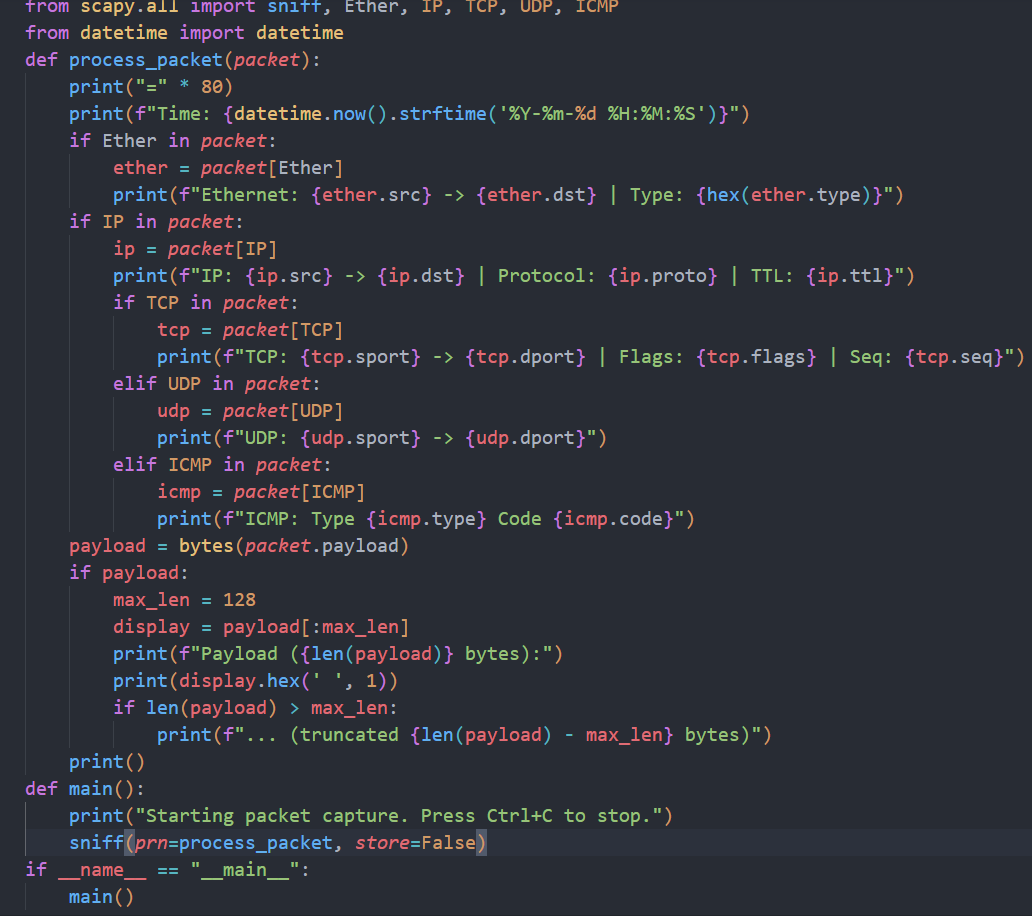
**Aim**

The primary aim of this script is to **capture network packets** traversing a local interface in real-time and **decode key header information** from the Ethernet, IP, TCP, UDP, and ICMP layers. It functions as a basic network sniffer and protocol analyzer, printing details like source/destination MAC and IP addresses, port numbers, protocol types, and a hexadecimal dump of the packet payload.

**Algorithm**

1. **Import necessary libraries**: Import Scapy functions (sniff, Ether, IP, TCP, UDP, ICMP) for packet capture/analysis and datetime for timestamping.
2. **Define process\_packet function**:
   * Print a time-stamped separator for each new packet.
   * **Check for Ethernet layer**: Print source (src) and destination (dst) MAC addresses and the EtherType.
   * **Check for IP layer**: Print src/dst IP addresses, protocol number, and Time To Live (TTL).
   * **Check for TCP layer (nested in IP)**: Print src/dst ports, flags, and sequence number.
   * **Check for UDP layer (nested in IP)**: Print src/dst ports.
   * **Check for ICMP layer (nested in IP)**: Print message type and code.
   * **Check for Payload**: Convert the raw payload bytes to a hexadecimal string, print the length, and display up to the first 128 bytes.
3. **Define main function**:
   * Start the capture process using sniff().
   * Set the prn (print function) argument to call process\_packet for every captured packet.
   * Set store=False to process packets in real-time without storing them in memory.
4. **Execute main**: Run the main function when the script is executed.

**Source Code**

****

**Result (Sample Output)**

This is a simulated output showing how the script would display two different captured packets (e.g., a **DNS query via UDP** and an **HTTP or HTTPS session via TCP**).

