A PBL-II (AIM2270) Synopsis on

Network Intrusion Detection System

Submitted to Manipal University Jaipur

Towards the partial fulfillment for the Award of the Degree of

B. Tech Computer Science and Engineering (Artificial Intelligence and Machine Learning

In Computers Science and Engineering (AIML)

BY

Ashwarya Pradhan
23FE10CAI00268

Kasmya Bhatia
23FE10CAI00010

MANIPAL UNIVERSITY
JAIPUR

INSPIRED BY LIFE

Under the guidance of

Dr.Gautam Kumar

Gautam Kr.

Department of Artificial Intelligence and Machine Learning

Manipal University Jaipur

Jaipur, Rajasthan

INDEX

Introduction to the Problem

Cyber threats are more advanced than ever, and organizations are struggling to keep up. Hackers exploit vulnerabilities to launch attacks like malware infections, data breaches, and Denial-of-Service (DoS) attacks, often slipping past traditional security measures. Without a proper detection system, these threats can cause:

- Financial losses – Cyber Attacks can result in theft, ransom demands, or costly recovery efforts.
- Operational disruptions – Downtime from attacks can halt business processes and impact productivity.
- Data breaches – Sensitive information can be stolen, leading to legal and reputational damage.

While tools like firewalls and antivirus software help block known threats, they are not enough to detect real-time network intrusions or sophisticated attack patterns. This is where a Network Intrusion Detection System (NIDS) becomes essential:

- Monitors network traffic in real time to spot suspicious activities.
- Identifies cyber threats early before they escalate into major security incidents.
- Provides alerts to security teams so they can respond quickly and minimize damage.

With the increasing complexity of cyber threats, organizations need a smarter, faster, and more proactive way to protect their networks—one that goes beyond traditional defenses. A well-implemented NIDS helps bridge this gap, offering continuous monitoring and enhanced security.

Literature Survey

A comprehensive literature survey of Network Intrusion Detection Systems (NIDS) reveals the variety of existing software, hardware, and methods for detecting network-based threats. These systems are critical for cybersecurity as they help monitor network traffic, identify unauthorized activities, and provide proactive measures against cyber-attacks. Below is a summary of recent NIDS solutions, along with their pros and cons:

1. Snort (Open-source Intrusion Detection System)

Snort is a widely-used open-source IDS that analyzes network traffic for signs of malicious behavior. It uses signature-based detection, which matches packets to known attack signatures.

Pros:

- Highly customizable with rule sets for tailored detection.

- Open-source, allowing full transparency and community-driven development.

- Well-supported with an active community and extensive documentation.

- Integrates with SIEM systems for enhanced monitoring.

Cons:

- High false positive rate due to reliance on static signatures.

- Does not detect new, unknown attacks without updated signatures.

- Performance can degrade under high traffic loads without proper optimization.

2. Suricata (High-performance Network IDS/IPS)

Suricata is another open-source IDS that provides high-performance packet capture and analysis. It supports signature-based, anomaly-based, and hybrid detection methods.

Pros:

- High throughput and low latency for large networks.

- Multi-threaded design for better scalability and efficiency.

- Built-in intrusion prevention capabilities (IPS) that block detected attacks in real-time.

- Supports IPv6, which is essential for modern networks.

Cons:

- Complex setup compared to simpler IDS like Snort.

- Requires significant system resources for high-speed traffic analysis.

- False positives still present, especially in high-traffic environments.

3. Zeek (formerly Bro) (Network Monitoring Framework)

Zeek is a powerful network monitoring tool used for real-time traffic analysis. It is not strictly an IDS but provides anomaly-based detection by analyzing various traffic protocols.

Pros:

- Excellent at anomaly detection and identifying sophisticated attacks (e.g., zero-day).

- Scalable for both small networks and large enterprises.

- Provides detailed logging of network activities for forensic analysis.

- Can integrate with other security tools and services like SIEMs.

Cons:

- Not fully automated in threat detection and mitigation.

- Requires advanced configuration and understanding of network protocols.

- High complexity for newcomers and lacks a user-friendly interface.

4. Snort + Suricata Hybrid Solution

This solution combines Snort's signature-based detection with Suricata’s performance and multi-method detection capabilities to create a more comprehensive and efficient NIDS.

Pros:

- Utilizes both signature-based and anomaly-based detection, providing a robust solution against known and unknown threats.

- Scalable and capable of handling high traffic loads.

- Can be used in a distributed architecture for larger networks.

Cons:

- High resource consumption, which may affect performance in resource-constrained environments.

- More complex to manage and maintain, requiring skilled personnel.

- Increased risk of false positives due to combining multiple detection methods.

5. Cisco Firepower (Commercial Network Security Solution)

Cisco Firepower is a commercial solution that provides integrated intrusion detection, intrusion prevention, and advanced malware protection for enterprise networks.

Pros:

- Comprehensive protection: combines multiple security features like IPS, URL filtering, and anti-malware.

- Integrated with Cisco network infrastructure, providing seamless protection across the network.

- High accuracy in detecting threats due to advanced machine learning algorithms.

- Provides detailed visualization of traffic and attack patterns.

Cons:

- Expensive, making it less accessible for small businesses.

- Vendor lock-in with Cisco’s hardware and software ecosystem.

- Complex configuration, requiring specialized knowledge for optimal deployment.

6. AWS GuardDuty (Cloud-based Threat Detection)

AWS GuardDuty is a cloud-native IDS for AWS environments, which uses machine learning and anomaly detection to monitor network activity and identify malicious behavior.

Pros:

- Scalable, designed for cloud environments, making it ideal for AWS-based infrastructures.

- Automated threat detection using machine learning models, minimizing manual rule configuration.

- Seamlessly integrates with other AWS services like CloudWatch and Lambda for automated responses.

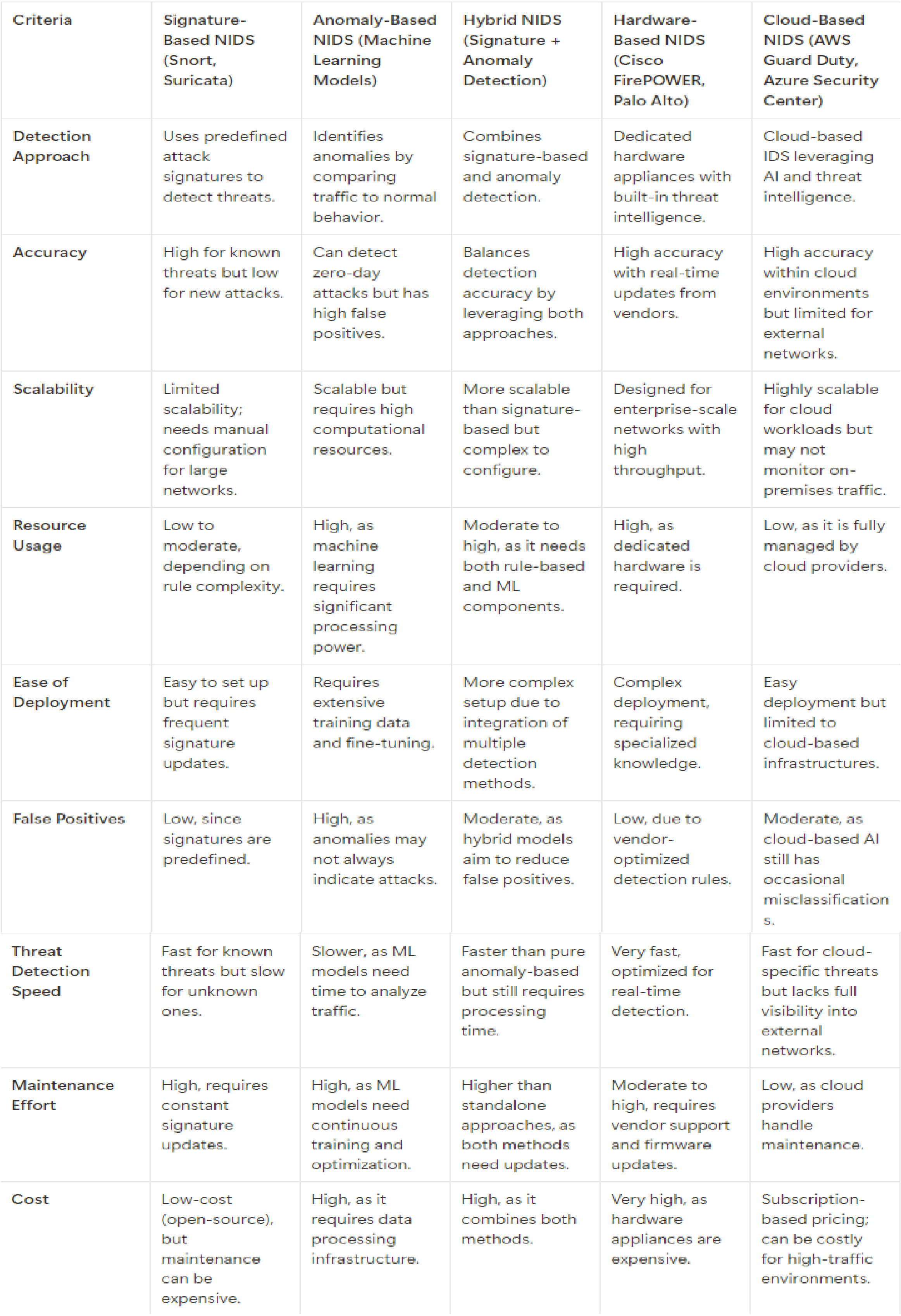
Cons:

- Limited to AWS infrastructure, cannot be used for on-premises or multi-cloud environments.

- Cost-based on usage, which may lead to unexpected billing for high-volume environments.

- Less effective for detecting network attacks outside of AWS resources.

Comparative Study



Criteria	Signature-Based NIDS (Snort, Suricata)	Anomaly-Based NIDS (Machine Learning Models)	Hybrid NIDS (Signature + Anomaly Detection)	Hardware-Based NIDS (Cisco FirePOWER, Palo Alto)	Cloud-Based NIDS (AWS Guard Duty, Azure Security Center)
Detection Approach	Uses predefined attack signatures to detect threats.	Identifies anomalies by comparing traffic to normal behavior.	Combines signature-based and anomaly detection.	Dedicated hardware appliances with built-in threat intelligence.	Cloud-based IDS leveraging AI and threat intelligence.
Accuracy	High for known threats but low for new attacks.	Can detect zero-day attacks but has high false positives.	Balances detection accuracy by leveraging both approaches.	High accuracy with real-time updates from vendors.	High accuracy within cloud environments but limited for external networks.
Scalability	Limited scalability; needs manual configuration for large networks.	Scalable but requires high computational resources.	More scalable than signature-based but complex to configure.	Designed for enterprise-scale networks with high throughput.	Highly scalable for cloud workloads but may not monitor on-premises traffic.
Resource Usage	Low to moderate, depending on rule complexity.	High, as machine learning requires significant processing power.	Moderate to high, as it needs both rule-based and ML components.	High, as dedicated hardware is required.	Low, as it is fully managed by cloud providers.
Ease of Deployment	Easy to set up but requires frequent signature updates.	Requires extensive training data and fine-tuning.	More complex setup due to integration of multiple detection methods.	Complex deployment, requiring specialized knowledge.	Easy deployment but limited to cloud-based infrastructures.
False Positives	Low, since signatures are predefined.	High, as anomalies may not always indicate attacks.	Moderate, as hybrid models aim to reduce false positives.	Low, due to vendor-optimized detection rules.	Moderate, as cloud-based AI still has occasional misclassifications.
Threat Detection Speed	Fast for known threats but slow for unknown ones.	Slower, as ML models need time to analyze traffic.	Faster than pure anomaly-based but still requires processing time.	Very fast, optimized for real-time detection.	Fast for cloud-specific threats but lacks full visibility into external networks.
Maintenance Effort	High, requires constant signature updates.	High, as ML models need continuous training and optimization.	Higher than standalone approaches, as both methods need updates.	Moderate to high, requires vendor support and firmware updates.	Low, as cloud providers handle maintenance.
Cost	Low-cost (open-source), but maintenance can be expensive.	High, as it requires data processing infrastructure.	High, as it combines both methods.	Very high, as hardware appliances are expensive.	Subscription-based pricing; can be costly for high-traffic environments.

Objective

The objectives of the Network Intrusion Detection System (NIDS) are as follows:

1. **Strengthen Network Security**
    - Continuously monitor network traffic in real-time to detect any suspicious or unauthorized activities.
    - Identify and mitigate cyber threats such as malware, hacking attempts, and Denial-of-Service (DoS) attacks before they can cause significant damage.

2. **Improve Accuracy in Threat Detection**
    - Employ a combination of signature-based and anomaly-based detection techniques to identify both known and novel threats.
    - Minimize false positives by incorporating Artificial Intelligence (AI) and machine learning to enhance threat identification and classification.

3. **Provide Instant Alerts and Quick Response**
    - Generate real-time notifications to alert security teams whenever a potential threat is detected.
    - Collaborate with existing security infrastructure to enable immediate response and prevent the escalation or spread of attacks.

4. **Ensure Flexibility and Seamless Integration**
    - Ensure scalability to accommodate the needs of organizations of varying sizes, from small businesses to large enterprises, without compromising performance.
    - Facilitate smooth integration with firewalls, Security Information and Event Management (SIEM) systems, and cloud security tools to provide a comprehensive cybersecurity solution.

Planning of Work

Step 1: Problem Definition & Requirement Analysis

Objective: Understand the security challenges and define the requirements for building an efficient NIDS.

Key Features:

- Identify common cyber threats (DoS, malware, unauthorized access, etc.).
- Define system goals: real-time monitoring, accurate threat detection, scalability, and ease of integration.
- Determine project scope (on-premises, cloud-based, hybrid).
- Establish the types of threats to detect (signature-based, anomaly-based, hybrid).
- Identify necessary compliance and security regulations (GDPR, ISO 27001, etc.).

Step 2: Research & Literature Survey

Objective: Study existing NIDS solutions, compare detection techniques, and find areas for improvement.

Key Features:

- Research open-source (Snort, Suricata) and commercial (Cisco FirePOWER, AWS GuardDuty) NIDS solutions.
- Analyze various threat detection techniques (signature-based, machine learning, behavioral analysis).
- Identify pros and cons of existing methodologies.
- Understand system limitations (high false positives, processing latency, etc.).
- Define what improvements can be made in the new NIDS system.

Step 3: System Design & Architecture Development

Objective: Develop a high-level design of the NIDS, including components, data flow, and technology stack.

Key Features:

- Design System Architecture: Define how network traffic will be collected, analyzed, and logged.
- Select Detection Methods: Choose signature-based, anomaly-based, or hybrid detection.
- Develop Data Flow Model:
    - Packet capture from network interfaces.
    - Traffic preprocessing and filtering.
    - Threat detection module for identifying malicious activities.
    - Alert generation and logging system.
- Technology Stack Selection:
    - Programming language: Python, C++ (for packet capture and analysis).
    - Databases: SQL or NoSQL for log storage.
    - Tools: Wireshark, Scapy for traffic analysis, Elasticsearch & Kibana for visualization.
- Security Considerations: Implement encryption for logs, secure access control, and threat intelligence integration.

Step 4: Implementation & Development

Objective: Build core system modules and integrate them into a working NIDS.

Key Features:

- Traffic Monitoring Module:
    - Capture live network traffic using pcap libraries (libpcap, WinPcap).
    - Extract metadata: IP addresses, protocols, ports, packet size.
- Intrusion Detection Engine:
    - Implement signature-based detection (predefined attack patterns).

    - Develop anomaly-based detection (machine learning for unknown threats).
    - Optimize algorithms to minimize false positives and enhance detection speed.
- Real-Time Alerting System:
    - Generate alerts via email, SMS, or dashboard notifications when a threat is detected.
- Logging & Reporting:
    - Store logs in a structured format (JSON, database storage).
    - Provide exportable reports for cybersecurity teams.
- Optional Dashboard & Visualization:
    - Use Grafana/Kibana for real-time network monitoring.
    - Display threat trends, affected systems, and historical attack data.

Step 5: Testing & Evaluation

Objective: Ensure system accuracy, stability, and efficiency in real-world conditions.

Key Features:

- Unit Testing: Test individual modules (traffic monitoring, detection, alerting).
- Integration Testing: Ensure smooth interaction between components.
- Simulated Attack Scenarios:
    - Deploy tools like Metasploit, Kali Linux to simulate network attacks.
    - Evaluate detection accuracy for malware, DoS, brute force attacks, SQL injection.
- False Positive & False Negative Analysis:
    - Tune detection thresholds to minimize false alarms.
- Performance Testing:
    - Evaluate system response time for high traffic loads.
    - Optimize resource usage (CPU, memory).

Step 6: Deployment & Scalability Testing

Objective: Ensure the NIDS can operate efficiently in different environments and scale with increasing network traffic.

Key Features:

- On-Premises Deployment: Install on Linux-based servers or dedicated security appliances.
- Cloud Deployment: Deploy in AWS, Azure, or Google Cloud for cloud-based monitoring.
- Hybrid Implementation: Integrate both on-premises and cloud monitoring for extended security.
- Scalability Testing: Test with increased network load and optimize system response.
- Compatibility Testing: Ensure smooth integration with firewalls, SIEM systems, endpoint detection tools.

Step 7: Documentation & User Training

Objective: Provide proper documentation for system usage, troubleshooting, and maintenance.

Key Features:

- Create detailed user guides on system installation, configuration, and usage.
- Provide technical documentation for developers and cybersecurity teams.
- Conduct training sessions for IT teams on managing alerts and responding to threats.
- Develop FAQs and troubleshooting guides for common issues.

Step 8: Final Review & Future Enhancements

Objective: Assess system performance, gather feedback, and plan for future improvements.

Key Features:

- Performance Review: Analyze system logs and reports to evaluate effectiveness.
- User Feedback: Gather insights from administrators and security teams.
- Future Enhancements:
  - Implement AI-driven adaptive threat detection.
  - Add automated incident response capabilities.
  - Develop mobile app notifications for real-time alerts.
- Regular Updates & Maintenance: Ensure threat detection models and signature databases remain up-to-date.

Bibliography

- Scapy Documentation
  Title: Scapy: A Python Tool for Network Interaction
  Source: Scapy Documentation
  URL: https://scapy.readthedocs.io/en/latest/

- Snort
  Title: Snort: Lightweight Intrusion Detection for Networks
  Author: Roesch, M. (1999)
  Source: Snort
  URL: https://www.snort.org/

- Suricata
  Title: Suricata: High-Performance Network IDS, IPS, and Network Security Monitoring Engine
  Source: Suricata Documentation (OISF)
  URL: https://suricata.io/

- Kibana Documentation
  Title: Kibana: Data Visualization and Exploration Tool for Elasticsearch
  Source: Elastic Kibana Documentation
  URL: https://www.elastic.co/kibana/

- Elasticsearch
  Title: Elasticsearch: The Definitive Guide
  Authors: Gormley, C., & Tong, Z. (2015)
  Publisher: O'Reilly Media
  ISBN: 978-1449358546

- Wireshark
  Title: Wireshark User’s Guide
  Author: Combs, G. (2006)
  Source: Wireshark Documentation
  URL: https://www.wireshark.org/

- International Journal of Computer Applications
  Title: A Survey of Network Intrusion Detection Systems Using Machine Learning Techniques
  Authors: Ahmed, M., Mahmood, A. N., & Hu, J. (2016)
  Journal: *International Journal of Computer Applications*, 975, 8887
  DOI: 10.5120/ijca2016909962

- International Journal of Computer Applications
  Title: Comparative Study of Network Intrusion Detection with Machine Learning Algorithms
  Author: Rafique, M. (2020)
  Journal: *International Journal of Computer Applications*, 180(25), 12-16
  DOI: 10.5120/ijca2020920828

- Packt Publishing
  Title: Practical Guide to Intrusion Detection Systems
  Publisher: Packt Publishing (2019)
  ISBN: 978-1789804931

- Pearson Education
  Title: Network Security Essentials
  Author: Stallings, W. (2014)
  Publisher: Pearson Education
  ISBN: 978-0133354863