Network Intrusion Prevention System (IPS)

Name:Samiksha Patil

Intern ID:280

Introduction

A Network Intrusion Prevention System (IPS) is a security solution that monitors traffic in real-time and takes preventive action against malicious activity. Unlike an IDS (Intrusion Detection System) that only alerts, an IPS actively blocks traffic.

This project builds a lightweight IPS capable of blocking basic attacks:

1.ICMP ping floods

2.TCP SYN floods / half-open connections

3.Port scans (NULL/FIN/SYN scans)

4.Suspicious HTTP/SQL injection payloads

Objectives

1. Detect and block ICMP floods (ping attacks).

2.Drop TCP flood attempts or half-open SYN connections.

3.Prevent scanning patterns (Nmap SYN/NULL/FIN scans).

4. Block malicious payloads (HTTP SQL injection signatures).

5. Provide logs + alerts for blocked traffic.

Core Features

1. ICMP Flood Protection

    -Detect repeated pings in short time.

    -Block offending IP.

2. TCP SYN Flood & Half-Open Handling

    -Track incomplete TCP handshakes.

    -Drop IPs with excessive half-open requests.

3. Scan Pattern Detection

    -NULL, FIN, Xmas scans → typical Nmap patterns.

    -Block repeated port probe attempts.

4. Payload Filtering

    -Detect suspicious strings in HTTP traffic:

1)"UNION SELECT"

2)"DROP TABLE"

3)"' OR 1=1--" (SQL Injection patterns).

5. Logging & Alerts

-Log all dropped packets.

-Optionally send alerts.

Sample Run (Demo Flow)

Input:

normal.pcap (benign traffic)

attack.pcap (includes ping floods + SQL injection attempt)

IPS Process:

1. Capture packets using scapy sniff.

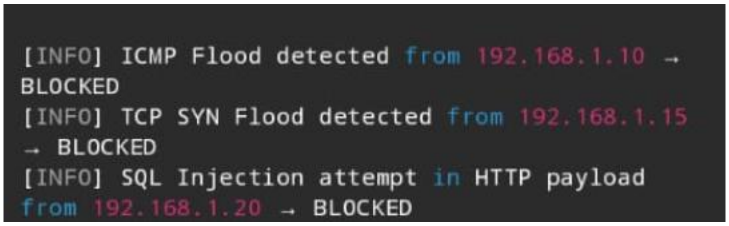
2. Apply rules:

-If ICMP flood detected → block IP.

-If repeated SYN flood → block IP.

-If suspicious payload → block connection.

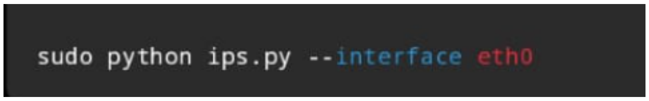
Output:



```
[INFO] ICMP Flood detected from 192.168.1.10 →
BLOCKED
[INFO] TCP SYN Flood detected from 192.168.1.15
→ BLOCKED
[INFO] SQL Injection attempt in HTTP payload
from 192.168.1.20 → BLOCKED
```

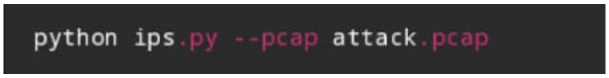
Example Commands

Run IPS on live traffic:



```
sudo python ips.py --interface eth0
```

Run against a PCAP file:



```
python ips.py --pcap attack.pcap
```

Testing (Deliverables)

1. Demo

   -Run against benign traffic (normal.pcap) → traffic passes.

   -Run against attack traffic (attack.pcap) → malicious packets blocked.

2. Report

   -Explain detection logic (thresholds, signatures).

   -Discuss false positives (e.g., heavy legitimate ping use).

   -Suggest improvements.

3. Unit Tests

   -Test ICMP detection logic.

   -Test SYN flood blocking.

   -Test payload filtering.

Advantages

1.Lightweight → can run on Linux, Windows, or inside VM.

2.Teaches fundamentals of IPS logic.

3.Prevents common network attacks.

4.Works in real time or offline mode with PCAP.

Limitations

1. Cannot handle encrypted traffic (HTTPS).

2.High-speed networks may overload Python.

3.Rules are signature/threshold-based, not AI-powered.

4.Advanced evasion techniques (fragmentation, tunneling) may bypass.

Future Improvements

1. Support TLS/SSL inspection.

2.Add machine learning anomaly detection.

3.Integration with iptables or Suricata/Snort rules.

4.Centralized log/alert dashboard.

5.Multi-threaded support for high throughput.