**Experiment No. 7**

**Aim:** Study of packet sniffer tools Wireshark:

1. Observer performance in promiscuous as well as non-promiscuous mode.
2. Show the packets can be traced based on different filters

**Learning Objectives:**

The primary objective of this experiment is to introduce students to packet sniffer tools, focusing on Wireshark, and help them understand how these tools can be used to monitor network traffic. The specific learning objectives are as follows:

To comprehend the concept of packet sniffing and the different modes (promiscuous and non-promiscuous).

To learn how to use Wireshark to capture and analyse network packets.

To explore packet filtering techniques in Wireshark to trace specific types of network traffic.

**Related Theory:**

**What is Wireshark?**

Wireshark is a popular and widely-used open-source network packet analyzer and protocol analyzer. It is a software tool that allows users to capture, view, and analyze the data packets that are transmitted over a computer network. Wireshark is available for various operating systems, including Windows, macOS, and Linux.

**Key features and functions of Wireshark include:**

1. Packet Capture: Wireshark can capture network traffic in real-time from various network interfaces, including wired and wireless connections. It allows users to select the network interface to monitor.
2. Packet Analysis: It provides a detailed view of captured packets, including information about their source and destination, packet type, protocols used, and the content of the data payload.
3. Powerful Filters: Wireshark offers extensive filtering capabilities, allowing users to focus on specific types of packets based on various criteria like IP addresses, port numbers, protocols, and more. This makes it a valuable tool for network troubleshooting and security analysis.
4. Protocol Support: Wireshark supports a wide range of network protocols, including Ethernet, Wi-Fi, TCP/IP, HTTP, HTTPS, DNS, FTP, and many others. It can decode and display the structure of these protocols, making it easier to understand network communication.
5. Packet Decoding: Wireshark can decode and display the content of data packets, helping network administrators and security professionals identify potential issues, errors, or security threats.
6. Packet Playback: Users can replay captured packets to analyze network behaviors and test network configurations.
7. Export and Save: It allows users to save captured packets to files for later analysis or sharing with others.

**Wireshark is widely used for various purposes, including:**

* Network Troubleshooting: It is a valuable tool for diagnosing network problems, identifying performance issues, and pinpointing the source of network errors.
* Network Security Analysis: Security professionals use Wireshark to monitor network traffic for suspicious or malicious activity. It helps in detecting intrusion attempts and analyzing vulnerabilities.
* Protocol Development and Testing: Wireshark is used by developers to test and debug network protocols and applications.
* Network Monitoring and Optimization: Network administrators use Wireshark to monitor and optimize network performance, ensuring efficient data transmission.

**1. Observer performance in promiscuous as well as non-promiscuous mode.**

Promiscuous mode captures all network traffic on a segment, while non-promiscuous mode captures only packets specifically addressed to the device.

**Procedure:**

1. **Capture in Promiscuous Mode:** a. Launch Wireshark and select the network interface to monitor. b. Start capturing network traffic in promiscuous mode. c. Analyze the captured packets to observe the types and volume of traffic visible in this mode. d. Note any security or privacy implications associated with promiscuous mode.
2. **Capture in Non-Promiscuous Mode:** a. Close the previous capture session and start a new one, but this time, select non-promiscuous mode. b. Begin capturing network traffic. c. Analyze the packets captured in non-promiscuous mode and compare them to what was observed in promiscuous mode. d. Note the differences in captured traffic and discuss why some packets may not be visible in non-promiscuous mode.
3. **Observer Performance Evaluation:** a. Compare the two capture sessions to determine which mode provided a more comprehensive view of the network traffic. b. Discuss the advantages and disadvantages of each mode in terms of network monitoring and security. c. Consider the ethical and legal implications of using promiscuous mode for network monitoring.
4. **Security and Privacy Considerations:** a. Discuss the potential security risks and privacy concerns associated with promiscuous mode. How might unauthorized packet capture impact network security? b. Analyze whether using promiscuous mode for monitoring could violate privacy regulations or policies.

**2. Show the packets can be traced based on different filters**

Wireshark provides a powerful filtering mechanism that allows users to focus on specific packets based on criteria such as source/destination IP addresses, protocols, port numbers, and more. This filtering capability is essential for network administrators and security professionals to identify and troubleshoot network issues and detect malicious activity.

**Procedure:**

1. **Capture Network Traffic:** a. Start Wireshark and select the network interface you want to capture traffic from. b. Begin capturing network traffic.
2. **Apply Basic Filters:** a. Instruct students to apply basic filters to narrow down the captured packets. For example, they can filter by:
   * Source and destination IP addresses
   * Port numbers
   * Protocol (e.g., HTTP, DNS) b. Have students analyze the filtered packets and discuss the specific traffic they can observe.
3. **Advanced Filtering:** a. Introduce more advanced filtering techniques, such as: ● Logical combinations (AND, OR, NOT) of filters.
   * Display filters using expressions (e.g., "ip.addr==192.168.1.1 && dns"). b. Let students experiment with these advanced filters and observe the results.
4. **Filter by Packet Type:** a. Encourage students to filter packets by type, such as:
   * ICMP (ping) packets
   * ARP (Address Resolution Protocol) packets
   * TCP or UDP packets b. Discuss the significance of each packet type in network communication.
5. **Filter by Packet Content:** a. Show students how to filter packets based on their content. For example, they can search for specific keywords or patterns within packet payloads. b. Explain the use of regular expressions for more complex content filtering.
6. **Custom Filters:** a. Have students create their own custom filters based on specific criteria relevant to a given network scenario or problem. b. Discuss how custom filters can be valuable for network troubleshooting and security.

**Leaning Outcome:**

By the end of this experiment, students should be able to:

* Understand the concept of packet sniffing and the differences between promiscuous and non-promiscuous modes.
* Use Wireshark to capture network packets in both modes.
* Apply different filters in Wireshark to isolate and analyze specific packets based on various criteria.

**Implementation:**







