How to Capture Network Packets on Linux Using tcpdump

Capturing network packets is an essential task for network administrators and systems engineers. It allows you to monitor and analyze network traffic, diagnose network issues, and enhance security by detecting suspicious activities. In the Linux environment, this task can be efficiently performed using tools like tcpdump, Wireshark, and others. This article will focus on tcpdump, a powerful command-line packet analyzer.

Examples:

1. **Basic Packet Capture with tcpdump**

   To capture packets on a specific network interface, you can use the following command:

   ```
   sudo tcpdump -i eth0
   ```

   This command captures all packets on the eth0 interface. You can replace eth0 with the name of your network interface.

2. **Capture Packets and Save to a File**

   If you want to save the captured packets to a file for later analysis, you can use the -w option:

   ```
   sudo tcpdump -i eth0 -w capture.pcap
   ```

   This command captures packets on the eth0 interface and saves them to a file named capture.pcap.

3. **Read Captured Packets from a File**

   To read packets from a previously saved file, use the -r option:

   ```
   sudo tcpdump -r capture.pcap
   ```

   This command reads and displays the packets from the capture.pcap file.

4. **Filter Packets by IP Address**

You can filter packets by IP address using the following command:

```
sudo tcpdump -i eth0 host 192.168.1.1
```

This command captures only the packets to and from the IP address 192.168.1.1 on the eth0 interface.

5. **Filter Packets by Port**

   To capture packets on a specific port, use the port filter:

   ```
   sudo tcpdump -i eth0 port 80
   ```

   This command captures packets on port 80 (HTTP) on the eth0 interface.

6. **Verbose Output**

   For more detailed output, you can use the -v, -vv, or -vvv options:

   ```
   sudo tcpdump -i eth0 -vv
   ```

   This command provides a more verbose output for the captured packets.