```
import socket
import struct
import textwrap
```

```
def format_output(data, width=80):
    return '\n'.join([data[i:i+width] for i in range(0, len(data), width)])
```

```
def ethernet_frame(data):
    """Unpacks Ethernet frame"""
    dest_mac, src_mac, proto = struct.unpack('! 6s 6s H', data[:14])
    return get_mac_address(dest_mac), get_mac_address(src_mac), socket.htons(proto), data[14:]
```

```
def get_mac_address(bytes_addr):
    """Formats MAC address"""
    return ':'.join(map('{:02x}'.format, bytes_addr)).upper()
```

```
def ipv4_packet(data):
    """Unpacks IPv4 packet"""
    version_header_length = data[0]
    header_length = (version_header_length & 15) * 4
    ttl, proto, src, target = struct.unpack('! 8x B B 2x 4s 4s', data[:20])
    return header_length, ttl, proto, ipv4(src), ipv4(target), data[header_length:]
```

```
def ipv4(addr):
    """Formats an IPv4 address"""
    return '.'.join(map(str, addr))
```

```
def main():
    # Create a raw socket to capture packets
    conn = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, socket.ntohs(3))
    print("Sniffer is running... Press Ctrl+C to stop.")
```

```
def main():
    # Create a raw socket to capture packets
    conn = socket.socket(socket.AF_PACKET, socket.SOCK_RAW, socket.ntohs(3))
    print("Sniffer is running... Press Ctrl+C to stop.")

    try:
        while True:
            raw_data, addr = conn.recvfrom(65535)
            dest_mac, src_mac, eth_proto, data = ethernet_frame(raw_data)
            print(f'\nEthernet Frame: \nDestination: {dest_mac}, Source: {src_mac}, Protocol: {eth_proto}')

            # Check if the Ethernet frame contains an IPv4 packet
            if eth_proto == 8:  # 8 indicates IPv4
                ip_header_length, ttl, proto, src_ip, target_ip, data = ipv4_packet(data)
                print(f'IPv4 Packet: \nSource: {src_ip}, Target: {target_ip}, Protocol: {proto}')

                # Further packet types (TCP/UDP/ICMP)
                if proto == 1:
                    print("ICMP Packet")
                elif proto == 6:
                    print("TCP Packet")
                elif proto == 17:
                    print("UDP Packet")
                else:
                    print("Other Protocol")
    except KeyboardInterrupt:
        print("\nSniffer stopped.")

if __name__ == "__main__":
    main()
```

```
Streaming output truncated to the last 5000 lines.
Source: 172.28.0.12, Target: 172.28.0.12, Protocol: 6
TCP Packet

Ethernet Frame:
Destination: 00:00:00:00:00:00, Source: 00:00:00:00:00:00, Protocol: 8
```

```
IPv4 Packet:
Source: 172.28.0.12, Target: 172.28.0.12, Protocol: 6
TCP Packet

Ethernet Frame:
Destination: 00:00:00:00:00:00, Source: 00:00:00:00:00:00, Protocol: 8
IPv4 Packet:
Source: 172.28.0.12, Target: 172.28.0.12, Protocol: 6
TCP Packet

Ethernet Frame:
Destination: 00:00:00:00:00:00, Source: 00:00:00:00:00:00, Protocol: 8
IPv4 Packet:
Source: 127.0.0.1, Target: 127.0.0.1, Protocol: 6
TCP Packet

Ethernet Frame:
Destination: 00:00:00:00:00:00, Source: 00:00:00:00:00:00, Protocol: 8
IPv4 Packet:
Source: 127.0.0.1, Target: 127.0.0.1, Protocol: 6
TCP Packet

Ethernet Frame:
Destination: 00:00:00:00:00:00, Source: 00:00:00:00:00:00, Protocol: 8
IPv4 Packet:
Source: 127.0.0.1, Target: 127.0.0.1, Protocol: 6
TCP Packet

Ethernet Frame:
Destination: 00:00:00:00:00:00, Source: 00:00:00:00:00:00, Protocol: 8
IPv4 Packet:
Source: 127.0.0.1, Target: 127.0.0.1, Protocol: 6
TCP Packet

Ethernet Frame:
Destination: 00:00:00:00:00:00, Source: 00:00:00:00:00:00, Protocol: 8
IPv4 Packet:
Source: 127.0.0.1, Target: 127.0.0.1, Protocol: 6
TCP Packet

Ethernet Frame:
Destination: 00:00:00:00:00:00, Source: 00:00:00:00:00:00, Protocol: 8
IPv4 Packet:
Source: 127.0.0.1, Target: 127.0.0.1, Protocol: 6
TCP Packet

Ethernet Frame:
Destination: 00:00:00:00:00:00, Source: 00:00:00:00:00:00, Protocol: 8
IPv4 Packet:
Source: 127.0.0.1, Target: 127.0.0.1, Protocol: 6
TCP Packet
```