Name: Priyush B. Khobragade

PRN: 211112018

Batch: 03

EXPERMINT: 09

●Aim: -RIP protocol graphical simulation using packet trace.

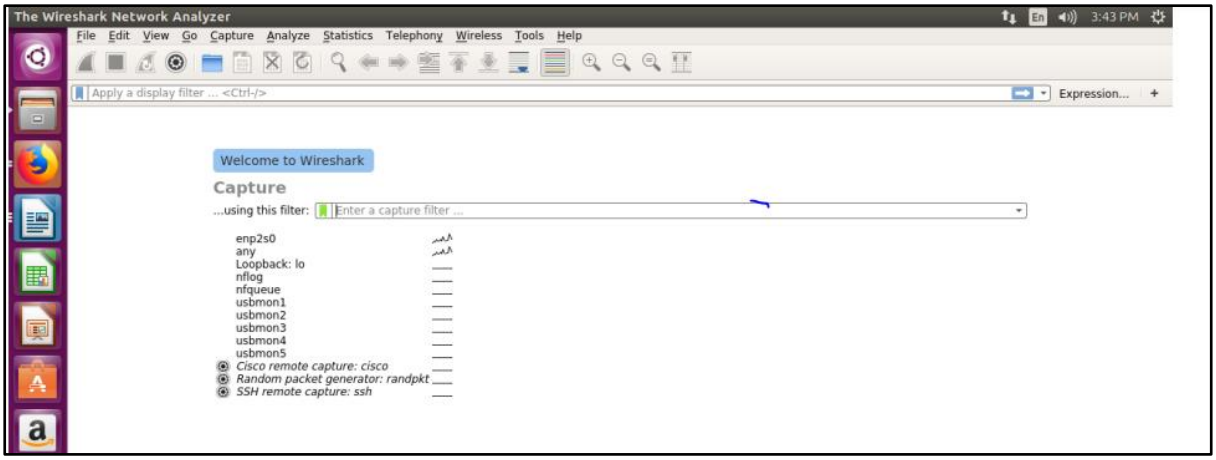
●Theory:

Capturing Packets:

when you start a Wireshark without opening a capture file or starting a capture process, a welcome screen is displayed.

This window will always display currently opened capture files and the capture available interfaces.

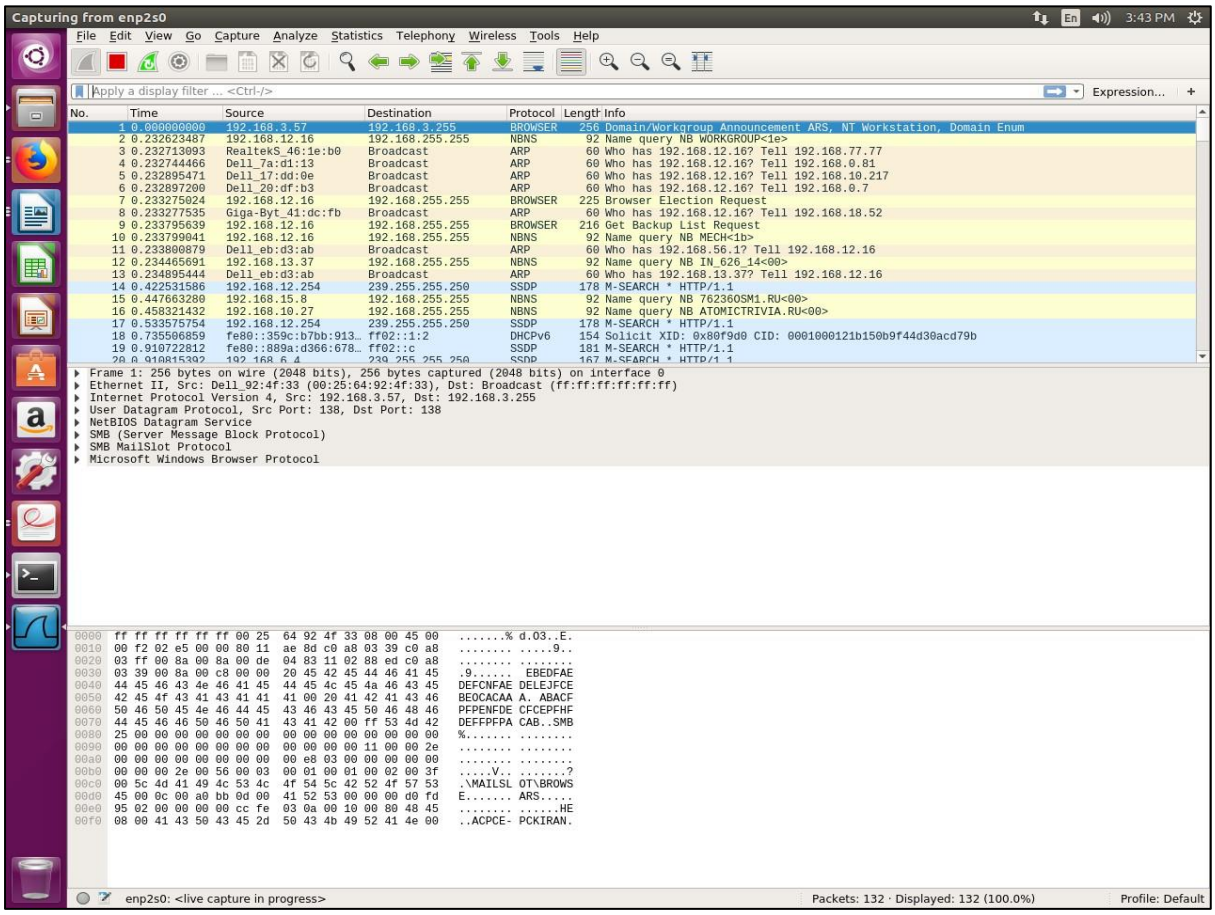
The first step involves selecting the network interface to capture its data. Remember, that the interfaces are different for different operating systems.



Filtering Packets:

If you ‘re trying to inspect something specific, such as the traffic a program sends when phoning home, it helps to close down all other applications using the network so you can narrow down the traffic. Still, you‘ll likely have a large amount of packets to sift through. That ‘s where Wireshark‘s filters come in. The most basic way to apply a filter is by typing it into the filter box at the top of the window and clicking

Apply (or pressing Enter). For example, type HTTP and you‘ll see only HTTP packets. When you start typing, Wireshark will help you autocomplete your filter.



<u>Let's have a look at these columns and what type of information they provide us with.</u>

- No - Represents a specific sequence number of the network packet. To classify a given packet, one can use this.
- Time - This is the time that a specific packet has been recorded.
- Source - This represents where we are getting the packets from. This is denoted as Internet Protocols (IP Addresses).
- Destination - This is used to represent the Internet Protocol (IP Address) where the packet is going.
- Protocol - This refers to the protocol of the data you have captured. This could be TCP, ARP et cetera
- Length- This is used to represent the size of the packet captured.
- Info - This gives you additional information about the packet you have captured.

Packet details panel:

Now that we can capture some data, try to click on a single row, and you will notice that some data is being displayed on the immediate window.

```
▸ Frame 14377: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on interface wlp2s0, id 0
▸ Ethernet II, Src: LiteonTe_40:73:14 (20:68:9d:40:73:14), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
▸ Internet Protocol Version 4, Src: 192.168.43.73, Dst: 239.255.255.250
▸ User Datagram Protocol, Src Port: 55249, Dst Port: 1900
▸ Simple Service Discovery Protocol
```

```
▸ Frame 14377: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits) on interface wlp2s0, id 0
▸ Ethernet II, Src: LiteonTe_40:73:14 (20:68:9d:40:73:14), Dst: IPv4mcast_7f:ff:fa (01:00:5e:7f:ff:fa)
▸ Internet Protocol Version 4, Src: 192.168.43.73, Dst: 239.255.255.250
▸ User Datagram Protocol, Src Port: 55249, Dst Port: 1900
▾ Simple Service Discovery Protocol
  ▸ M-SEARCH * HTTP/1.1\r\n
    HOST: 239.255.255.250:1900\r\n
    MAN: "ssdp:discover"\r\n
    MX: 1\r\n
    ST: urn:dial-multiscreen-org:service:dial:1\r\n
    USER-AGENT: Google Chrome/86.0.4240.198 Linux\r\n
    \r\n
    [Full request URI: http://239.255.255.250:1900*]
    [HTTP request 2/4]
    [Prev request in frame: 14367]
    [Next request in frame: 14380]
```

Packet bytes panel:

Remember when you clicked a given row from the packet details above, you could get details on the window .

```
0000  20 68 9d 40 73 14 0a ed  9d 68 2a ba 08 00 45 00   h·@s··· ·h*···E·
0010  00 34 b5 4d 40 00 93 06  3b 3d 28 4d e2 fa c0 a8   ·4·M@··· ;=(M····
0020  2b 49 01 bb db 06 f6 9e  b3 57 f7 44 11 a3 80 10   +I····· ·W·D····
0030  04 3e 7f 46 00 00 01 01  08 0a 7d 2f 1a 9f c8 91   ·>·F···· ··}/····
```

- **Conclusion**: Thus, we have studied about Analysis of Packet headers in **wire shark.**