Proof of Concept (PoC) Report  
Network Intrusion Detection System (NIDS)

# Objective

The objective of this Proof of Concept (PoC) is to design and implement a lightweight   
Network Intrusion Detection System (NIDS) that can monitor network traffic (live or from PCAP files)   
and raise alerts for suspicious or malicious activities. The focus is on detecting ICMP pings,   
TCP connection attempts, common port scanning techniques, and simple suspicious behaviors such as floods.

# Scope

This PoC is limited to basic detection of network activities that can be identified using   
packet headers. The scope includes:  
- ICMP Echo Requests/Replies (Ping detection)  
- TCP SYN attempts (connection initiation and half-open connections)  
- Common scan patterns (SYN/NULL/FIN scans, multi-port sweeps)  
- Suspicious behaviors such as repeated floods or high-rate attempts  
The system does not perform deep packet inspection or advanced anomaly detection at this stage.

# Environment Setup

The PoC was developed in Python using the Scapy library for packet parsing and sniffing.   
The project is designed to run on Windows (with Administrator privileges for live capture).  
Key components:  
- Language: Python 3.10+  
- Libraries: Scapy, Pytest  
- Execution: CLI (Command Prompt or PowerShell)  
- Input: PCAP file or live capture

# Implementation

The solution is divided into the following modules:  
1. \*\*alerts.py\*\* – Handles alert logging (console + log file).  
2. \*\*parser.py\*\* – Packet parsing utilities (extracting IP, TCP, ICMP fields).  
3. \*\*detector.py\*\* – Core detection logic:  
 - ICMP Flood detection: Alerts when ICMP echo requests exceed threshold within a time window.  
 - TCP SYN/connection attempts: Tracks SYNs and detects half-open connections.  
 - Port scans: Alerts on SYN/NULL/FIN scans with multiple distinct ports.  
 - Suspicious behavior: Detects high-rate attempts to multiple ports/hosts.  
4. \*\*main.py\*\* – Entry point, supports `--pcap` and `--live` modes.

# Testing & Results

The PoC was tested using two PCAP files:  
- \*\*Normal traffic PCAP\*\*: Contains standard ICMP ping and normal TCP connections. The IDS raised only informational alerts.  
- \*\*Attack traffic PCAP\*\*: Contains SYN scans, NULL scans, and ICMP floods. The IDS successfully detected:  
 - ICMP\_FLOOD alerts  
 - SYN\_SCAN alerts (multiple ports)  
 - NULL\_SCAN alerts (empty TCP flags)  
 - HALF\_OPEN\_SURGE alerts for SYN floods  
  
Unit tests were implemented for ICMP floods, SYN scan detection, and NULL/FIN scans using mocked packets.

# False Positives Considerations

Potential false positives may arise due to:  
- Legitimate monitoring tools sending frequent ICMP pings (e.g., health checks)  
- CDN services or vulnerability scanners producing port-scan-like behavior  
- Load balancers/NAT gateways creating unusual TCP flag patterns  
Mitigations include: tuning thresholds, allowlisting known IPs, and adding rule-based filtering.

# Next Steps

For future iterations, the following enhancements are proposed:  
- Add UDP-based detection (UDP scans, amplification attempts)  
- Implement TCP stream reassembly for better half-open connection detection  
- Persist detection results into a database (SQLite/PostgreSQL)  
- Export alerts in JSON format and integrate with SIEM tools  
- Add configuration-driven rule engine for flexible tuning