Project Proposal

Intrusion Detection and Prevention System (IDPS)

Group Members

Hamza Zaka Khan - 2024213

Muhammad Saad - 2024444

1. Problem Description

In modern computing environments, cyber-attacks are becoming increasingly sophisticated and frequent. Traditional defenses such as firewalls and antivirus software often fail to identify complex or zero-day attacks. An Intrusion Detection and Prevention System (IDPS) goes beyond simple detection—it not only monitors and identifies potential threats but also takes automated actions to prevent or mitigate them. The project aims to design and implement a simplified IDPS that can efficiently detect, analyze, and block malicious activities using data structures and algorithms studied in this course.

2. Project Aim

The primary objective of this project is to develop an Intrusion Detection and Prevention System capable of detecting and mitigating network-based attacks in real-time. The system will analyze incoming packets, recognize patterns of malicious behavior, and take preventive actions such as blocking IPs or alerting administrators. This project emphasizes algorithmic efficiency and data structure optimization to handle large-scale data quickly.

3. Key Data Structures

The IDPS will make use of multiple data structures for efficient operation:

- • Hash Tables — for storing and quickly accessing IP addresses and attack signatures.
- • Queues — for real-time packet buffering and sequential processing.
- • Trees (Binary Search Tree / Trie) — for efficient pattern and signature matching.
- • Graphs — to represent network connections and detect abnormal traffic routes.
- • Heaps / Priority Queues — to prioritize alerts based on severity or frequency.
- • Arrays / Vectors — for storing feature sets and maintaining packet data sequences.

4. Algorithms

The following algorithms will be integrated into the IDPS for detection and prevention:

- • Pattern Matching Algorithms — such as KMP or Rabin-Karp for detecting known attack signatures.
- • Anomaly Detection Algorithm — based on statistical thresholds or deviation from normal traffic behavior.
- • Sorting and Searching Algorithms — for efficiently managing network packet data.
- • Graph Traversal (BFS/DFS) — for detecting unusual traffic flows between network nodes.
- • Rule-based Prevention Algorithm — to automatically block or quarantine detected threats.

5. Data Flow: Input, Processing, and Output

Input: The system will receive simulated network data, including packet headers, IPs, ports, and protocols.

Processing: The IDPS will analyze each packet using pattern matching and anomaly detection algorithms. If a packet is identified as malicious, the prevention module will execute appropriate actions such as blocking or alerting.

Output: The system will display detailed logs, alerts, and prevention actions (e.g., 'IP Blocked', 'Threat Neutralized'), including timestamps and severity ratings.

6. Integration of Data Structures and Algorithms Concepts

This project integrates multiple DSA concepts from CS221, including:
• Efficient use of hash-based indexing for rapid threat lookup.
• Use of queues and priority queues for real-time traffic management.
• Implementation of tree and graph structures for hierarchical and network-based analysis.
• Application of search and pattern-matching algorithms to identify malicious traffic.
• Optimization of time complexity for real-time detection and prevention tasks.

7. Expected Outcome

The resulting IDPS prototype will be capable of both detecting and preventing potential network intrusions. It will demonstrate how data structures and algorithms can be effectively applied in cybersecurity systems to enhance performance, accuracy, and scalability.

8. Data Repository

Link : https://github.com/saadii403/ideal-funicular/tree/main