from scapy.all import sniff

from scapy.layers.inet import IP, TCP, UDP

def packet\_callback(packet):

# Check if the packet has an IP layer

if IP in packet:

ip\_layer = packet[IP]

print(f"Source IP: {ip\_layer.src}")

print(f"Destination IP: {ip\_layer.dst}")

# Determine protocol

if TCP in packet:

protocol = "TCP"

payload = bytes(packet[TCP].payload)

elif UDP in packet:

protocol = "UDP"

payload = bytes(packet[UDP].payload)

else:

protocol = "Other"

payload = bytes(packet.payload)

print(f"Protocol: {protocol}")

print(f"Payload: {payload[:20]}") # Print first 20 bytes of payload

print("-" \* 50)

def http\_packet\_callback(packet):

if packet.haslayer(TCP) and packet.haslayer(IP):

ip\_layer = packet[IP]

tcp\_layer = packet[TCP]

payload = bytes(packet[TCP].payload)

# Check if the packet contains HTTP data

if b"HTTP" in payload:

print(f"Source IP: {ip\_layer.src}")

print(f"Destination IP: {ip\_layer.dst}")

print(f"Payload: {payload[:50]}") # Print first 50 bytes of HTTP payload

print("-" \* 50)

# Start sniffing only HTTP traffic (port 80)

print("Starting HTTP packet sniffer...")

sniff(filter="tcp port 80", prn=http\_packet\_callback, count=10)

# Start sniffing

print("Starting packet sniffer...")

sniff(prn=packet\_callback, count=10)