Network Monitor with sniff&scapy

I wrote a network monitor that can queries port protocol and only when its UDP or TCP code can show port number . Also code checks IP addresses in each packet. If the source IP is my computer, it is outgoing (OUT). If the destination IP is my computer, it is incoming (IN). If neither, it is OTHER

--THE CODE AND OUTPUT ->>

```
OUT | TCP | 172.16.34.112 :43012 → 139.144.71.6  :443
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
OUT | TCP | 172.16.34.112 :43012 → 139.144.71.6  :443
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
OUT | TCP | 172.16.34.112 :43012 → 139.144.71.6  :443
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
OUT | TCP | 172.16.34.112 :43012 → 139.144.71.6  :443
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
OUT | TCP | 172.16.34.112 :43012 → 139.144.71.6  :443
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
OUT | TCP | 172.16.34.112 :43012 → 139.144.71.6  :443
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
OUT | TCP | 172.16.34.112 :43012 → 139.144.71.6  :443
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
OUT | TCP | 172.16.34.112 :43012 → 139.144.71.6  :443
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
OUT | TCP | 172.16.34.112 :43012 → 139.144.71.6  :443
OUT | TCP | 172.16.34.112 :43012 → 139.144.71.6  :443
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
OUT | TCP | 172.16.34.112 :43012 → 139.144.71.6  :443
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
OUT | TCP | 172.16.34.112 :43012 → 139.144.71.6  :443
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
OUT | TCP | 172.16.34.112 :43012 → 139.144.71.6  :443
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
OUT | TCP | 172.16.34.112 :43012 → 139.144.71.6  :443
IN  | TCP | 139.144.71.6  :443   → 172.16.34.112 :43012
```

- As we can see ,the tool give us the going , coming ip's and query the port protocol if its TCP , UDP give us to port number ,if its other return other and nothing.

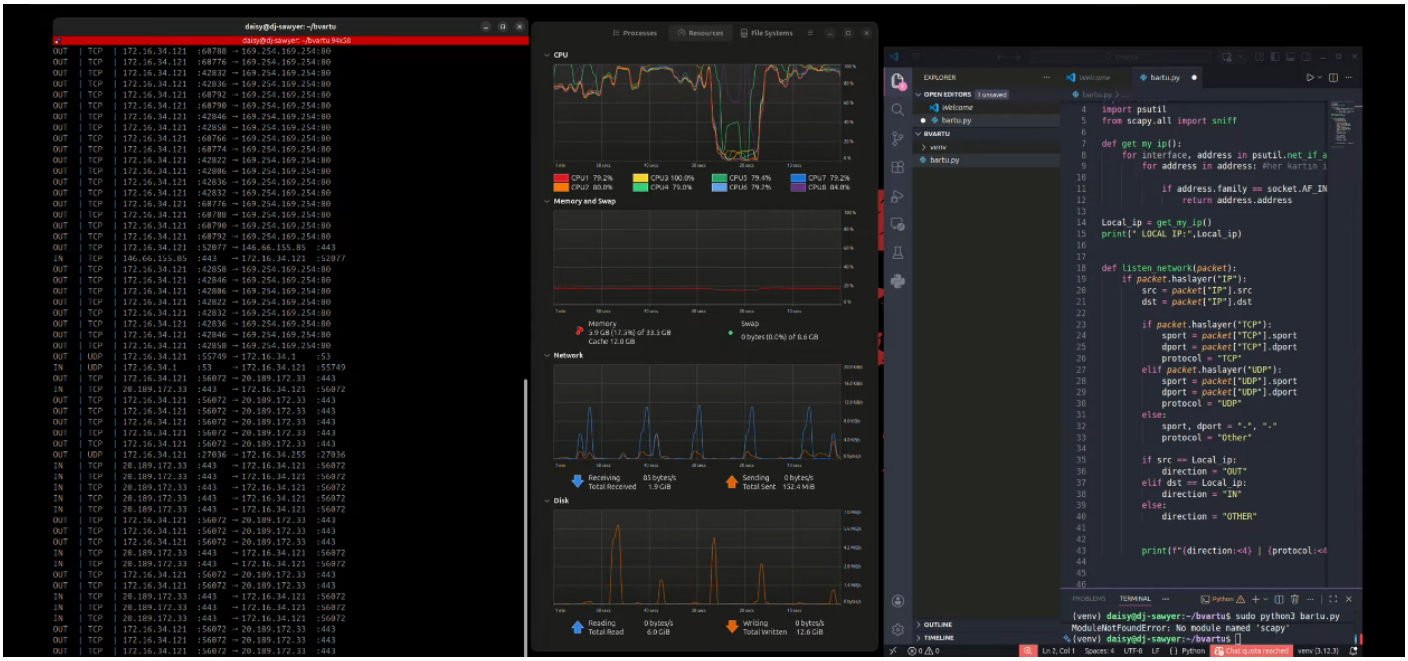
```python
#şimdi ağ kartından geçen veri paketlerini yakalıcaz belkide port number de eklerim ( eklendi hemde protokol sorgusu da yapıyor :/)

import socket
import psutil
from scapy.all import sniff

def get_my_ip():  1 usage
    for interface, address in psutil.net_if_addrs().items(): #bütün ağ kartlarını geziyorum geziyorum
        for address in address: #her kartın içindeki adreslere bakıyyorum

            if address.family == socket.AF_INET and not address.address.startswith("127"): # localhost ve ipv4 olanları atla
                return address.address

Local_ip = get_my_ip()
print(" LOCAL IP:",Local_ip)


def listen_network(packet):  1 usage
    if packet.haslayer("IP"):
        src = packet["IP"].src
        dst = packet["IP"].dst

        if packet.haslayer("TCP"):
            sport = packet["TCP"].sport
            dport = packet["TCP"].dport
            protocol = "TCP"
        elif packet.haslayer("UDP"):
            sport = packet["UDP"].sport
            dport = packet["UDP"].dport
            protocol = "UDP"
        else:
            sport, dport = "-", "-"
            protocol = "Other"


        if src == Local_ip:
            direction = "OUT"
        elif dst == Local_ip:
            direction = "IN"
        else:
            direction = "OTHER"


        print(f"{direction:<4} | {protocol:<4} | {src:<15}:{sport:<5} → {dst:<15}:{dport:<5}")



sniff(prn=listen_network)
```

- Mybe i might turn this code to website, aint wrote some filtering for now ,if its projct get bigeer, for sure i will wrote.

--CPU USSAGE TEST --> I have test it for 2 pc (i5 and i7)

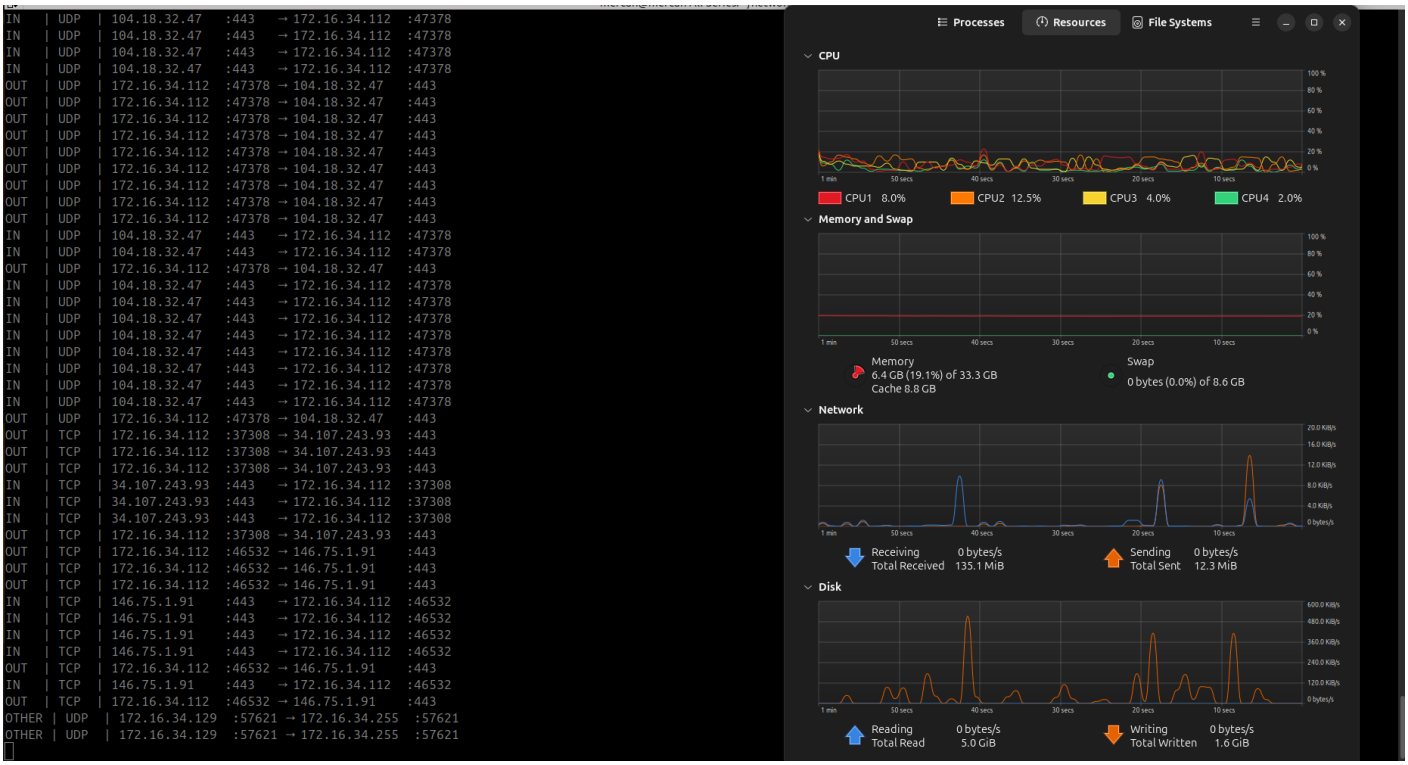
? 7 WHAT ?

I now his is shouldent be that much , its becouse the monitor that connected the pc , its huge . So CPU cant reach it :/



i 5

Okey this is what has to be normally its can be reach %40 but stabilizied %20....



Revision #1
Created 2026-03-21 17:36:31 UTC

Updated 2026-03-25 16:42:59 UTC