Network Traffic Analyzer

"Background: Also called “Packet Sniffing”, network traffic analysis and monitoring is a process used to keep an eye on all incoming and outgoing traffic, network activity, and availability. It is used to detect anomalies, identifying operational issues, comparing real-time networks with historical data to identify potential threats and issues. For a project, you can analyze the data packets that are sent across the network. After analyzing these packets you can work on a report to give your insights, and if required, countermeasures if there is an anomaly. Summary: Also called “Packet Sniffing”, network traffic analysis and monitoring is a process used to keep an eye on all incoming and outgoing traffic, network activity, and availability. It is used to detect anomalies, identifying operational issues, comparing real-time networks with historical data to identify potential threats and issues. For a project, you can analyze the data packets that are sent across the network. After analyzing these packets you can work on a report to give your insights, and if required, countermeasures if there is an anomaly. Objective: : We need a software in which using your networking skills you can find out such anomalies in the network also as an extension, you could also devise software that could identify if there is a traffic analyzer in the network by an external agent."

CODE,

Creating a complete and fully functional network traffic analyzer software is beyond the scope of a simple code snippet. However, I can provide you with a basic outline of the steps you can follow to build a simple network traffic analyzer using Python and Scapy library. This will allow you to capture and analyze network packets and detect anomalies. Please note that for a production-level solution, you would need to consider performance, security, and other factors that are not covered in this basic outline.

```
import sys
from scapy.all import sniff, IP


def process_packet(packet):
    if IP in packet:
        src_ip = packet[IP].src
        dst_ip = packet[IP].dst
        print(f"Source IP: {src_ip} --> Destination IP: {dst_ip}")


def network_traffic_analyzer(interface, packet_count):
    print(f"Starting network traffic analysis on {interface}.")
    try:
        sniff(iface=interface, prn=process_packet, count=packet_count)
    except KeyboardInterrupt:
        print("Network traffic analysis stopped by the user.")
        sys.exit(0)


if __name__ == "__main__":
    interface = "eth0"  # Change this to the appropriate network interface on your
system
    packet_count = 10  # Number of packets to capture (change as needed)
    network_traffic_analyzer(interface, packet_count)
```

By,
M.Lakshmi priya
B.Leelavanitha
Azra gulnaz
Honey
Akhila

Thank you.