Title: Network IDS

Name: Khushi Ganesh Suvarna

Intern ID: 121

Executive Summary

This project presents a prototype of a **Network Intrusion Detection System (NIDS)** developed in Python. The tool can analyze both **live traffic** and **stored packet captures (PCAPs)**. It is designed to raise alerts for common suspicious patterns such as:

- Excessive ICMP echo requests (ping floods)
- High-rate TCP SYN attempts and SYN floods
- Scanning activity across multiple ports
- Large volumes of half-open connections

The IDS was tested in controlled conditions and successfully flagged abnormal traffic while ignoring normal browsing. Although it cannot compete with full-featured tools like Snort, it demonstrates the **core principles of traffic monitoring and anomaly detection**.

---

2. Objectives

The purpose of this PoC was to:

1. Implement a simple IDS prototype in Python using Scapy.
2. Detect a small set of common reconnaissance and flooding techniques.
3. Validate detection with both benign and malicious network traffic.
4. Explore potential weaknesses such as false positives.
5. Produce a short report for documentation and future reference.

---

3. Approach

3.1 Tools and Environment

- **Programming Language:** Python 3
- **Library:** Scapy for packet capture and parsing

- **System:** Linux workstation with administrative privileges
- **Testing Utilities:** Ping, Nmap, and sample PCAP files

3.2 Detection Rules

- **ICMP Flood:** More than 20 ICMP requests from a host in 10 seconds triggers an alert.
- **SYN Flood:** More than 30 SYNs in 10 seconds from a host raises a warning.
- **Port Scan:** Contacting 10 or more different ports in 10 seconds indicates a scan.
- **Half-Open Connections:** If over 70% of SYNs remain unanswered by ACKs, the connection is considered suspicious.

3.3 System Workflow

1. **Traffic Capture** – Packets are collected live or read from PCAP.
2. **Analysis** – Sliding counters and thresholds are applied.
3. **Alerting** – Alerts are logged with timestamps when thresholds are exceeded.

---

4. Results

The IDS was tested on two kinds of data:

- **Normal browsing traffic** – No alerts were generated, confirming baseline accuracy.
- **Attack simulations** –
    - A ping flood produced an ICMP flood alert.
    - An Nmap SYN scan triggered a scan detection.
    - Half-open TCP tests resulted in a high half-open connection alert.

These results show that the detection logic works as intended for the selected scenarios.

---

5. Limitations

- **Accuracy issues:** Legitimate monitoring tools or load balancers may resemble attacks.
- **Performance constraints:** Single-threaded design works only for smaller traffic volumes.
- **Scope:** Only basic L3/L4 behaviors are monitored; application-level attacks are out of scope.

---

6. Recommendations

- Expand coverage to UDP anomalies (e.g., amplification attempts).
- Save alerts into persistent log files and allow SIEM integration.
- Add a simple dashboard for visualization.
- Provide whitelist functionality for trusted hosts.
- Map detections to frameworks such as MITRE ATT&CK for better context.