main.py

Save Run

```python
import pcapy
import dpkt

def packet_handler(header, data):
  try:
    # Parse the Ethernet frame
    eth = dpkt.ethernet.Ethernet(data)

    # Check if the Ethernet frame contains IP packet
    if isinstance(eth.data, dpkt.ip.IP):
      ip = eth.data

      # Extract source and destination IP addresses
      src_ip = ip.src
      dst_ip = ip.dst

      # Extract protocol
      protocol = ip.data.__class__.__name__

      print(f"Source IP: {src_ip}, Destination IP: {dst_ip}, Protocol:
          {protocol}")

      # Handle different protocols
      if isinstance(ip.data, dpkt.tcp.TCP):
        tcp = ip.data
        payload = tcp.data
```

main.py

Save Run

```
        # Handle different protocols
        if isinstance(ip.data, dpkt.tcp.TCP):
            tcp = ip.data
            payload = tcp.data
            print("TCP Payload:", payload)

        elif isinstance(ip.data, dpkt.udp.UDP):
            udp = ip.data
            payload = udp.data
            print("UDP Payload:", payload)

        elif isinstance(ip.data, dpkt.icmp.ICMP):
            icmp = ip.data
            payload = icmp.data
            print("ICMP Payload:", payload)

    except Exception as e:
        print("Error:", e)

def start_sniffing(interface):
    cap = pcapy.open_live(interface, 65536, True, 100)
    cap.loop(-1, packet_handler)

if __name__ == "__main__":
    interface = "eth0" # Change this to your network interface
    start_sniffing(interface)
```