# Name : Sandhya Singh

# Intern ID : 233

# Proof of Concept (POC) – Lightweight Network Intrusion Detection System

## Objective

To build a lightweight Network Intrusion Detection System (IDS) in Python that:  
- Monitors live or offline traffic (PCAP).  
- Detects ICMP pings (echo requests/replies).  
- Detects TCP connection attempts (SYN packets).  
- Detects common scan patterns (SYN/NULL/FIN scans, repeated connection attempts).  
- Detects suspicious behaviors (ICMP floods, SYN floods).

## Tools Used

- Programming Language: Python 3  
- Libraries: scapy, collections, time  
- Traffic Generation Tools: ping, nmap  
- Test PCAPs: Generated with Wireshark/Tshark

## Implementation Steps

1. Setup Environment  
 pip install scapy  
  
2. Develop IDS Script (ids.py)  
 - Detect ICMP Echo requests/replies.  
 - Detect TCP SYN attempts.  
 - Track repeated SYN attempts across ports (port scan).  
 - Track high-rate ICMP/SYN packets (flood detection).  
  
3. Run IDS on live interface or offline PCAPs.  
 sudo python3 ids.py  
 sudo python3 ids.py -r traffic.pcap  
  
4. Generate Traffic  
 - Normal traffic: Web browsing.  
 - ICMP traffic: ping <target\_ip>  
 - SYN scan traffic: nmap -sS <target\_ip>

## Detection Logic (Code Snippet)

if packet.haslayer(ICMP):  
 if packet[ICMP].type == 8:  
 print(f"[ALERT] ICMP Echo Request from {src}")  
  
if packet.haslayer(TCP) and packet[TCP].flags == "S":  
 print(f"[ALERT] TCP SYN attempt from {src} to {dst}:{dport}")

## Demo Results

Normal PCAP  
- Minimal/no alerts.

Attack PCAP  
- ICMP Flood:  
 [ALERT] ICMP Flood detected from 192.168.1.5  
- Port Scan:  
 [ALERT] Port Scan detected from 192.168.1.10 (>10 ports in 5s)  
- SYN Flood:  
 [ALERT] SYN Flood detected from 192.168.1.12  
  
(\*Insert screenshots of terminal alerts here\*)

## Conclusion

- Successfully detected ICMP pings, SYN attempts, port scans, and flood behavior.  
- Works both in live capture and PCAP replay mode.  
- Can be extended with:  
 - Signature-based rules.  
 - Logging to files.  
 - Real-time dashboard.

## Next Steps

- Add detection for NULL/FIN scans.  
- Build a web UI for alerts.  
- Integrate with a database for storing incidents.