**IPS — Weekly Task**

**Title:** Network IPS — block pings, malicious connections, and simple exploits

# Objective

Build a lightweight Intrusion Prevention System that not only detects but also **blocks** malicious traffic in real time. The IPS should:

* Block ICMP ping floods.

1. sudo apt update

sudo apt install python3

sudo apt install python3 python3-pip nftables hping3 nmap curl -y

sudo apt install python3-scapy -y

python3 -c "import scapy.all; print('Scapy installed OK!')"

1. sudo nft add table inet ips

sudo nft add chain inet ips input '{ type filter hook input priority 0; policy accept; }'

sudo nft add set inet ips blacklist { type ipv4\_addr\; flags timeout\; }

sudo nft add rule inet ips input ip saddr @blacklist drop

1. sudo nft list ruleset
2. We don’t care about banning or thresholds right now. First we must prove:  
    Scapy can actually **see packets** on your interface.

sudo nano /opt/test\_sniff.py

#!/usr/bin/env python3

from scapy.all import sniff, IP

def show(pkt):

if IP in pkt:

print(f"Got packet: {pkt[IP].src} -> {pkt[IP].dst}")

print("[TEST] Starting sniff on eth0...")

sniff(prn=show, store=0, iface="eth0")

1. sudo python3 /opt/test\_sniff.py
2. open a new terminal

ping -c 3 8.8.8.8

Expected output in Terminal 1:

[TEST] Starting sniff on eth0...

Got packet: 10.0.2.15 -> 8.8.8.8

Got packet: 8.8.8.8 -> 10.0.2.15

**ICMP + SYN Flood Detection**

1. sudo nano /opt/ips\_main.py

#!/usr/bin/env python3

from scapy.all import sniff, IP, ICMP, TCP

import subprocess, time

from collections import defaultdict, deque

# Detection settings

WINDOW = 5

ICMP\_THRESHOLD = 3 # packets in 5s

SYN\_THRESHOLD = 3 # SYNs in 5s

BAN\_TIME = 60 # seconds

icmp\_counts = defaultdict(lambda: deque())

syn\_counts = defaultdict(lambda: deque())

def ban\_ip(ip, reason):

print(f"[IPS] Banned {ip} for {BAN\_TIME}s ({reason})")

result = subprocess.run(

["sudo", "nft", "add", "element", "inet", "ips", "blacklist",

f"{{ {ip} timeout {BAN\_TIME}s }}"],

capture\_output=True, text=True

)

if result.stderr:

print("[ERROR]", result.stderr.strip())

def handle\_icmp(pkt):

if ICMP in pkt and IP in pkt:

src = pkt[IP].src

now = time.time()

icmp\_counts[src].append(now)

while icmp\_counts[src] and now - icmp\_counts[src][0] > WINDOW:

icmp\_counts[src].popleft()

print(f"[DEBUG] ICMP count {src}: {len(icmp\_counts[src])}")

if len(icmp\_counts[src]) > ICMP\_THRESHOLD:

ban\_ip(src, f"ICMP flood >{ICMP\_THRESHOLD}/{WINDOW}s")

icmp\_counts[src].clear()

def handle\_syn(pkt):

if TCP in pkt and IP in pkt and pkt[TCP].flags == "S":

src = pkt[IP].src

now = time.time()

syn\_counts[src].append(now)

while syn\_counts[src] and now - syn\_counts[src][0] > WINDOW:

syn\_counts[src].popleft()

print(f"[DEBUG] SYN count {src}: {len(syn\_counts[src])}")

if len(syn\_counts[src]) > SYN\_THRESHOLD:

ban\_ip(src, f"SYN flood >{SYN\_THRESHOLD}/{WINDOW}s")

syn\_counts[src].clear()

def process(pkt):

if IP in pkt:

if ICMP in pkt:

handle\_icmp(pkt)

if TCP in pkt:

handle\_syn(pkt)

print("[IPS] Starting... Listening for ICMP and TCP SYN floods")

sniff(prn=process, store=0, iface="eth0")

1. Terminal 1: sudo python3 /opt/ips\_main.py

Terminal 2: ping -c 5 8.8.8.8

Expected: [DEBUG] ICMP count 10.0.2.15: 1

[DEBUG] ICMP count 10.0.2.15: 2

...

[IPS] Banned 10.0.2.15 for 60s (ICMP flood >3/5s)

1. Terminal 2: sudo hping3 -S -p 80 -i u1000 -c 50 10.0.2.15

Expected in Terminal 1:

[DEBUG] SYN count 10.0.2.15: 1

[DEBUG] SYN count 10.0.2.15: 2

...

[IPS] Banned 10.0.2.15 for 60s (SYN flood >3/5s)

1. **Verifying Bans**

Check current blacklist:

sudo nft list set inet ips blacklist

Example output:

set blacklist {

type ipv4\_addr

flags timeout

elements = { 10.0.2.15 expires 59s, 8.8.8.8 expires 55s }

}

**Managing Bans**

**Clear all bans:**

sudo nft flush set inet ips blacklist

**Delete one IP (Zsh-safe):**

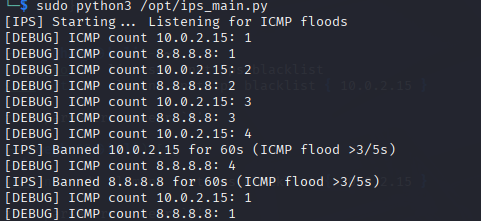
sudo nft delete element inet ips blacklist '{ 10.0.2.15 }'

**Permanent ban (no timeout):**

sudo nft add element inet ips blacklist { 10.0.2.15 }

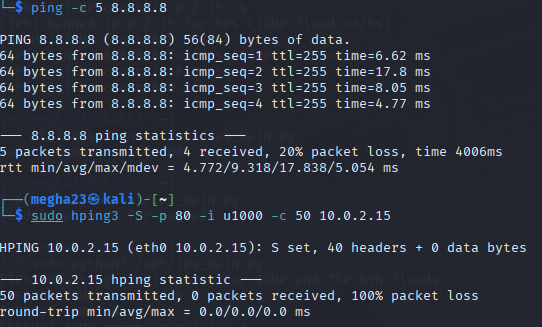
**Output:**

* IPS running

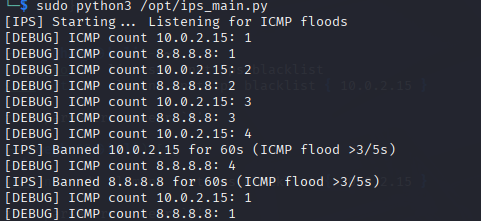


* ICMP flood test

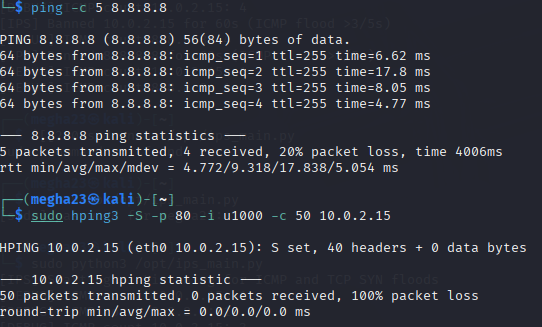
Terminal 2:



IPS output in terminal 1

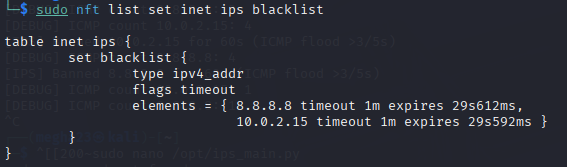


* SYN flood test

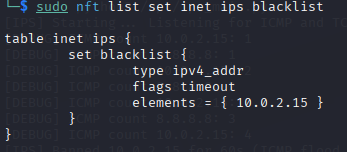


* nftables blacklist showing banned IPs

Blacklist Verification



Blacklist Empty After Timeout



**Conclusion**

This IPS successfully:

* Detected ICMP and SYN flood attacks
* Automatically added attackers to a firewall blacklist
* Blocked malicious traffic for a defined period
* Allowed verification and management of bans via nftables