CodeAlpha

Task : 1 Build a network sniffer in Python that captures and analyzes network traffic. This project will help you understand how data flows on a network and how network packets are structured

from scapy.all import sniff, hexdump

from scapy.layers.l2 import Ether

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

from datetime import datetime

def process\_packet(packet):

print(f"\n{'='\*50}")

print(f"Time: {datetime.now().strftime('%Y-%m-%d %H:%M:%S')}")

print(f"Packet Summary: {packet.summary()}")

print(f"{'-'\*50}")

# Display Ethernet Layer

if packet.haslayer(Ether):

eth = packet[Ether]

print(f"Ethernet Frame:")

print(f"\tSource MAC: {eth.src}")

print(f"\tDestination MAC: {eth.dst}")

print(f"\tType: {eth.type}")

# Display IP Layer

if packet.haslayer(IP):

ip = packet[IP]

print(f"IP Layer:")

print(f"\tVersion: {ip.version}")

print(f"\tHeader Length: {ip.ihl}")

print(f"\tTTL: {ip.ttl}")

print(f"\tProtocol: {ip.proto}")

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

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

# Display TCP Layer

if packet.haslayer(TCP):

tcp = packet[TCP]

print(f"TCP Layer:")

print(f"\tSource Port: {tcp.sport}")

print(f"\tDestination Port: {tcp.dport}")

print(f"\tSequence Number: {tcp.seq}")

print(f"\tAcknowledgment Number: {tcp.ack}")

print(f"\tFlags: {tcp.flags}")

# Display UDP Layer

if packet.haslayer(UDP):

udp = packet[UDP]

print(f"UDP Layer:")

print(f"\tSource Port: {udp.sport}")

print(f"\tDestination Port: {udp.dport}")

print(f"\tLength: {udp.len}")

# Display ICMP Layer

if packet.haslayer(ICMP):

icmp = packet[ICMP]

print(f"ICMP Layer:")

print(f"\tType: {icmp.type}")

print(f"\tCode: {icmp.code}")

print(f"\tChecksum: {icmp.chksum}")

# Display Raw Data

# if packet.haslayer("Raw"):

# print("Raw Data:")

# print(hexdump(packet["Raw"].load))

# print(f"{'='\*50}\n")

def main():

print("Starting packet sniffer...\n")

sniff(prn=process\_packet, store=False)

if \_\_name\_\_ == "\_\_main\_\_":

main()

Output :

