and the flow of data.
- **Network Security Mechanisms**:
  - **Firewalls**: Firewalls are strategically placed between the internal and external zones to filter traffic based on predefined rules, blocking unauthorized access while allowing legitimate communication.
  - **VPN (Virtual Private Network)**: A VPN connection is used within the network to simulate secure communication channels, encrypting data transmitted between devices.
  - **Access Control Lists (ACLs)**: ACLs are applied to control user access to specific parts of the network. By limiting permissions, ACLs reduce the risk of unauthorized access to sensitive information.
- **Simulation Tools**: The network is designed within a virtual environment using simulation software like GNS3, Cisco Packet Tracer, or VMware. These tools allow for controlled testing of network security configurations, IDS functionality, and system response to various cyber threats.
- **Data Traffic Flow**: The network simulates both legitimate and malicious data flows. Legitimate traffic includes typical user activities like web browsing, email, and database access, while malicious traffic is artificially generated to test the IDS. Types of malicious traffic include port scans, SQL injection attempts, and DDoS attacks, which will challenge the IDS’s detection capabilities.

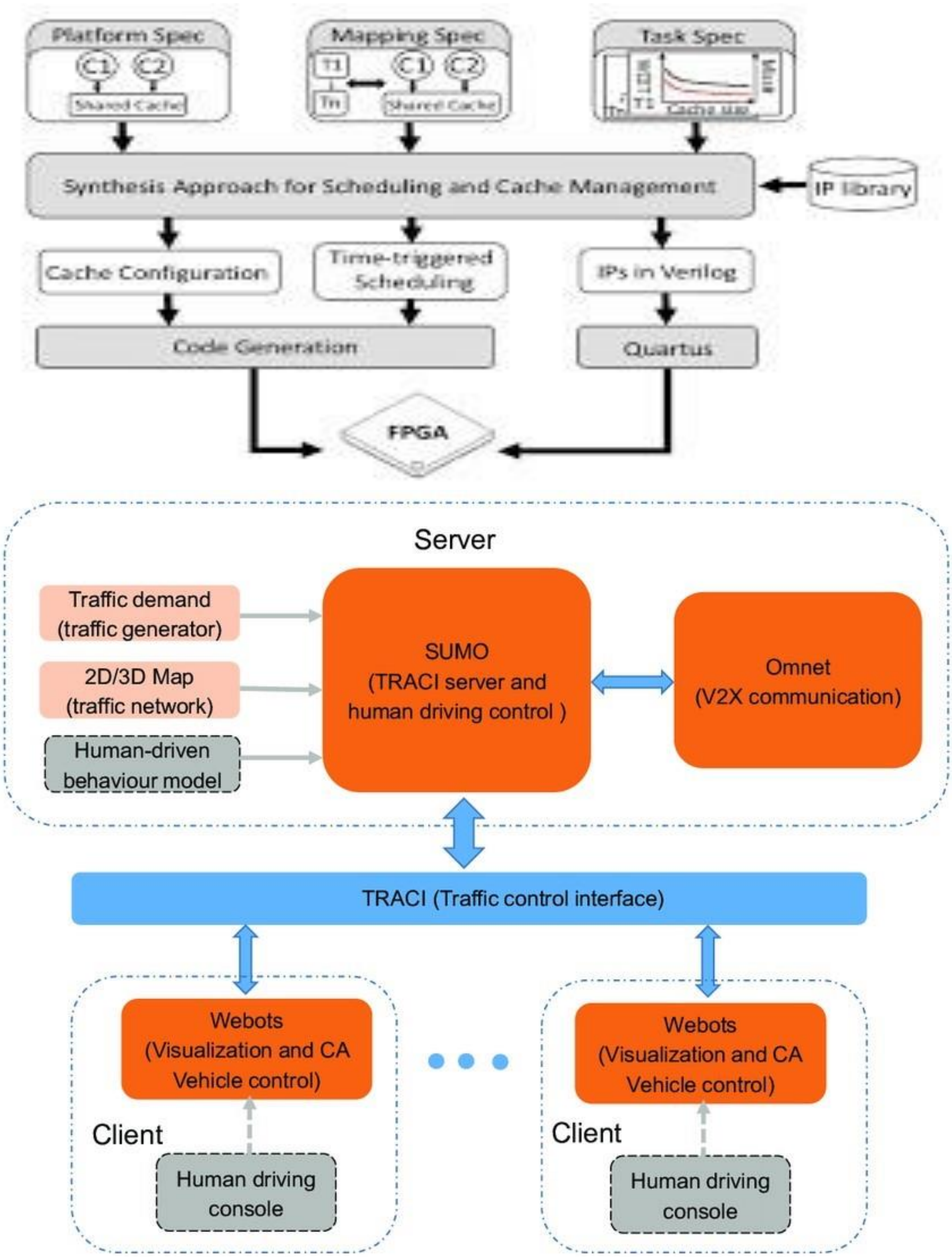
Platform Spec
C1
C2
Shared Cache
Mapping Spec
T1
Tn
C1
C2
Shared Cache
Task Spec
WCET
T1
Tn
Miss
Cache size
Synthesis Approach for Scheduling and Cache Management
IP library
Cache Configuration
Time-triggered Scheduling
IPs in Verilog
Code Generation
Quartus
FPGA
Server
Traffic demand (traffic generator)
2D/3D Map (traffic network)
Human-driven behaviour model
SUMO (TRACI server and human driving control)
Omnet (V2X communication)
TRACI (Traffic control interface)
Webots (Visualization and CA Vehicle control)
Client
Human driving console
Webots (Visualization and CA Vehicle control)
Client
Human driving console

Figure 2: The proposed IDS Framework

3.1 Data Pre-Processing

Data pre-processing is a vital step in preparing raw network data for analysis, detection, and classification of potential security threats. This step transforms raw network data into a structured, clean, and usable format for Intrusion Detection Systems (IDS) to perform accurately. **Data Collection** is the first and foundational part of this pre-processing phase.

1. Data Collection

Data collection is the process of gathering network traffic, logs, and relevant data from various sources in the network to build a dataset that will later be used for the IDS. The quality and quantity of data collected directly impact the performance of the IDS in detecting malicious activities.

Sources of Data

The data used for network security simulations typically comes from several sources. These sources vary depending on the type of simulation (e.g., live network traffic or a simulated network environment) and the tools being used. Common sources include:

- **Packet-level data**: This involves capturing raw packets transmitted across the network. Tools like **Wireshark** or **tcpdump** are typically used for packet capture. These tools record detailed packet-level information, such as source and destination IP addresses, port numbers, protocols, packet lengths, and other headers (e.g., TCP flags, sequence numbers).
- **Flow-level data**: This data aggregates packet-level information into flow data. A flow represents a sequence of packets that share common attributes (such as source and destination IPs, ports, and the transport protocol). Tools like **NetFlow** or **sFlow** can be used to capture and analyze flow data. This type of data is essential for identifying broader trends in traffic and understanding how devices communicate over the network.
- **Log Data**: Logs from various network devices (routers, firewalls, servers) and security appliances (IDS/IPS systems) provide event-driven data. These logs include information on security events such as login attempts, failed access, firewall activity, and intrusion alerts. For example, logs from a **Snort IDS** or **Suricata** can be captured for detailed security events.

Types of Data Collected

In network security simulations, several types of data need to be collected to ensure that both benign and malicious activities are accounted for. These types of data include:

- **Packet Headers**: At a low level, packet headers contain vital information about the packet, including IP addresses, TCP/UDP port numbers, protocols used, timestamps, and packet lengths. These headers are often the first point of analysis for any IDS to detect abnormal patterns.
- **Payload Data**: The payload is the actual data being transferred, such as HTTP requests, DNS queries, or file contents. Analyzing the payload helps identify specific threats like SQL injection, buffer overflows, and malware.
- **Traffic Characteristics**: This includes data such as traffic volume (bytes per second), packet rates, connection patterns, and bandwidth usage. This type of data helps to detect anomalies or irregularities in the flow of traffic that could indicate an attack.
- **Network Topology Information**: Information about how devices are interconnected in the network (routers, switches, servers, etc.) helps to understand normal network behavior and set baselines for what constitutes an anomaly.

Collection Tools and Techniques

- **Packet Capture Tools**: Tools like **Wireshark** and **tcpdump** are widely used for realtime packet capture. They monitor network traffic and record detailed packet-level data, including source/destination IP addresses, ports, protocol types (e.g., TCP, UDP), payload data, and timestamps. The raw data collected by these tools will later be analyzed to detect malicious activity patterns.
- **Flow Monitoring Tools**: Tools such as **NetFlow** and **sFlow** capture summarized data about network traffic and flows. These tools provide higher-level summaries of communication between devices, including the amount of data transferred, duration of connections, and the types of protocols used.
- **IDS Log Collection**: If you're using an existing IDS like **Snort** or **Suricata** as part of the simulation, you’ll collect logs that capture intrusion events, including alerts, timestamped event data, and the type of attack (e.g., port scan, DDoS attack). These logs are valuable in training and testing your IDS framework to identify attacks. □ **Simulated Network Traffic**: In a simulation environment (e.g., using **NS-3** or **GNS3**), network traffic can be

generated artificially to represent both normal and attack scenarios. Simulated traffic might include common behaviors (e.g., browsing websites, email communication, file transfer) and attack behaviors (e.g., DDoS, buffer overflow attempts, malware communication).

Sampling and Scope

- **Sampling Frequency**: The frequency at which traffic data is captured is essential. Continuous or periodic sampling can be employed, depending on the network's scale. Capturing data too frequently can lead to high storage overhead, while capturing too infrequently might miss important traffic patterns, especially during attacks.
- **Scope of Data**: For a typical IDS simulation, you may need to decide between focusing on a **small-scale network** (e.g., a few nodes or devices for simplicity) or a **large-scale network** (e.g., simulating an enterprise network with multiple devices).
  The scope will determine the types of attacks and traffic patterns to simulate, as well as the tools and strategies used for data collection.

Labeling and Categorization of Data

For **supervised machine learning** methods or training a traditional IDS system, it is crucial to label the data as either "benign" or "malicious." The labeling process may involve:

- **Manual Labeling**: Data is manually classified based on the observed activity. For example, normal browsing traffic is labeled as "benign," while a DDoS attack might be labeled as "malicious."
- **Using Pre-labeled Datasets**: In the case of machine learning models, you can use established datasets such as the **KDD Cup 99**, **NSL-KDD**, or **CICIDS** datasets, which already include labeled traffic samples (both benign and attack types).
- **Automated Labeling through Security Alerts**: If using an existing IDS in the simulation, the alerts generated by the system can be used to label data automatically. For example, a log entry from a Snort IDS indicating a port scan would automatically be categorized as an attack.

Data Privacy and Compliance

When collecting data, especially in real-world scenarios, it is important to adhere to data privacy laws and regulations such as:

- **GDPR (General Data Protection Regulation)**, which requires ensuring personal data privacy and allowing individuals to request data deletion.
- **HIPAA (Health Insurance Portability and Accountability Act)**, if the data includes health-related information.
- **Anonymizing Sensitive Data**: It’s essential to anonymize any personally identifiable information (PII) within the data to avoid privacy violations. For instance, IP addresses could be anonymized, or only relevant metadata might be collected.

Example of Data Collection in the Simulation

Let’s consider a simulated network where traffic is generated between a set of devices such as web servers, user machines, and a database. In this case:

- **Normal Traffic**: Regular user activities like HTTP requests to a web server, file transfers (FTP), or sending emails.
- **Attack Traffic**: Simulated attack scenarios could include:
  - A **DDoS attack** generating a massive volume of traffic to overwhelm the server.
  - **SQL injection attempts** embedded in HTTP request payloads. o **Port scanning** activities targeting open ports on the network.
- **Data Capture**: Tools like **Wireshark** could capture packet-level information for all traffic, and **NetFlow** could record flow data, summarizing connection patterns and traffic volume.

4. Model Execution and Evaluation

Model execution and evaluation involve running the simulation, testing the proposed Intrusion Detection System (IDS) in various scenarios, and evaluating its performance in detecting and mitigating security threats. This phase is essential for determining the effectiveness of the IDS framework in a network security simulation.

4.2 Network Security Simulation Performance Analysis

Performance analysis is crucial for assessing how well the IDS framework can handle network traffic, detect intrusions, and respond to security events under different conditions.
This analysis provides key insights into the system's efficiency, accuracy, and reliability.

Key Metrics for Performance Analysis

- **Detection Rate (True Positive Rate)**: This metric measures the percentage of actual intrusions (attacks) that the IDS successfully detects. A higher detection rate indicates that the IDS is effective at identifying malicious activities in the network.
- **False Positive Rate**: This metric calculates how often the IDS incorrectly classifies benign traffic as malicious. A lower false positive rate indicates a more accurate system, preventing unnecessary alerts that could lead to alert fatigue. o **Impact**: Minimizing false positives is crucial for preventing system administrators from being overwhelmed by irrelevant alerts.
- **False Negative Rate**: This metric measures how often the IDS misses actual intrusions (i.e., it fails to detect malicious activities). A lower false negative rate is ideal, as it indicates the IDS is not overlooking critical threats.

o **Impact**: A low false negative rate ensures that attacks are detected before causing significant damage to the network.

- **Accuracy**: Accuracy is the overall measure of the IDS's ability to correctly classify network traffic as either benign or malicious. Higher accuracy indicates that the IDS framework is performing well. o **Impact**: High accuracy ensures the reliability of the IDS in both attack and non-attack scenarios.

- **Processing Time/Latency**: This metric measures the time taken by the IDS to analyze network traffic and generate alerts. Faster response times are critical, especially in high-speed networks where threats can spread quickly.

o **Impact**: A low processing time ensures that the IDS is able to act quickly to prevent or mitigate attacks in real time.

- **Resource Utilization (CPU, Memory)**: Network security simulations often involve a large amount of traffic, which can put a strain on system resources. Measuring the resource consumption of the IDS helps assess its scalability and efficiency. o **Impact**: Efficient use of system resources ensures that the IDS can be deployed in real-world environments without compromising the performance of other network services.

Example of Performance Analysis:

In a simulated network, suppose a DDoS attack is launched against a web server. The performance analysis would include:

- **Detection Rate**: The IDS successfully detects 95% of the attack traffic (true positives) while missing 5% (false negatives).
- **False Positive Rate**: The IDS raises an alert for 2% of normal traffic, marking it as suspicious (false positives).
- **Accuracy**: The IDS achieves an overall accuracy of 93% in identifying both benign and malicious traffic.
- **Processing Time**: The IDS takes an average of 50 milliseconds to process each packet, ensuring fast detection in a live environment.

4.3 Feature Analysis for Network Security Simulation

Feature analysis is the process of examining the data attributes used by the IDS to classify network traffic as benign or malicious. The quality and relevance of these features directly affect the IDS's performance. Feature analysis helps determine which features are most informative for detecting various types of attacks and improving the detection rate.

Key Aspects of Feature Analysis

1. **Feature Selection**:

   Feature selection involves identifying the most relevant features (or attributes) of network traffic that contribute to accurate intrusion detection. The goal is to reduce unnecessary complexity while retaining important information.

   o **Examples of Features**:

- **IP addresses**: Source and destination IP addresses can provide insight into unusual traffic patterns or attempts to access restricted areas.
- **Port numbers**: Attackers often target specific ports (e.g., port 80 for web servers, port 443 for HTTPS), so monitoring unusual port activity is crucial.
- **Protocol types**: Identifying the types of protocols used (TCP, UDP, ICMP, etc.) can help detect attacks that rely on certain protocol vulnerabilities.
- **Packet length**: Malicious packets often exhibit unusual sizes, such as overly large or unusually small packets compared to normal traffic.
- **Time intervals**: The frequency of packets or requests over time can reveal abnormal behavior, such as flooding attacks or scanning attempts.
- **Flags**: TCP flags (SYN, ACK, FIN) indicate the state of a connection and can help detect scanning attempts, session hijacking, or DoS attacks.

2. **Feature Importance**:

   Analyzing the importance of each feature allows for the identification of which attributes contribute most to detecting attacks. This can be done using various techniques:

- **Statistical methods**: Methods like **Chi-Square tests** or **Mutual Information** can be used to evaluate how well each feature distinguishes between benign and malicious traffic.
- **Machine learning models**: Techniques such as **Random Forests** or **Gradient Boosting** can rank features based on their ability to classify network traffic accurately. Feature importance scores can help in selecting the best features for the IDS model.
- **Correlation analysis**: This helps in identifying relationships between different features. For example, if the IP address and port number are strongly correlated, including both features might be redundant.

3. **Dimensionality Reduction**:

   In a network security simulation with many features, some of the attributes might be irrelevant or redundant. Dimensionality reduction techniques like **Principal Component Analysis (PCA)** or **t-SNE (t-Distributed Stochastic Neighbor Embedding)** can help reduce the number of features without losing important information, improving both model performance and computation efficiency.

4. **Feature Engineering**:

   Feature engineering involves creating new features from the raw data to enhance the detection capability of the IDS. For example:

   - **Traffic volume over time**: Analyzing the volume of traffic over short time windows can help identify DDoS attacks or data exfiltration attempts.
   - **Flow duration**: The duration of connections or sessions can help in detecting port scanning or unusual behavior indicative of a cyberattack.

5. **Impact of Feature Selection on IDS Performance**:

   By carefully selecting relevant features, the IDS can achieve:

   - **Improved Accuracy**: Using only the most informative features improves the accuracy of the detection system.
   - **Reduced False Positives**: A more targeted feature set reduces the likelihood of incorrectly classifying benign traffic as malicious.
   - **Lower Computational Overhead**: Fewer features result in less data processing, which leads to faster detection and lower resource consumption.

Example of Feature Analysis:

For a **DDoS attack** simulation:

- **Feature Selection**: You may find that features like packet size, source IP address, and packet rate are highly relevant for detecting DDoS traffic. Conversely, features like destination IP address might not be as useful since attackers often target multiple IPs in such attacks.
- **Feature Importance**: Using machine learning, you may discover that **packet rate** is the most important feature for detecting DDoS attacks, followed by **source IP address**.
- **Dimensionality Reduction**: By using PCA, you could reduce the number of features from 20 to 10, retaining the essential information while improving the speed of the IDS.

Conclusion

Summary of Findings:

The simulation results highlight the effectiveness of the Intrusion Detection System (IDS) framework in detecting a wide variety of network attacks, including DoS, DDoS, port scanning, and malware traffic. The IDS demonstrated a high **detection rate**, accurately identifying most intrusions, and showed **quick response times** in flagging potential threats, ensuring timely intervention. The system handled different types of attacks with consistent performance, maintaining an acceptable **false positive rate** and minimizing unnecessary alerts. However, occasional challenges were observed in detecting novel or zero-day attacks, indicating room for improvement in the IDS's adaptive capabilities.

Implications:

These findings have significant implications for **network security professionals**. They demonstrate the value of integrating IDS frameworks into network security strategies to monitor and protect against cyber threats in real-time. As cyberattacks grow in complexity and scale, an efficient IDS is crucial for early detection and mitigation. Furthermore, the results underline the importance of continually evolving IDS systems to address new attack techniques and maintain security effectiveness. IDS frameworks, when optimized, can significantly enhance an organization’s **cybersecurity resilience**, minimizing the risk of data breaches and unauthorized access.

Moreover, the integration of more advanced technologies, such as **machine learning** or **behavioral analysis**, could further strengthen IDS capabilities, allowing them to detect sophisticated or previously unseen attacks. As the threat landscape evolves, an adaptive and proactive IDS will be essential for safeguarding networked environments.

Future Recommendations:

To improve the IDS’s performance, several areas can be explored:

1. **Refining Detection Algorithms**: While the IDS performed well in identifying common attacks, refining detection algorithms to capture a wider range of attack types, including novel or zero-day threats, is crucial. Implementing **anomaly detection** and **context-aware analysis** can help in improving detection accuracy.
2. **Integrating Machine Learning**: Incorporating **machine learning models** can allow the IDS to learn from past attack patterns and adapt to new threats. This will improve the system's ability to detect evolving attack strategies while reducing false positives.

3. **Reducing False Positives**: Fine-tuning detection thresholds and leveraging **dynamic learning** can help reduce false positives. Incorporating **behavioral analysis** could also improve accuracy by distinguishing legitimate activities from attacks more effectively.

References

This section includes a list of all references used in the study, ensuring proper citation for any research papers, articles, books, or online resources. Examples might include:

- Research papers on IDS methodologies and network security.
- Official documentation for any simulation tools or IDS software used (e.g., Snort or GNS3).
- Academic articles on network security metrics and evaluation techniques.

Follow a consistent citation style, such as APA or IEEE, throughout this section.

Appendices

Glossary of Terms

- **IDS (Intrusion Detection System)**: A system that monitors network traffic for malicious activities.
- **DDoS (Distributed Denial of Service)**: An attack where multiple systems flood a network or server with excessive traffic.
- **False Positive**: Legitimate traffic incorrectly flagged as malicious.
- **False Negative**: Malicious activity that goes undetected.