External Traffic Capture Setup Guide

🎯 Architecture Overview

```
|----------------------------------------------------------------|  |
|                    Your IDS Pipeline Setup                     |  |
|----------------------------------------------------------------|  |


 |-------------------------|        |-------------------------|
 | External Device         | Ethernet| Your IDS System        |
 | (PCAP Replay)           |-------->| (This Laptop)          |
 |                         |  Cable  |                        |
 | • Kali Linux            |        | USB Adapter            |
 | • Raspberry Pi          |        | enx00e04c36074c        |
 | • Another PC            |        |                        |
 | • Attack Gen            |        |   |-----------------|  |
 |                         |        |   | AF_PACKET       |  |
 | Replays PCAP:           |        |   | Suricata        |  |
 | • tcpreplay             |        |   |-----------------|  |
 | • scapy                 |        |           |            |
 | • hping3                |        |           ▼            |
 | • metasploit            |        |   |-----------------|  |
 |                         |        |   |     Kafka       |  |
 | IP: 192.168.100.2       |        |   |-----------------|  |
 |                         |        |           |            |
 |-------------------------|        |           ▼            |
                                    |   |-----------------|  |
                                    |   | ML Consumer     |  |
                                    |   | (Detection)     |  |
                                    |   |-----------------|  |
                                    |                        |
                                    | IP: 192.168.100.1      |
                                    |-------------------------|
```
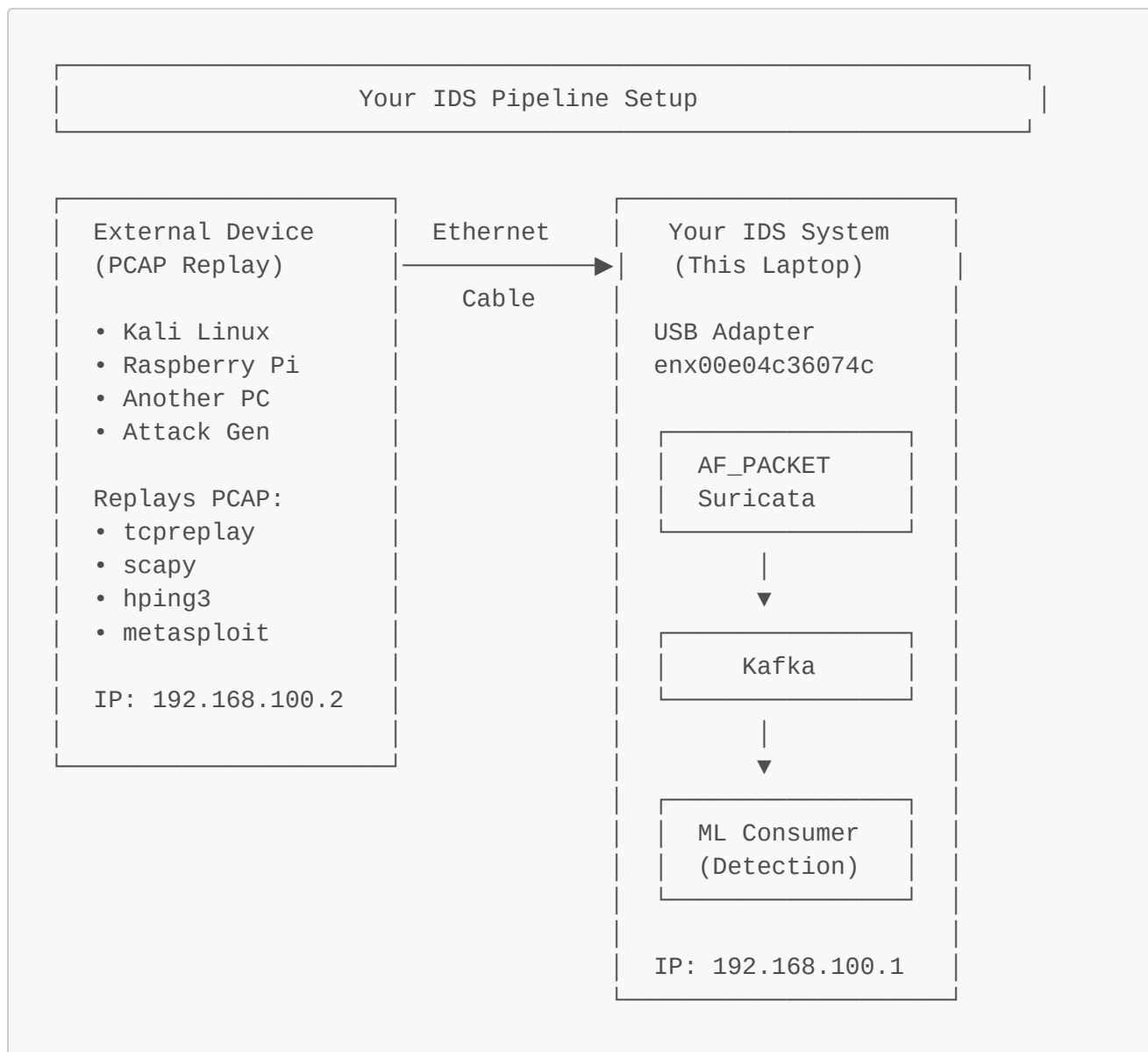
🚀 Quick Start

Step 1: Setup Your IDS System (This Laptop)

```
cd ~/Programming/IDS/dpdk_suricata_ml_pipeline/scripts

# Configure USB adapter for external traffic
sudo ./00_setup_external_capture.sh

# Start the complete IDS pipeline
sudo ./quick_start.sh
# Select option 1: Start Complete Pipeline
```

Step 2: Connect Physical Cable

- Connect Ethernet cable from external device to your USB adapter
- Wait for link up (check with `ip link show enx00e04c36074c`)

Step 3: Configure External Device

```
# On the external device (e.g., Kali Linux, Raspberry Pi, etc.)
sudo ip addr add 192.168.100.2/24 dev eth0
sudo ip link set eth0 up
sudo ip route add default via 192.168.100.1

# Test connectivity
ping 192.168.100.1
```

Step 4: Replay Traffic from External Device

```
# On the external device
# Option A: tcpreplay (best for PCAPs)
sudo tcpreplay -i eth0 -K --mbps 10 attack_traffic.pcap

# Option B: tcpreplay topspeed
sudo tcpreplay -i eth0 -t attack_traffic.pcap

# Option C: Loop PCAP
sudo tcpreplay -i eth0 -K --loop 10 --mbps 10 attack_traffic.pcap

# Option D: Scapy script
sudo python attack_generator.py
```

PROF

Step 5: Monitor on IDS System

```
# Watch live traffic
sudo tcpdump -i enx00e04c36074c -n

# Watch Suricata alerts
tail -f logs/suricata/eve.json | jq .

# Watch ML predictions
tail -f logs/ml/consumer.log

# Check stats
suricatasc -c dump-counters
```

🏷 Detailed Setup Instructions

On Your IDS System (This Laptop)

1. Configure Network Interface

```
cd ~/Programming/IDS/dpdk_suricata_ml_pipeline/scripts
sudo ./00_setup_external_capture.sh
```

This script will:

- ✅ Bring up USB adapter
- ✅ Assign IP: 192.168.100.1/24
- ✅ Enable promiscuous mode
- ✅ Optimize for packet capture
- ✅ Disable offload features

2. Start IDS Pipeline

```
# Option A: Interactive menu
sudo ./quick_start.sh
# Select: 1 (Start Complete Pipeline)

# Option B: Manual steps
sudo ./02_setup_kafka.sh                # Start Kafka
sudo ./03_start_suricata_afpacket.sh    # Start Suricata
./04_start_ml_consumer.sh               # Start ML Consumer
```

3. Verify Setup

```
# Check interface is up and configured
ip addr show enx00e04c36074c

# Should show:
# 4: enx00e04c36074c: <BROADCAST,MULTICAST,PROMISC,UP,LOWER_UP>
#     inet 192.168.100.1/24 ...
```

On External Device (Traffic Generator)

Device Options

- **Kali Linux** - Full pentesting suite

- **Raspberry Pi** - Lightweight attack generator
- **Another Laptop** - Any Linux/Mac/Windows system
- **VM** - Virtual machine with bridged networking

Network Configuration

Method 1: Static IP (Recommended)

```
# On external device
sudo ip addr flush dev eth0
sudo ip addr add 192.168.100.2/24 dev eth0
sudo ip link set eth0 up
sudo ip route add default via 192.168.100.1

# Verify
ip addr show eth0
ping 192.168.100.1
```

Method 2: DHCP (if you configure DHCP server)

```
sudo dhclient eth0
```

Traffic Generation Tools

1. tcpreplay (Recommended)

```
# Install
sudo apt install tcpreplay    # Ubuntu/Debian
sudo yum install tcpreplay    # CentOS/RHEL

# Basic replay
sudo tcpreplay -i eth0 capture.pcap

# Replay with speed control (10 Mbps)
sudo tcpreplay -i eth0 -K --mbps 10 capture.pcap

# Replay as fast as possible
sudo tcpreplay -i eth0 -t capture.pcap

# Loop replay 10 times
sudo tcpreplay -i eth0 --loop 10 capture.pcap

# Edit packets on-the-fly
sudo tcpreplay -i eth0 \
    --enet-dmac=00:e0:4c:36:07:4c \
```

```
        --enet-smac=aa:bb:cc:dd:ee:ff \
        capture.pcap
```

2. Scapy (Python)

```python
#!/usr/bin/env python3
from scapy.all import *

# Send packets to IDS
target_ip = "192.168.100.1"
iface = "eth0"

# Send single packet
packet = IP(dst=target_ip)/TCP(dport=80, flags="S")
send(packet, iface=iface)

# Send multiple packets
for i in range(100):
    packet = IP(dst=target_ip)/TCP(dport=80+i, flags="S")
    send(packet, iface=iface, verbose=0)

# Replay PCAP
packets = rdpcap("capture.pcap")
sendp(packets, iface=iface)
```

3. hping3 (Traffic Generator)

```bash
# Install
sudo apt install hping3

# SYN flood
sudo hping3 -S 192.168.100.1 -p 80 --flood

# Port scan simulation
for port in {1..1000}; do
    sudo hping3 -S 192.168.100.1 -p $port -c 1
done

# UDP flood
sudo hping3 --udp 192.168.100.1 -p 53 --flood

# ICMP flood
sudo hping3 --icmp 192.168.100.1 --flood
```

4. nmap (Port Scanning)

```
# SYN scan
sudo nmap -sS 192.168.100.1 -p 1-1000

# Aggressive scan
sudo nmap -A 192.168.100.1

# All ports
sudo nmap -p- 192.168.100.1
```

5. Custom Attack Scripts

```
# HTTP requests
while true; do
    curl http://192.168.100.1
    sleep 0.1
done

# Netcat connections
while true; do
    echo "GET / HTTP/1.0" | nc 192.168.100.1 80
done
```

🎮 Usage Scenarios

Scenario 1: Replay Known Attack PCAPs

```
# On external device
cd /path/to/pcaps
sudo tcpreplay -i eth0 -K --mbps 10 ddos_attack.pcap

# On IDS system
tail -f logs/suricata/eve.json | jq 'select(.event_type=="alert")'
```

Scenario 2: Generate Live Attack Traffic

```
# On external device - SQL injection attempts
for i in {1..100}; do
    curl "http://192.168.100.1/login?user=admin'+OR+'1'='1"
    sleep 0.5
done

# On IDS system - watch ML predictions
tail -f logs/ml/consumer.log | grep "attack"
```

Scenario 3: Port Scan Detection

```
# On external device
sudo nmap -sS -p 1-10000 192.168.100.1

# On IDS system
suricatasc -c dump-counters | grep -i scan
```

Scenario 4: DDoS Simulation

```
# On external device
sudo hping3 -S 192.168.100.1 -p 80 --flood --rand-source

# On IDS system
tail -f logs/suricata/eve.json | jq 'select(.event_type=="alert") | .alert'
```

🔧 Advanced Configuration

Enable SPAN/Mirror Port (Optional)

If you want to capture traffic between two other devices:

```
# On a switch or router, configure port mirroring to your IDS port
# This allows passive monitoring of real network traffic
```

Multiple External Devices

```
# Configure additional devices on the same network
# Device 2: 192.168.100.3
# Device 3: 192.168.100.4
# etc.

# They can all send traffic that your IDS will analyze
```

Traffic Shaping

```
# On external device - limit bandwidth
sudo tc qdisc add dev eth0 root tbf rate 10mbit burst 32kbit latency 400ms
```

```
# Remove limit
sudo tc qdisc del dev eth0 root
```

📊 Monitoring & Validation

Verify Traffic is Flowing

On IDS System:

```
# Live packet capture
sudo tcpdump -i enx00e04c36074c -n -c 10

# Count packets
sudo tcpdump -i enx00e04c36074c -n | pv -l -a > /dev/null

# Watch specific traffic
sudo tcpdump -i enx00e04c36074c -n 'tcp port 80'

# Packet statistics
watch -n 1 'ip -s link show enx00e04c36074c'
```

Check Suricata is Processing:

```
# Live alerts
tail -f logs/suricata/eve.json | jq .

# Alert count
cat logs/suricata/eve.json | jq 'select(.event_type=="alert")' | wc -l

# Stats
suricatasc -c dump-counters | grep -E "(capture|decode|alert)"
```

Check ML Pipeline:

```
# ML consumer logs
tail -f logs/ml/consumer.log

# Kafka messages
kafka-console-consumer.sh --bootstrap-server localhost:9092 \
    --topic suricata-alerts --from-beginning
```

🐛 Troubleshooting

Issue: No Connectivity Between Devices

Check physical connection:

```
# On IDS system
ip link show enx00e04c36074c | grep "state UP"
ethtool enx00e04c36074c | grep "Link detected"

# Should show: Link detected: yes
```

Check IP configuration:

```
# On IDS system
ip addr show enx00e04c36074c
# Should show: inet 192.168.100.1/24

# On external device
ip addr show eth0
# Should show: inet 192.168.100.2/24
```

Test with ping:

```
# From IDS system
ping 192.168.100.2

# From external device
ping 192.168.100.1
```

Check ARP:

```
# On both systems
arp -a | grep 192.168.100

# If missing, try ping first to populate ARP table
```

Issue: No Traffic Captured by Suricata

Verify promiscuous mode:

```
ip link show enx00e04c36074c | grep PROMISC
# Should show PROMISC flag
```

Check Suricata is listening:

```
ps aux | grep suricata
# Should show: suricata --af-packet=enx00e04c36074c

# Check Suricata logs
sudo tail -50 /var/log/suricata/suricata.log
```

Test with tcpdump:

```
# If tcpdump sees traffic but Suricata doesn't, there's a config issue
sudo tcpdump -i enx00e04c36074c -n
```

Issue: High Packet Loss

Check ring buffer size:

```
ethtool -g enx00e04c36074c
```

Check for drops:

```
ip -s link show enx00e04c36074c | grep -A 5 "RX:"
# Look for "dropped" counter
```

Reduce replay speed:

```
# On external device
sudo tcpreplay -i eth0 --mbps 5 capture.pcap  # Slower rate
```

Issue: External Device Can't Send Traffic

Check firewall:

```
# On IDS system - temporarily disable
sudo ufw disable
# Or allow traffic
sudo iptables -P INPUT ACCEPT
sudo iptables -P FORWARD ACCEPT
```

Check routing:

```
# On external device
ip route show
# Should have route to 192.168.100.0/24
```

📚 Example PCAP Files

You can use these public PCAP repositories:

1. Your Project PCAPs

```
# Use the samples in your project
ls ~/Programming/IDS/pcap_samples/
```

2. Download Attack PCAPs

```
# CICIDS2017 dataset (already in your notebooks)
# CICIDS2018 dataset (already in your notebooks)

# Additional sources:
# - https://www.malware-traffic-analysis.net/
# - https://www.netresec.com/?page=PcapFiles
# - https://github.com/markofu/hackerone
```

3. Generate Custom PCAPs

```
# Capture your own traffic
sudo tcpdump -i any -w capture.pcap -c 1000

# Transfer to external device
scp capture.pcap user@external-device:/tmp/
```

🎯 Testing Checklist

- ☐ USB adapter configured with IP 192.168.100.1
- ☐ Promiscuous mode enabled
- ☐ External device configured with IP 192.168.100.2
- ☐ Physical Ethernet cable connected
- ☐ Ping successful between devices

- ☐ Kafka running (`ps aux | grep kafka`)
- ☐ Suricata running (`ps aux | grep suricata`)
- ☐ ML consumer running (`ps aux | grep ml_kafka_consumer`)
- ☐ tcpdump shows packets on enx00e04c36074c
- ☐ Suricata eve.json is being written
- ☐ ML predictions appearing in logs

🚀 Complete Workflow Example

IDS System (Your Laptop)

```
# Terminal 1: Setup and start pipeline
cd ~/Programming/IDS/dpdk_suricata_ml_pipeline/scripts
sudo ./00_setup_external_capture.sh
sudo ./quick_start.sh  # Select option 1

# Terminal 2: Monitor traffic
sudo tcpdump -i enx00e04c36074c -n

# Terminal 3: Watch alerts
tail -f logs/suricata/eve.json | jq 'select(.event_type=="alert")'

# Terminal 4: Watch ML predictions
tail -f logs/ml/consumer.log
```

External Device (Attack Generator)

```
# Configure network
sudo ip addr add 192.168.100.2/24 dev eth0
sudo ip link set eth0 up

# Test connectivity
ping 192.168.100.1

# Replay attack traffic
sudo tcpreplay -i eth0 -K --mbps 10 /path/to/attack.pcap

# Watch your IDS system detect the attacks!
```

📖 Summary

Component	Configuration
IDS System IP	192.168.100.1/24

Component	Configuration
External Device IP	192.168.100.2/24
Interface	enx00e04c36074c (USB adapter)
Mode	AF_PACKET (promiscuous)
Setup Script	`00_setup_external_capture.sh`
Start Pipeline	`quick_start.sh` option 1
Replay Tool	tcpreplay (on external device)

🎓 Next Steps

1. ✅ Run setup script: `sudo ./00_setup_external_capture.sh`
2. ✅ Connect external device via Ethernet cable
3. ✅ Configure external device with IP 192.168.100.2
4. ✅ Start IDS pipeline: `sudo ./quick_start.sh`
5. ✅ Replay traffic from external device
6. ✅ Watch attacks being detected!

Your IDS is now ready to receive external traffic! 🎉