Network IDS – Weekly Task Report

NAME : SIDDHI SUNIL BENSEKAR.

INTERN ID : 344

Objective

Build a lightweight Network Intrusion Detection System (IDS) to detect ICMP pings, TCP connection attempts, and common scan patterns. The system should raise alerts for suspicious behavior detected in PCAP files.

Build a lightweight IDS to detect:

- ICMP floods
- TCP SYN scans
- Port scans
  ...using PCAP file analysis.

Detection Logic

Type of Attack	**Detection Rule**
ICMP Flood	Count ICMP packets per source IP. Alert if count > 5.
SYN Scan	Count TCP SYN packets without ACK per source. Alert if count > 10.
Port Scan	Detect SYN attempts to many different ports from the same source IP.

False Positives

- ICMP traffic may be legitimate (e.g., diagnostics).
- Dropped connections can cause SYN without ACK.
- Thresholds must be tuned to match normal behavior.

Next Steps

- Add UDP scan detection.
- Use time-window-based monitoring.
- Enable live sniffing and real-time dashboards.
- Integrate with external alert systems.

Demo Results

PCAP File	Description	Alerts Generated
normal_traffic.pcap	Normal browsing and DNS	No alerts
scan_activity.pcap	ICMP flood + SYN scan	ICMP flood from 192.168.1.10, SYN scan from 192.168.1.15

How IDS Works (Expanded View)

According to GeeksforGeeks and Stamus Networks:

- IDS monitors traffic and compares it against known attack patterns or behavioral anomalies.
- It can detect:
- **Reconnaissance** (ping sweeps, scans)
- **Exploitation attempts** (e.g., buffer overflow, SQL injection)
- **Privilege escalation**

- **Denial of Service (DoS)** attacks

Implementation Tips

```
from scapy.all import rdpcap, TCP


syn_counts = {}


packets = rdpcap("scan_activity.pcap")
for pkt in packets:
    if pkt.haslayer(TCP) and pkt[TCP].flags == "S":
        src = pkt[IP].src
        syn_counts[src] = syn_counts.get(src, 0) + 1


for ip, count in syn_counts.items():
    if count > 10:
        print(f"[ALERT] SYN scan from {ip}")
```