Network Packet Analyzer

import socket

import struct

def parse\_ip\_header(data):

ip\_header = struct.unpack('!BBHHHBBH4s4s', data[:20])

version\_ihl = ip\_header[0]

version = version\_ihl >> 4

ihl = version\_ihl & 0xF

iph\_length = ihl \* 4

protocol = ip\_header[6]

src\_ip = socket.inet\_ntoa(ip\_header[8])

dst\_ip = socket.inet\_ntoa(ip\_header[9])

return protocol, src\_ip, dst\_ip, data[iph\_length:]

def main():

print("=== Windows Network Packet Analyzer ===")

try:

sock = socket.socket(socket.AF\_INET, socket.SOCK\_RAW, socket.IPPROTO\_IP)

host = socket.gethostbyname(socket.gethostname())

sock.bind((host, 0))

sock.setsockopt(socket.IPPROTO\_IP, socket.IP\_HDRINCL, 1)

sock.ioctl(socket.SIO\_RCVALL, socket.RCVALL\_ON)

print(f"Sniffing on: {host}")

while True:

data, addr = sock.recvfrom(65565)

proto, src, dst, payload = parse\_ip\_header(data)

if proto == 1:

proto\_name = "ICMP"

elif proto == 6:

proto\_name = "TCP"

elif proto == 17:

proto\_name = "UDP"

else:

proto\_name = f"Other ({proto})"

print(f"{src} -> {dst} | Protocol: {proto\_name} | Payload: {len(payload)} bytes")

except PermissionError:

print("Permission denied. Run this script as administrator.")

except KeyboardInterrupt:

print("\nSniffer stopped by user.")

sock.ioctl(socket.SIO\_RCVALL, socket.RCVALL\_OFF)

except Exception as e:

print(f"Error: {e}")

sock.ioctl(socket.SIO\_RCVALL, socket.RCVALL\_OFF)

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

main()

Output:

