Experiment 08

Study of packet sniffer tools wireshark :-
a. Observer performance in promiscuous as well as non-promiscuous mode.
b. Show the packets can be traced based on different filters.

Roll No.	19
Name	Manav Jawrani
Class	D15-A
Subject	Security Lab
LO Mapped	LO3: Explore the different network reconnaissance tools to gather information about networks

Aim: Study of packet sniffer tools wireshark :-
a. Observer performance in promiscuous as well as non-promiscuous mode.
b. Show the packets can be traced based on different filters.

Introduction:

Wireshark:
Wireshark is the world's leading network traffic analyser, and an essential tool for any security professional or systems administrator. This free software lets you analyze network traffic in real time, and is often the best tool for troubleshooting issues on your network.
Wireshark intercepts traffic and converts that binary traffic into human-readable format. This makes it easy to identify what traffic is crossing your network, how much of it, how frequently, how much latency there is between certain hops, and so forth. While Wireshark supports more than two thousand network protocols, many of them esoteric, uncommon, or old, the modern security professional will find analyzing IP packets to be of most immediate usefulness. The majority of the packets on your network are likely to be TCP, UDP, and ICMP.
Packet sniffer is a basic tool for observing network packet exchanges in a computer. As the name suggests, a packet sniffer captures ("sniffs") packets being sent/received from/by your computer; it will also typically store and/or display the contents of the various protocol fields in these captured packets. A packet sniffer itself is passive. It observes messages being sent and received by applications and protocols running on your computer, but never sends packets itself.
Wireshark is a free open-source network protocol analyser. It is used for network troubleshooting and communication protocol analysis. Wireshark captures network packets in real time and displays them in human-readable format. It provides many advanced features including live capture and offline analysis, three-pane packet browser, coloring rules for analysis. This document uses Wireshark for the experiments, and it covers Wireshark installation, packet capturing, and protocol analysis.

Implementation:

Capturing and Filtering the Packets:

Step 1:

Download and install the Wireshark software. Go to http://www.wireshark.org/download.html and download and install the Wireshark for your computer.

Step 2:

Wireshark GUI: When you first open Wireshark, you'll be presented with the start screen. There are four primary areas to the start screen, some of which will carry over into the working screen once you pick an interface to work to capture traffic from.

Primary Areas of the Wireshark Start Screen :-

The Menu

The Main Toolbar

The Filter Toolbar

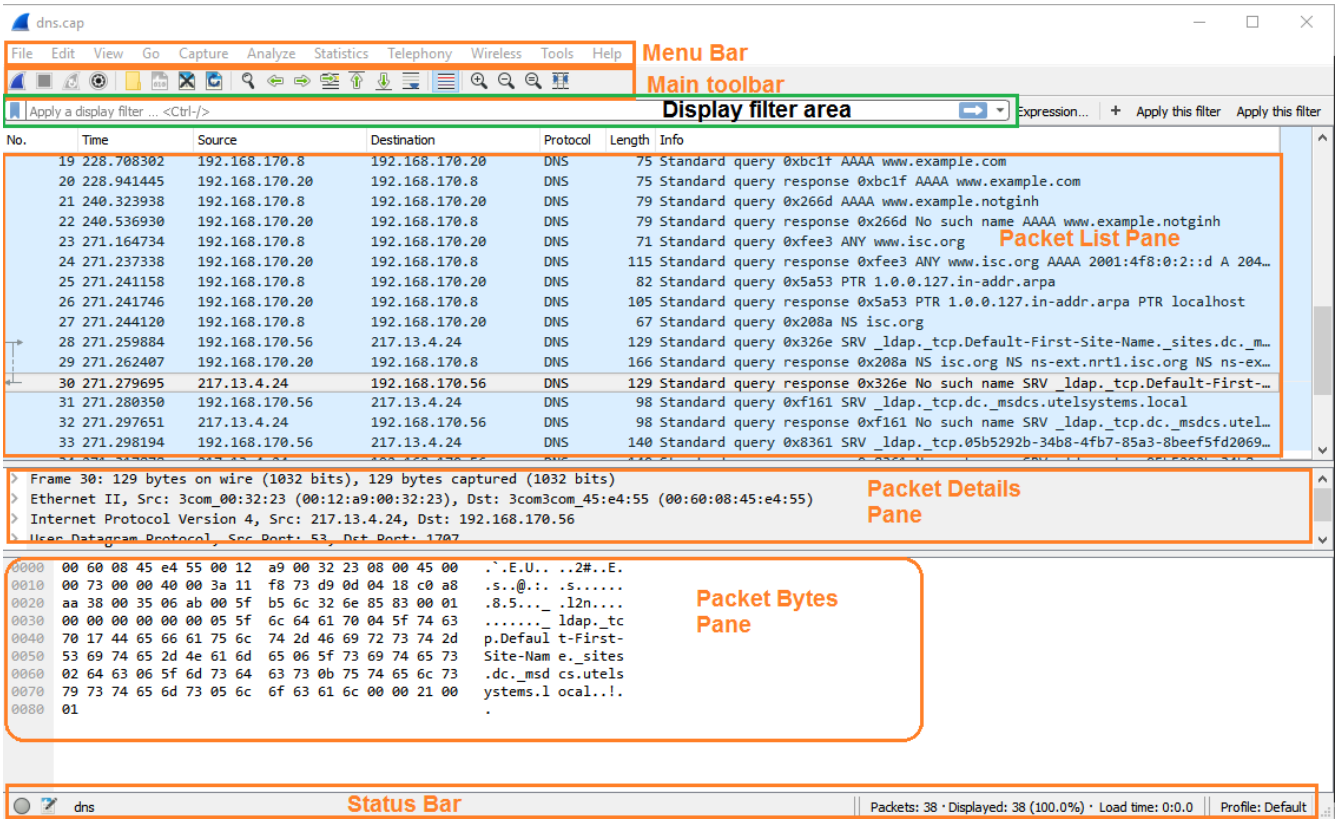
The Interface List

"The Menu"

Wireshark Main Toolbar: This is a quick access toolbar providing easy to use buttons for the most common functions of the main menu. Most of these buttons become active only after you've selected an interface to monitor.

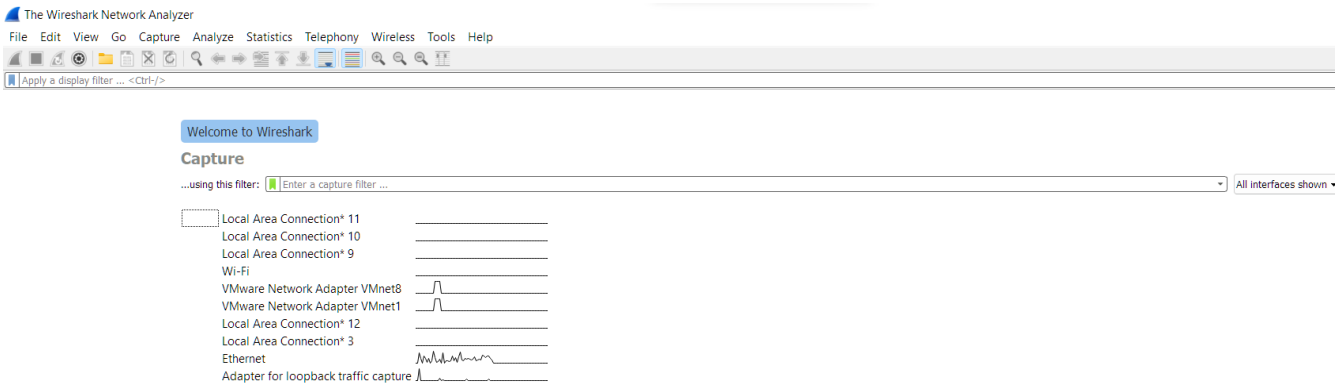
Wireshark Filter Toolbar: This toolbar allows you to quickly edit and apply display filters to your capture. Display filters allow you to narrow down the packets that you've captured to only those that are relevant to what you're trying to see such as specific IP address sources and destinations, protocols, MAC addresses, etc…

Wireshark Interface List: The Interface List is the area where the interfaces that your device has installed will appear. Before you can see packet data you need to pick one of the interfaces by clicking on it. You can choose a capture filter and type of interface to show in the interfaces lists at this screen as well. Clicking on and interface or opening an existing capture file will take you to the working screen



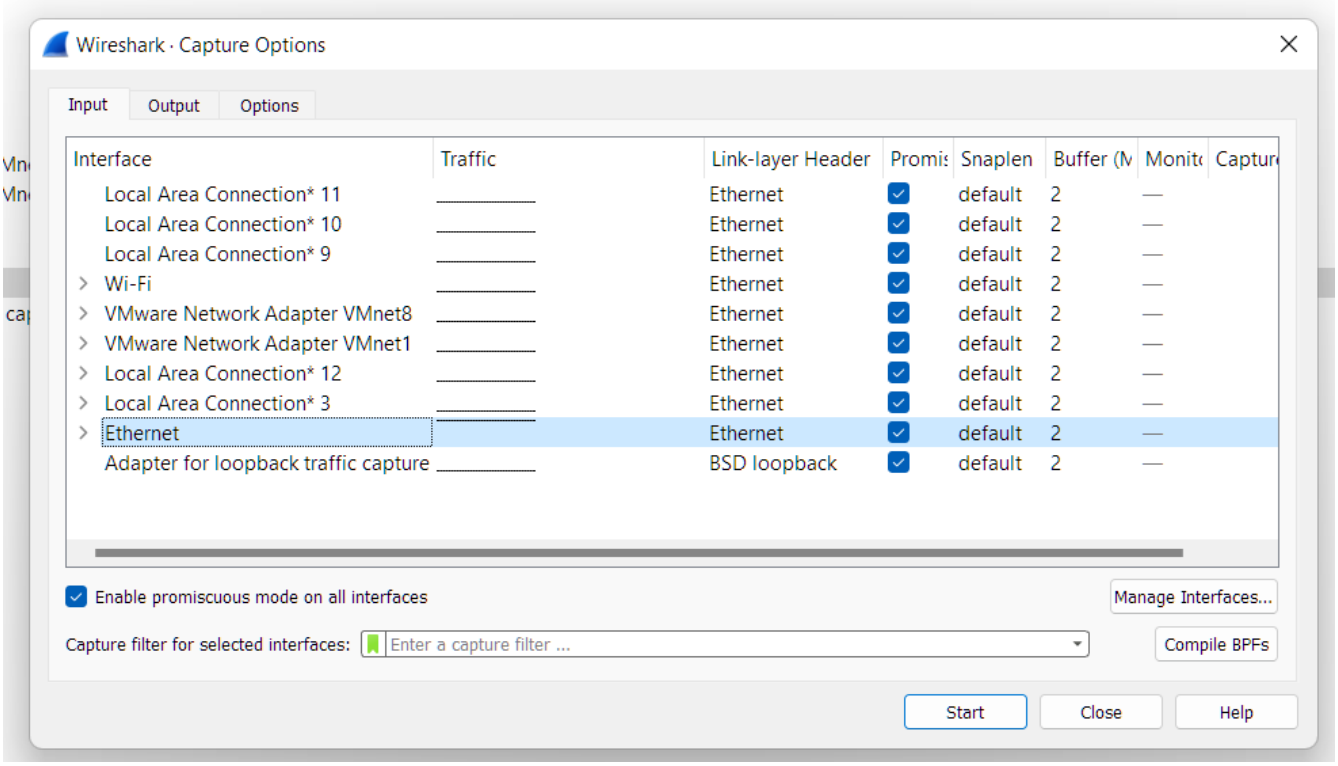
Step 3:

Start Wireshark: When you run the Wireshark program, the Wireshark graphic user interface will be seen as shown below.



Then, you need to choose an interface. If you are running the Wireshark on your laptop, you need to select the Ethernet interface. If you are at a desktop, you need to select the Ethernet interface being used. Note that there could be multiple interfaces. In general,

you can select any interface but that does not mean that traffic will flow through that interface. The network interfaces (i.e., the physical connections) that your computer has to the network are shown. After you select the interface, you can click start to capture the packets as shown in below:

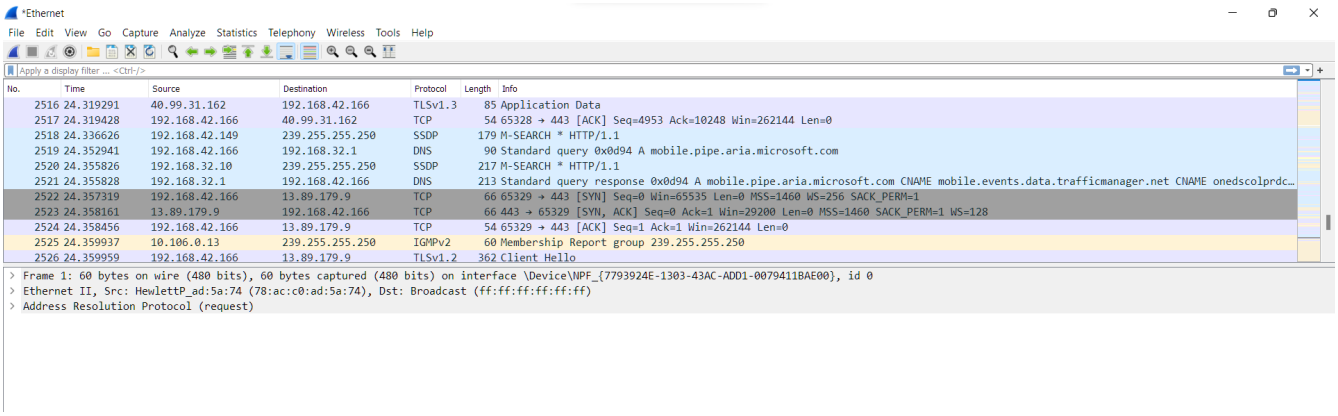


Step 4:

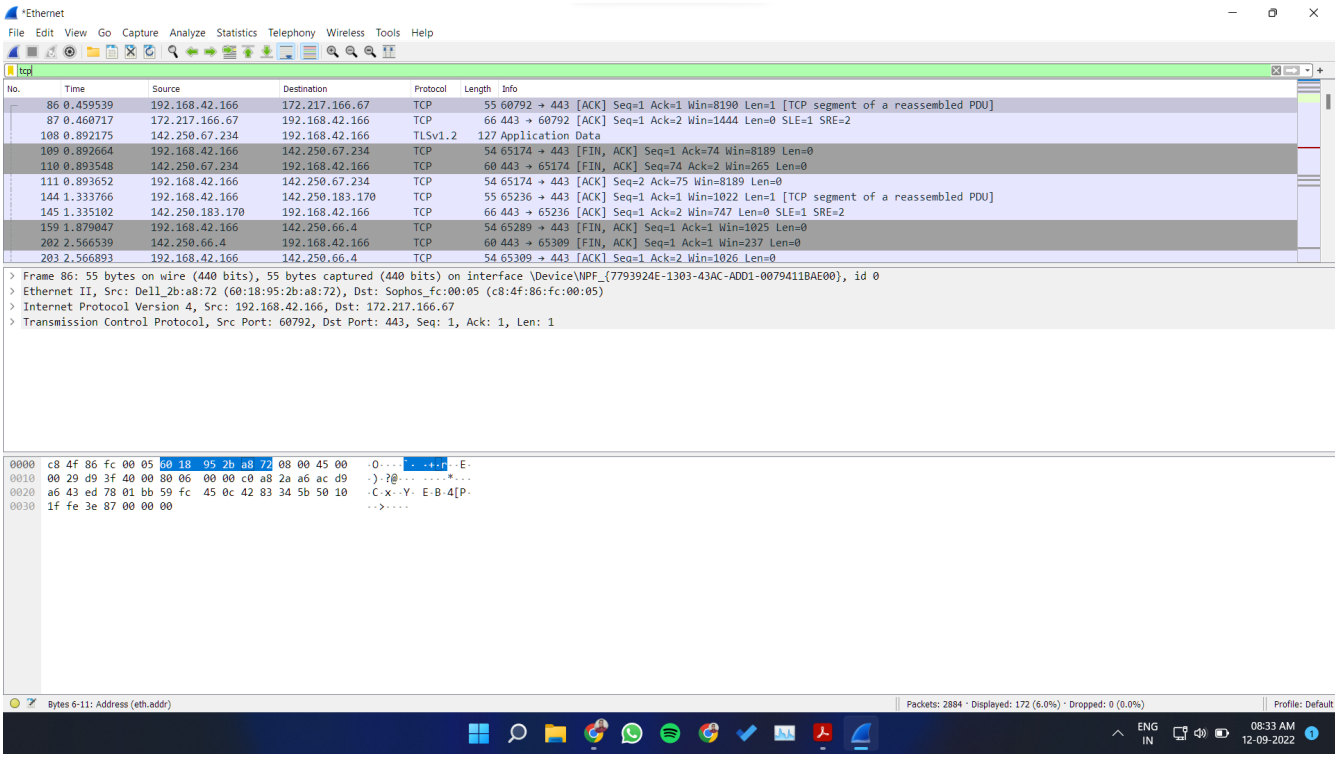
In your browser, go to any website. After your browser has displayed that website stop Wireshark packet capture by selecting stop in the Wireshark capture window. This will cause the Wireshark capture window to disappear and the main Wireshark window to display all packets captured since you began packet capture see image below:

display all packets captured since you began packet capture see image below:

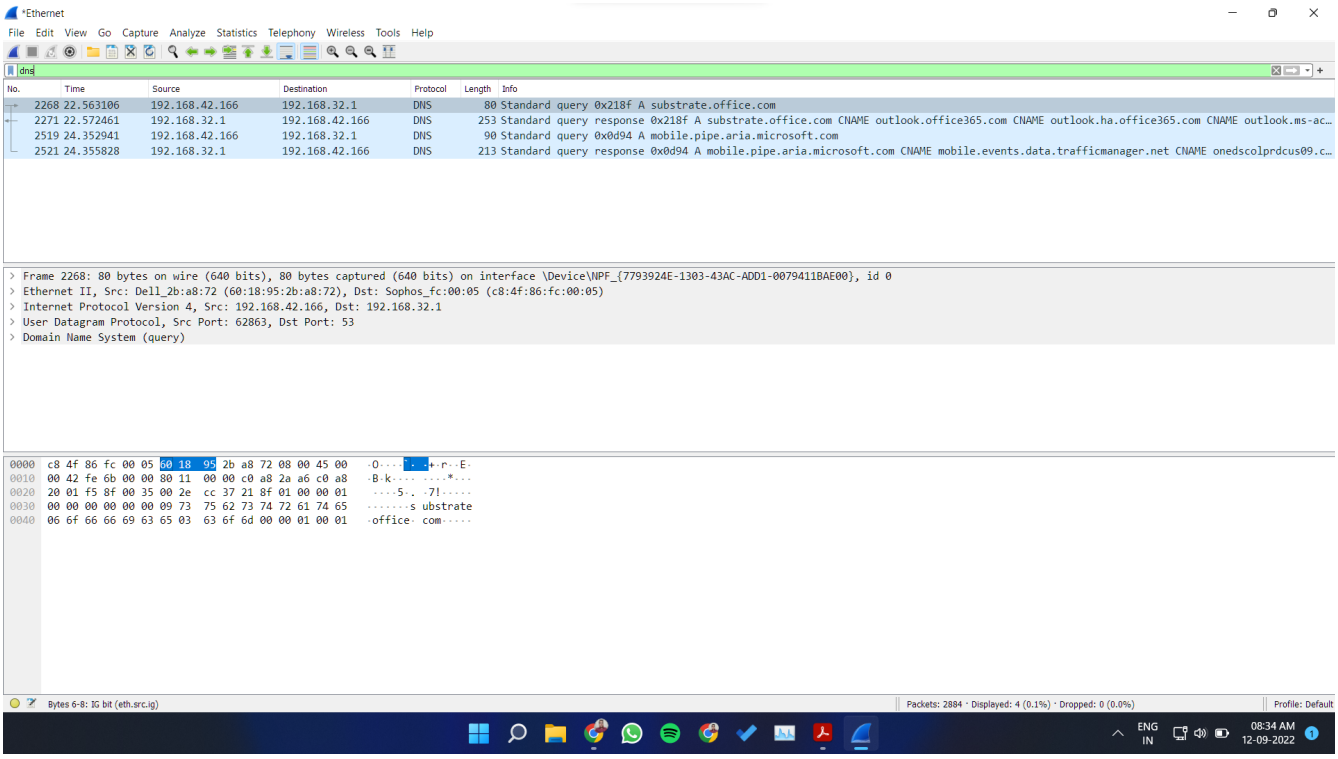
(NOTE: You'll probably see packets highlighted in green, blue, and black. Wireshark uses colors to help you identify the types of traffic at a glance. By default, green is TCP traffic, dark blue is DNS traffic, light blue is UDP traffic, and black identifies TCP packets with problems — for example, they could have been delivered out-of-order.)



You now have live packet data that contains all protocol messages exchanged between your computer and other network entities! However, as you will notice the HTTP/DNS/TCP/UDP messages are not clearly shown because there are many other packets included in the packet capture. Even though the only action you took was to open your browser, there are many other programs in your computer that communicate via the network in the background. To filter the connections to the ones we want to focus on, we have to use the filtering functionality of Wireshark by typing “http” in the filtering field as shown below:

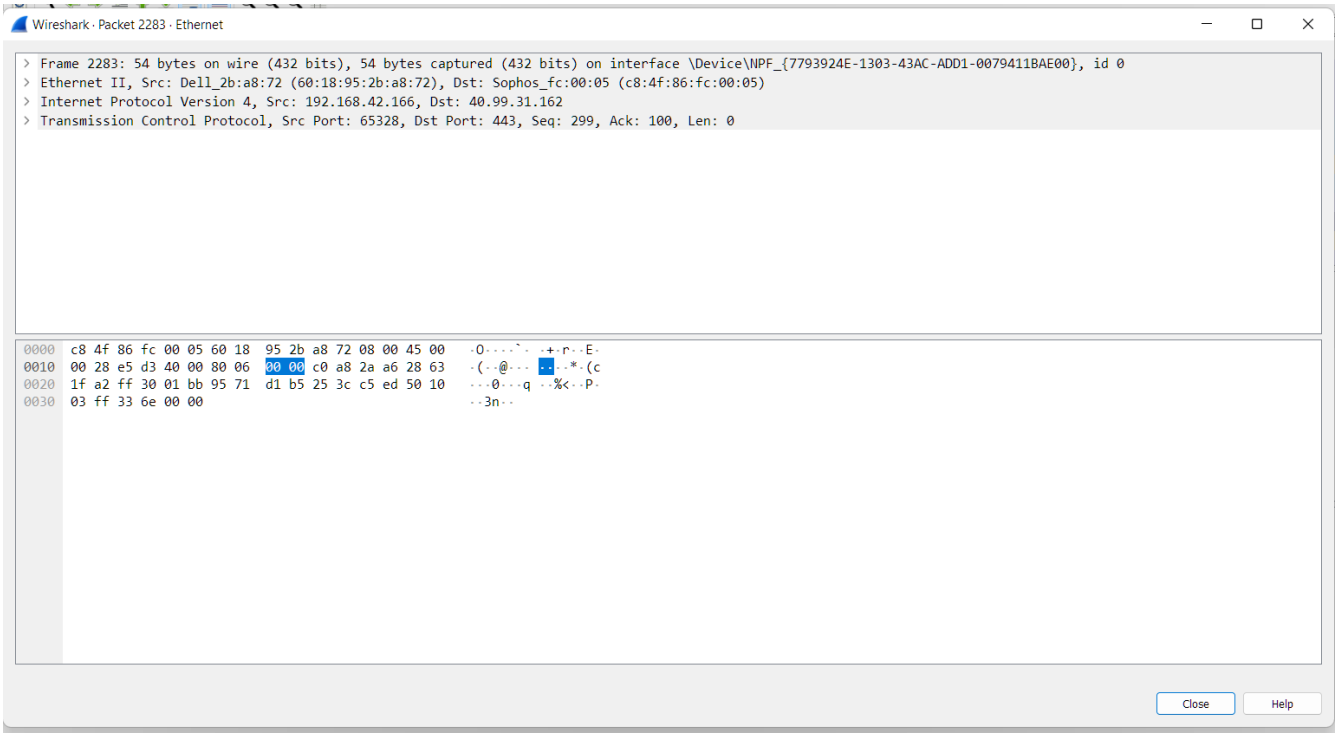


Now, we can try another protocol. Let's use Domain Name System (DNS) protocol as an example here:



Step 5:

By clicking on the particular packet or trace, the packet details can be obtained as follows:



Conclusion:

Thus, we have Observed performance in promiscuous as well as non promiscuous mode and also we have seen that the packets can be traced based on different filters.