LAB Manual
PART A
(PART A : TO BE REFFERED BY STUDENTS)

Experiment No. 5

A.1 Aim:
To install network sniffer (Wireshark) , analyze the working of packet sniffing and study other features of Wireshark

A.2 Prerequisite:
Role of packet sniffers in network security

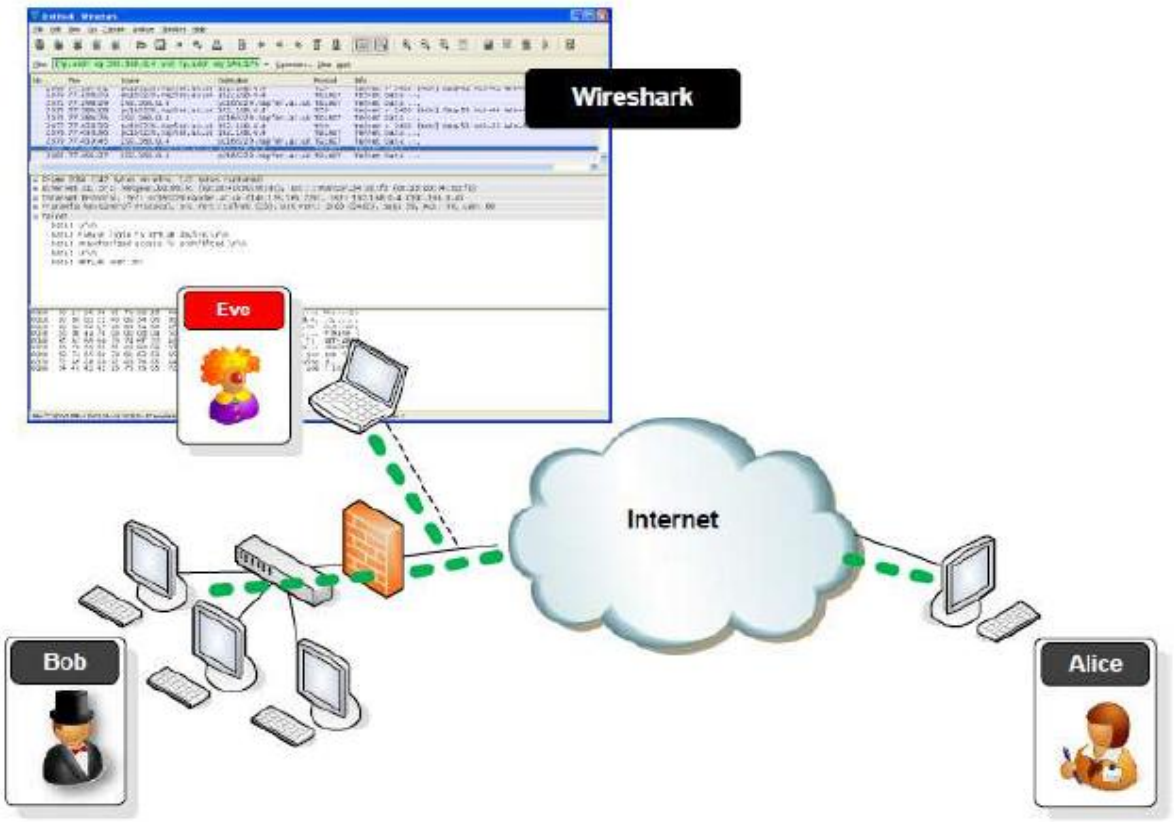
A.3 Outcome:
After successful completion of this experiment students will be able to
1. Appreciate Wireshark as a tool to analyze the packets travelling in a network.
2. Know how this tool can be used by malicious intruders to capture and analyze network traffic.

A.4 Theory:

A packet sniffer, sometimes referred to as a network monitor or network analyzer, can be used by a network or system administrator to monitor and troubleshoot network traffic. Using the information captured by the packet sniffer an administrator can identify erroneous packets and use the data to pinpoint bottlenecks and help maintain efficient network data transmission. In its simple form a packet sniffer simply captures all of the packets of data that pass through a given network interface. By placing a packet sniffer on a network in promiscuous mode, a malicious intruder can capture and analyze all of the network traffic.

Wireshark is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible.

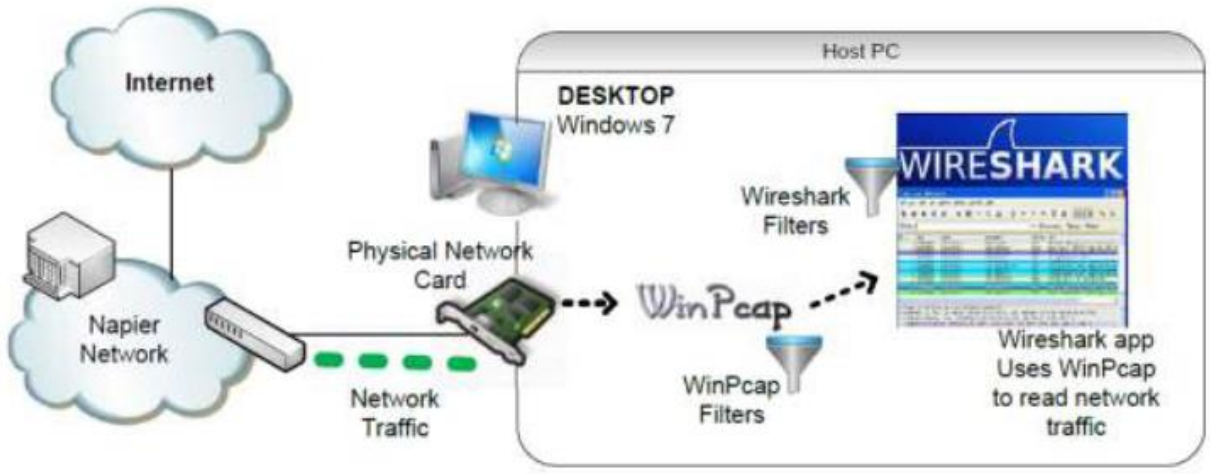
Packet Capture (Packet Sniffing)
A **packet sniffer** is an application which can capture and analyze network traffic which is passing through a system's Network Interface Card (NIC). The sniffer sets the card to **promiscuous mode** which means all traffic is read, whether it is addressed to that machine or not. The figure below shows an attacker sniffing packets from the network, and the **Wireshark** packet sniffer/analyzer (formerly known as ethereal).



Packet Analysis
Wireshark is an open source cross-platform packet capture and analysis tool, with versions for Windows and Linux. The GUI window gives a detailed breakdown of the network protocol stack for each packet, colorizing packet details based on protocol, as well as having functionality to filter and search the traffic, and pick out TCP streams. Wireshark can also save packet data to files for offline analysis and export/import packet captures to/from other tools. Statistics can also be generated for packet capture files.
Wireshark can be used for **network troubleshooting**, to **investigate security issues**, and to **analyze and understand network protocols**. The packet sniffer can exploit information passed in plaintext, i.e. not encrypted. Examples of **protocols** which pass information in plaintext are **Telnet, FTP, SNMP, POP, and HTTP**.

Wireshark is a GUI based network capture tool. There is a command line based version of the packet capture utility, called **TShark**. TShark provides many of the same features as it's big brother, but is console-based. It can be a good alternative if only command line access is available, and also uses less resources as it has no GUI to generate.

Using Wireshark to Capture Traffic



Select a Network Interface to Capture Packets through.

Start the Wireshark application. When Wireshark is first run, a default, or blank window is shown. To list the available network interfaces, select the **Capture->Interfaces** menu option.

Wireshark should display a popup window such as the one shown in Figure 2. To capture network traffic clicks the **Start** button for the network interface you want to capture traffic on. Windows can have a long list of virtual interfaces, before the Ethernet Network Interface Card (NIC).

Figure 2 - Wireshark Interfaces Window

Generate some network traffic with a Web Browser, such as Internet Explorer or Chrome. Your Wireshark window should show the packets, and now look something like.

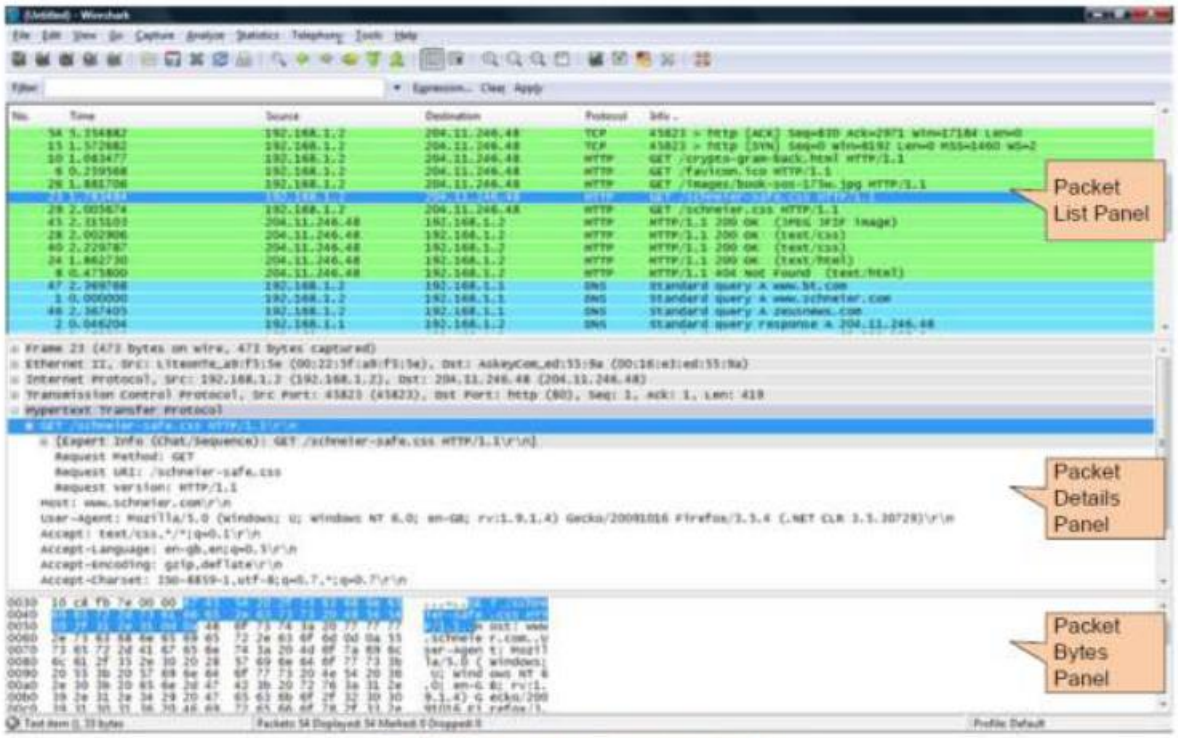


Figure 3 - Wireshark capuring traffic

To stop the capture, select the **Capture->Stop** menu option, Ctrl+E, or the Stop toolbar button. What you have created is a Packet Capture or ***„pcap'***, which you can now view and analyze using
the Wireshark interface or save to disk to analyze later.
The capture is split into 3 parts:
1. **Packet List Panel –** this is a list of packets in the current capture. It colors the packets based on the protocol type. When a packet is selected, the details are shown in the two panels below.
2. **Packet Details Panel –** this shows the details of the selected packet. It shows the different protocols making up the layers of data for this packet. Layers include Frame, Ethernet, IP, TCP/UDP/ICMP, and application protocols such as HTTP.
3. **Packet Bytes Panel –** shows the packet bytes in Hex and ASCII encodings.
To select more detailed options when starting a capture, select the **Capture->Options** menu option, or **Ctrl+K**, or the Capture Options button on the toolbar (the wrench). This should show a window such as shown in Figure 4.
Some of the more interesting options are:
 ***Capture Options > Interface* -** Again the important thing is to select the correct Network Interface to capture traffic through.
 ***Capture Options > Capture File* –** useful to save a file of the packet capture in real time, in case of a system crash.
 ***Display Options > Update list of packets in real time* –** A display option, which should be checked if you want to view the capture as it happens (typically switched off to capture straight to a file, for later analysis).
 ***Name Resolution > MAC name resolution* –** resolves the first 3 bytes of the MAC Address, the Organization Unique Identifier (OUI), which represents the Manufacturer of the Card.

 Name Resolution > Network name resolution – does a DNS lookup for the IP Addresses captured, to display the network name. Set to off by default, so covert scans do not generate this DNS traffic, and tip off who's packets you are sniffing.
Make sure the **MAC name resolution** is selected. Start the capture, and generate some Web traffic again, then stop the capture.

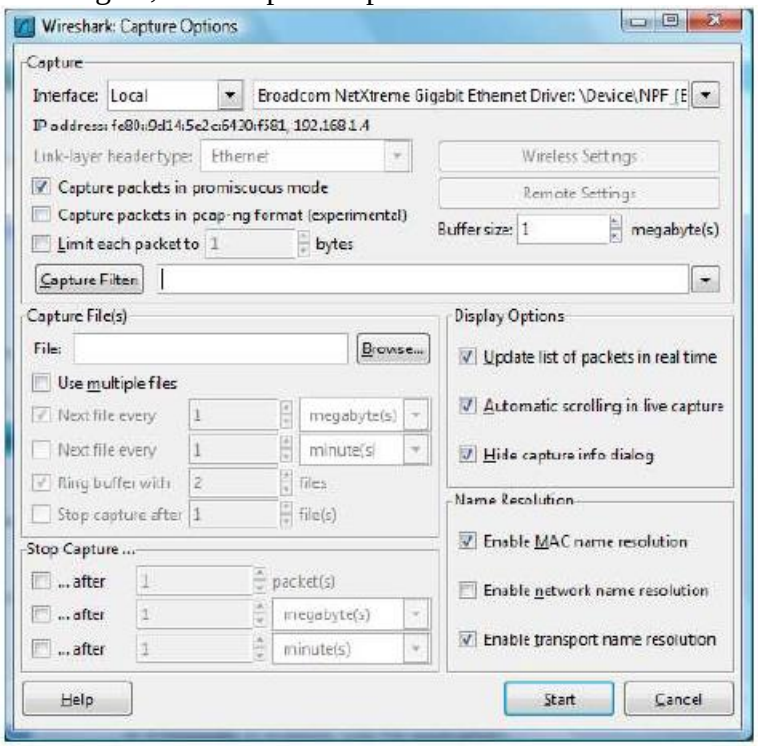


Figure 4 - Wireshark Capture Options

PART B

(PART B : TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

Roll. No. A022	Name: Kartik Padave
Class: B.Tech CSBS	Batch: 1
Date of Experiment:	Date of Submission:
Grade:	

B.1 Software installation issues faced:

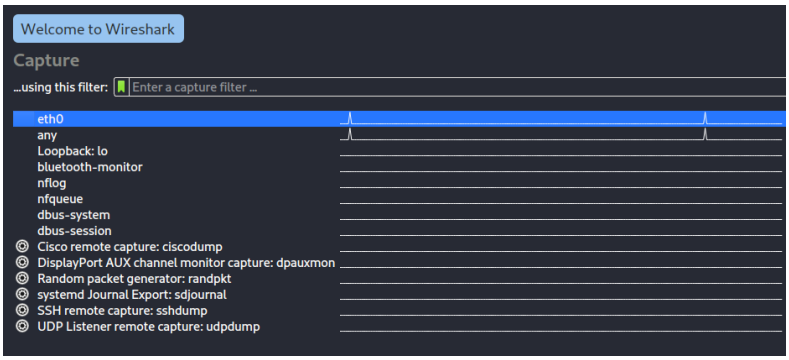
Steps to install Wireshark in Kali Linux:

1. Run command: $ sudo apt update
   This command updates APT before installing any new software.
2. Install wireshark: $ sudo apt-get install wireshark
   Installation will begin and will ask for permission type ‘y’ and hit enter.
3. To confirm installation:
   Type: $ wireshark -h

B.2 Input and Output:

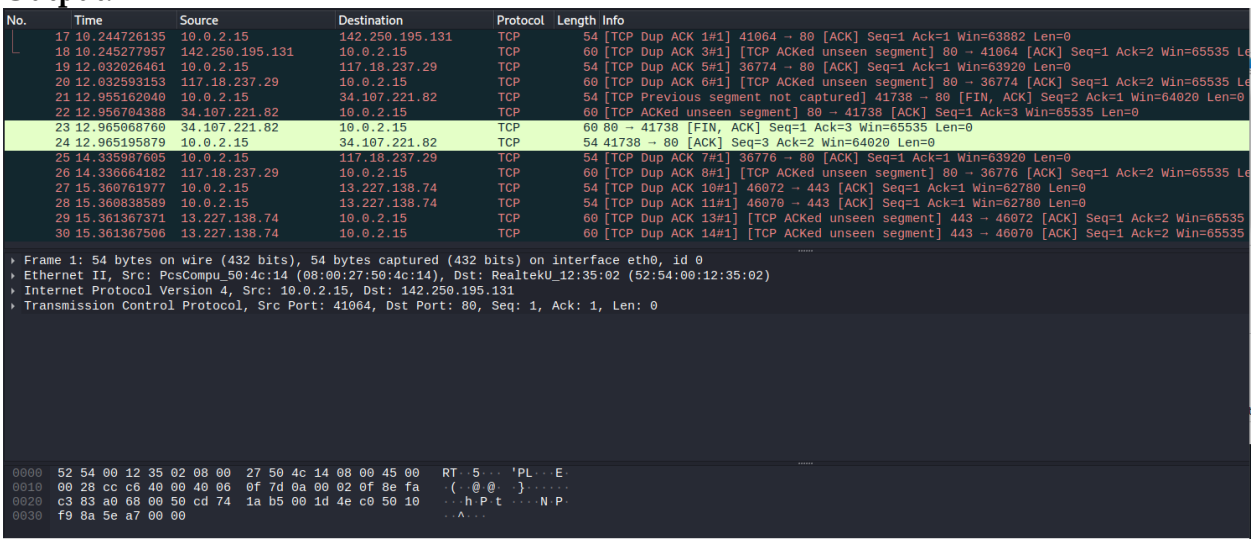
(Paste your program input and output in following format, If there is error then paste the specific error in the output part. In case of error with due permission of the faculty extension can be given to submit the error free code with output in due course of time. Students will be graded accordingly.)

Input:



Wireshark tool on startup.

Output:



Stats of eth0. This shows transfer of packets between the device, network and destination.

B.3 Observations and learning:

(Students are expected to comment on the output obtained with clear observations and learning for each task/ sub part assigned)

From the above experiment, we can observe that device's WiFi passes some default packets to the device even when there is nothing else using the WiFi. When we open an online service, we see some waves of network in Traffic Capture.

B.4 Conclusion:

(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)

We were able to observe the working of Wireshark Tool when there is only wifi working and one when we use the browser, successfully.

Questions of Curiosity

(To be answered by student based on the practical performed and learning/observations)

Q1: Give the uses of Wireshark tool

1. Packet Capture: Wireshark listens to a network connection in real time and then grabs entire streams of traffic – quite possibly tens of thousands of packets at a time.
2. Filtering: Wireshark is capable of slicing and dicing all of this random live data using filters. By applying a filter, you can obtain just the information you need to see.
3. Visualization: Wireshark, like any good packet sniffer, allows you to dive right into the very middle of a network packet. It also allows you to visualize entire conversations and network streams.

Q2: List some other packet sniffing tools.

1. Tcpdump
2. Cloudshark
3. Interceptor-ng
4. Nethogs
5. Ethercap