WIRESHARK

AIM:

To provide deeper understanding of Network Protocol Analysis using Wireshark

DESCRIPTION:

Wireshark is an open-source packet analyzer, which is used for education, analysis, software development, communication protocol development, and network troubleshooting.It is used to track the packets so that each one is filtered to meet our specific needs. It is commonly called as a sniffer, network protocol analyzer, and network analyzer. It is also used by network security engineers to examine security problems.

Wireshark is a free to use application which is used to apprehend the data back and forth. It is often called as a free packet sniffer computer application. It puts the network card into an unselective mode, i.e., to accept all the packets which it receives.

Wireshark can be used in the following ways:

- It is used by network security engineers to examine security problems.
- It allows the users to watch all the traffic being passed over the network.
- It is used by network engineers to troubleshoot network issues.
- It also helps to troubleshoot latency issues and malicious activities on your network.
- It can also analyze dropped packets.
- It helps us to know how all the devices like laptop, mobile phones, desktop, switch, routers, etc., communicate in a local network or the rest of the world.

Wireshark is similar to tcpdump in networking. Tcpdump is a common packet analyzer which allows the user to display other packets and TCP/IP packets, being transmitted and received over a network attached to the computer. It has a graphic

end and some sorting and filtering functions. Wireshark users can see all the traffic passing through the network.

Wireshark can also monitor the unicast traffic which is not sent to the network's MAC address interface. But, the switch does not pass all the traffic to the port. Hence, the promiscuous mode is not sufficient to see all the traffic. The various network taps or port mirroring is used to extend capture at any point.

Port mirroring is a method to monitor network traffic. When it is enabled, the switch sends the copies of all the network packets present at one port to another port.

Features of Wireshark

- It is multi-platform software, i.e., it can run on Linux, Windows, OS X, FreeBSD, NetBSD, etc.
- It is a standard three-pane packet browser.
- It performs deep inspection of the hundreds of protocols.
- It often involves live analysis, i.e., from the different types of the network like the Ethernet, loopback, etc., we can read live data.
- It has sort and filter options which makes ease to the user to view the data.
- It is also useful in VoIP analysis.
- It can also capture raw USB traffic.
- Various settings, like timers and filters, can be used to filter the output.
- It can only capture packet on the PCAP (an application programming interface used to capture the network) supported networks.
- Wireshark supports a variety of well-documented capture file formats such as the PcapNg and Libpcap. These formats are used for storing the captured data.
- It is the no.1 piece of software for its purpose. It has countless applications ranging from the tracing down, unauthorized traffic, firewall settings, etc.

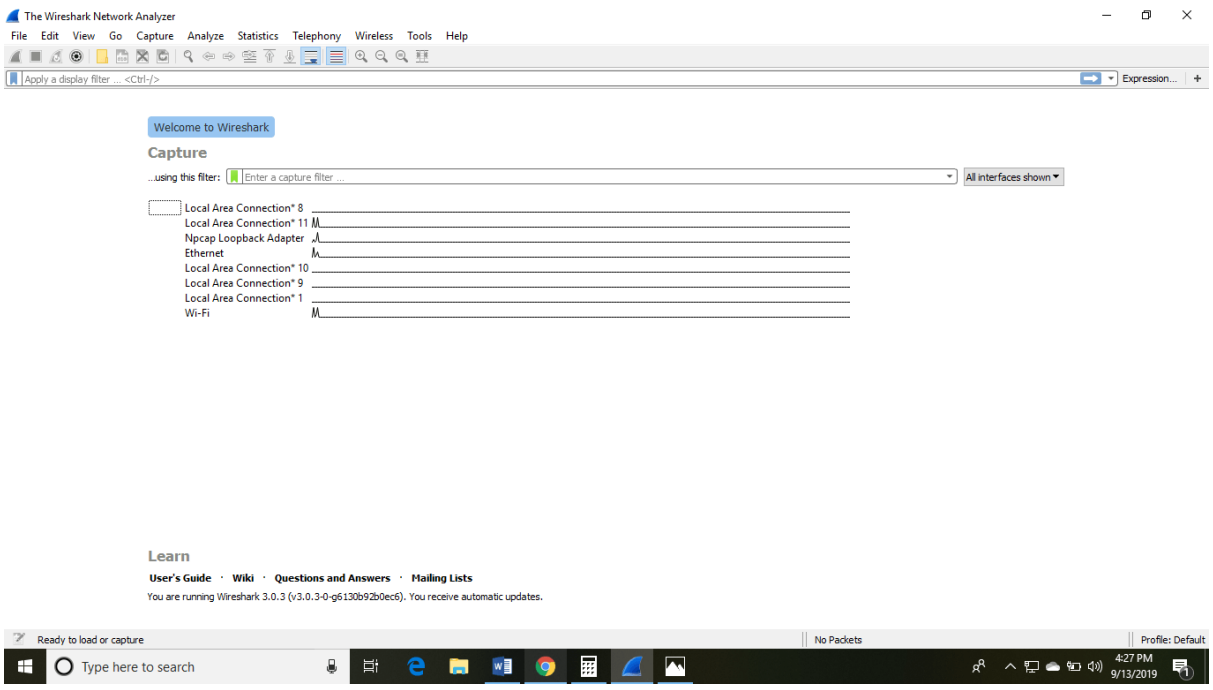
PROCEDURE

Below are the steps to install the Wireshark software on the computer:

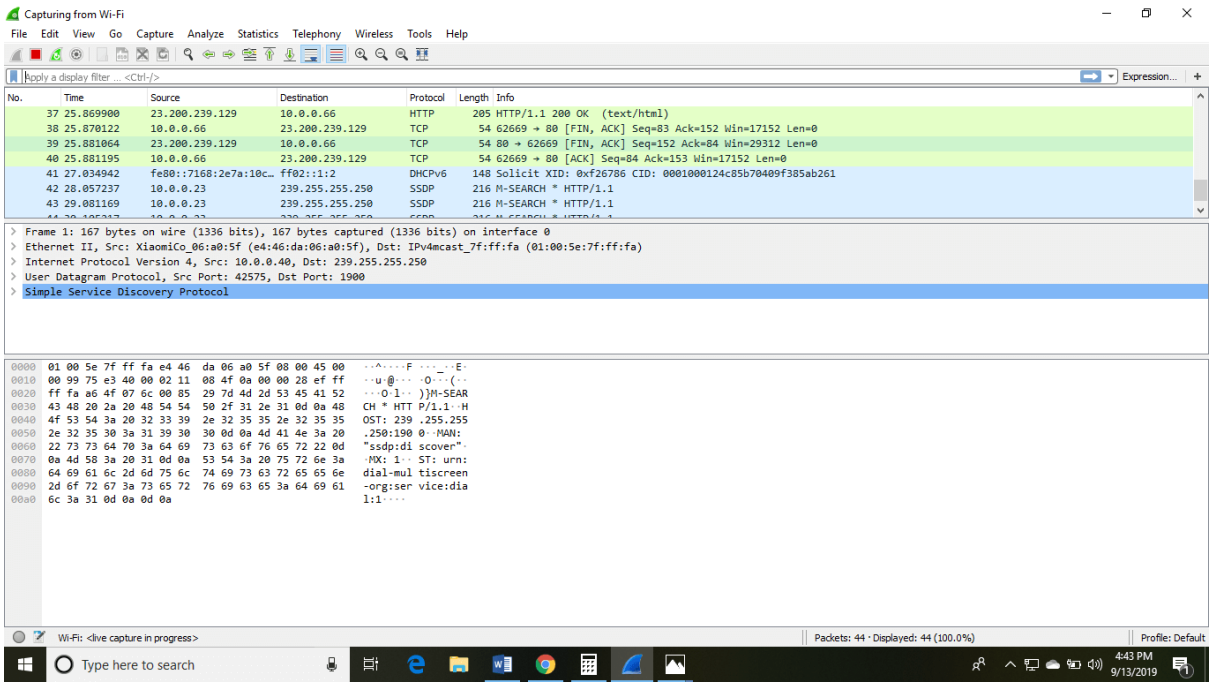
- Open the web browser.
- Search for 'Download Wireshark.'
- Select the Windows installer according to your system configuration, either 32-bt or 64-bit. Save the program and close the browser.
- Now, open the software, and follow the install instruction by accepting the license.

The Wireshark is ready for use.

On the network and Internet settings option, we can check the interface connected to our computer.If you are Linux users, then you will find Wireshark in its package repositories.By selecting the current interface, we can get the traffic traversing through that interface. The version used here is 3.0.3. This version will open as:



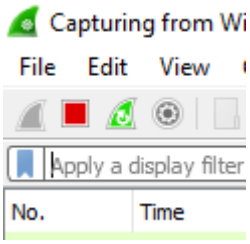
The Wireshark software window is shown above, and all the processes on the network are carried within this screen only. The options given on the list are the Interface list options. The number of interface options will be present. Selection of any option will determine all the traffic. For example, from the above fig. select the Wi-Fi option. After this, a new window opens up, which will show all the current traffic on the network. Below is the image which tells us about the live capture of packets and our Wireshark will look like:



The above arrow shows the packet content written in hexadecimal or the ASCII format. And the information above the packet content, are the details of the packet header.

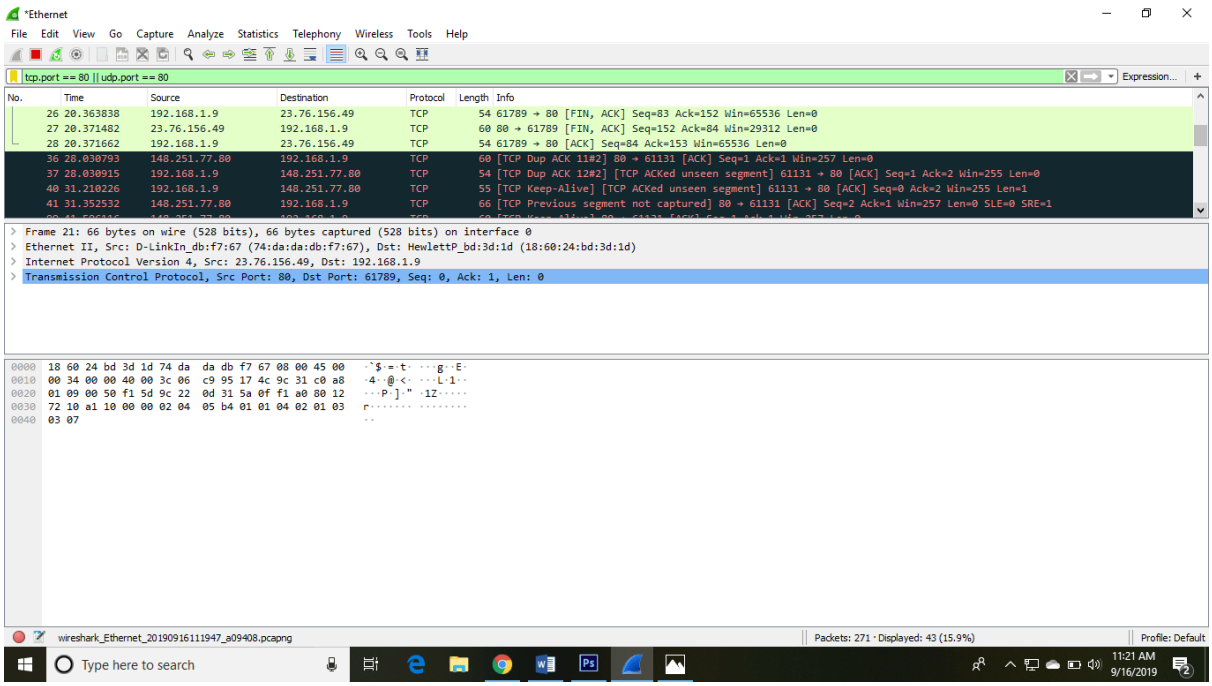
It will continue listening to all the data packets, and you will get much data. If you want to see a particular data, then you can click on the red button. The traffic will be stationary, and you can note the parameters like time, source, destination, the protocol being used, length, and the Info. To view in-depth detail, you can click on that particular address; a lot of the information will be displayed below that.

There will be detailed information on HTTP packets, TCP packets, etc. The red button is shown below:



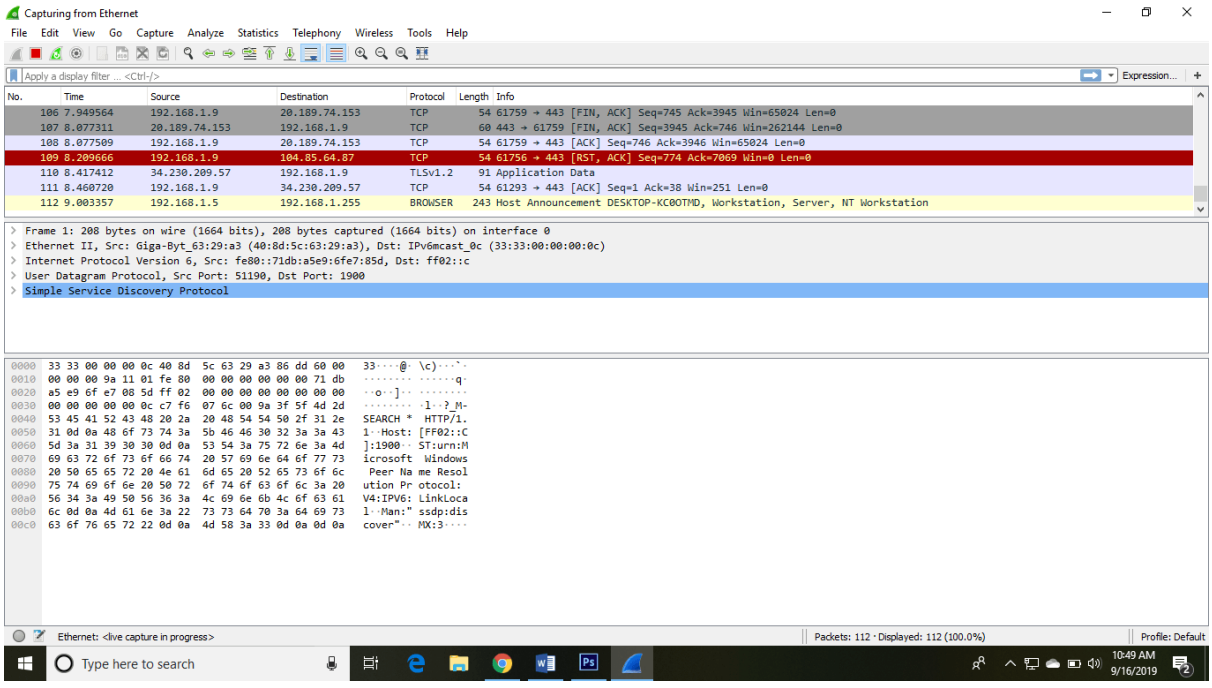
The screen/interface of the Wireshark is divided into five parts:

- ✓ First part contains a menu bar and the options displayed below it. This part is at the top of the window. File and the capture menus options are commonly used in Wireshark. The capture menu allows to start the capturing process. And the File menu is used to open and save a capture file.
- ✓ The second part is the packet listing window. It determines the packet flow or the captured packets in the traffic. It includes the packet number, time, source, destination, protocol, length, and info. We can sort the packet list by clicking on the column name.
- ✓ Next comes the packet header- detailed window. It contains detailed information about the components of the packets. The protocol info can also be expanded or minimized according to the information required.
- ✓ The bottom window called the packet contents window, which displays the content in ASCII and hexadecimal format.
- ✓ At last, is the filter field which is at the top of the display. The captured packets on the screen can be filtered based on any component according to your requirements. For example, if we want to see only the packets with the HTTP protocol, we can apply filters to that option. All the packets with HTTP as the protocol will only be displayed on the screen, shown below:

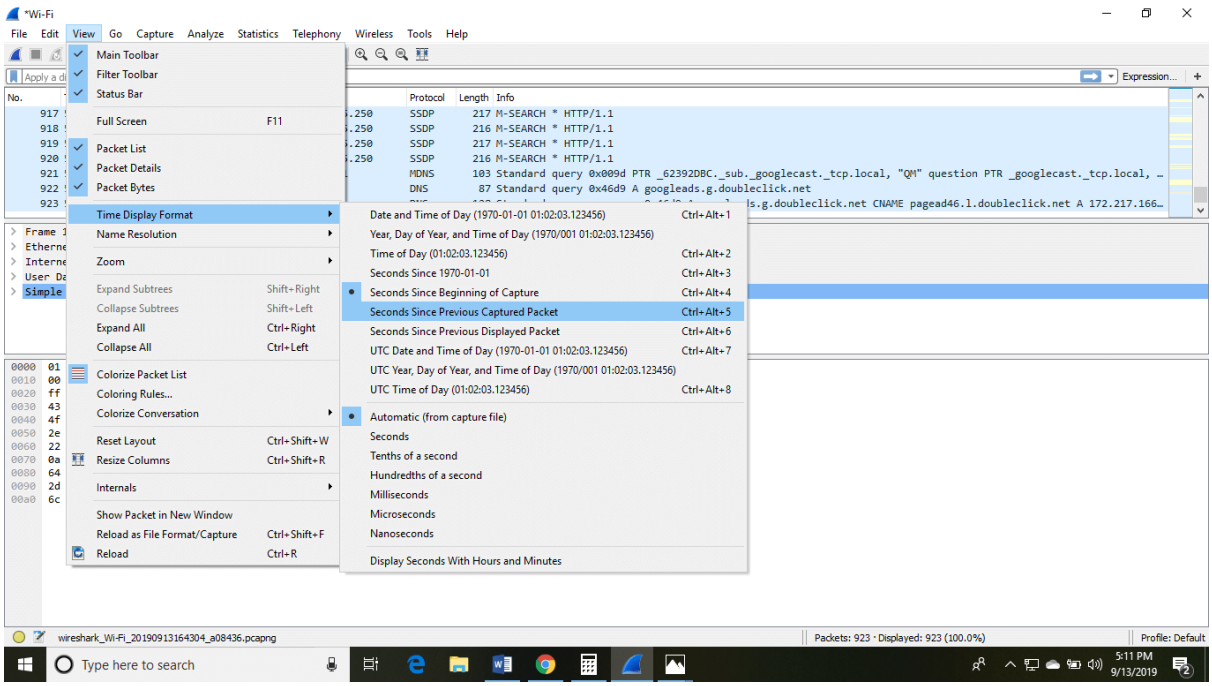


You can also select the connection to which your computer is connected. For example, in this PC, we have chosen the current network, i.e., the ETHERNET.

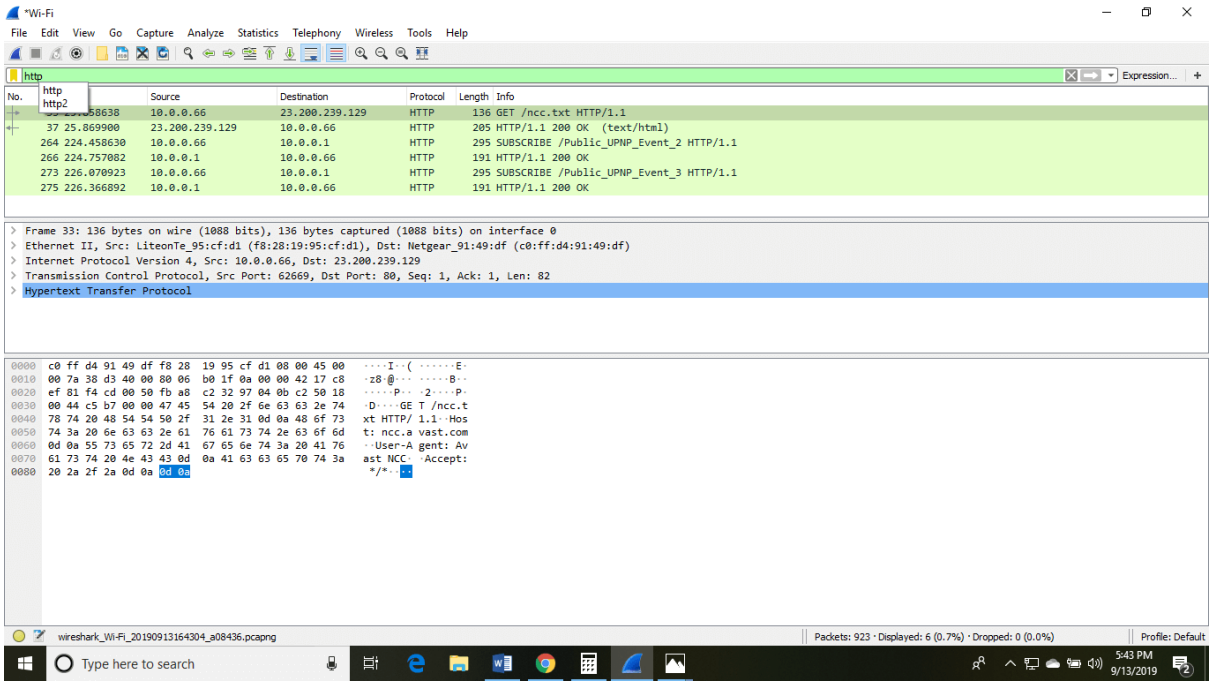
After connecting, you can watch the traffic below:



In view option on the menu bar, we can also change the view of the interface. You can change the number of things in the view menu. You can also enable or disable any option according to the requirements.

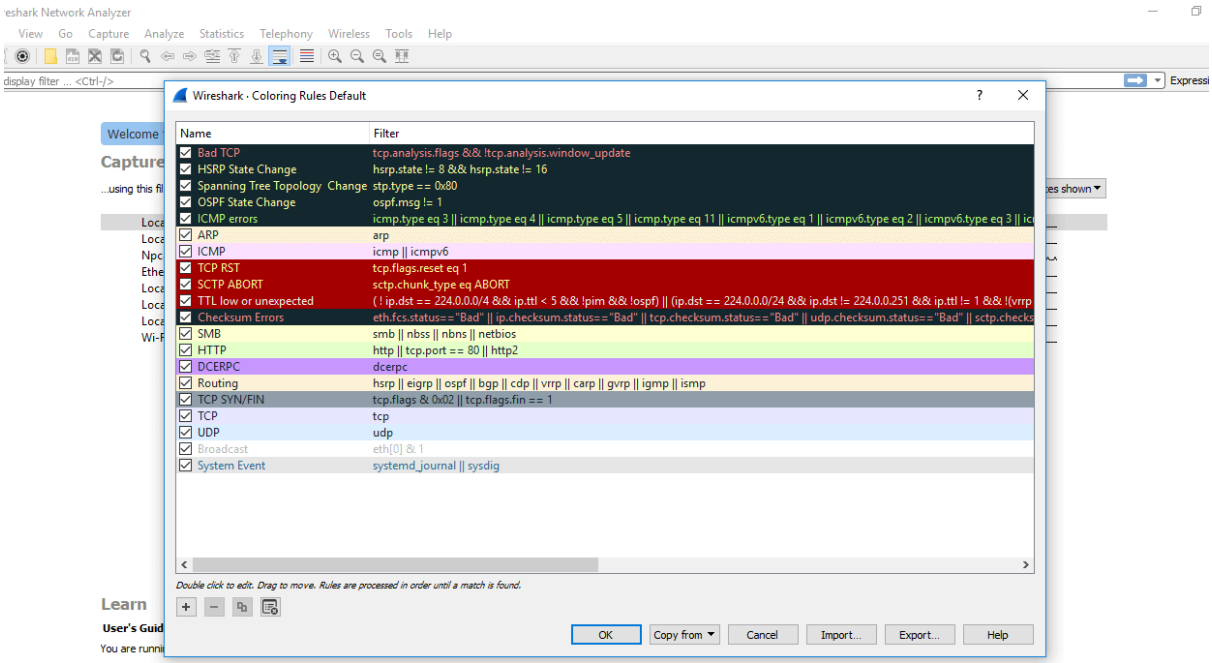


There is a filter block below the menu bar, from where a large amount of data can be filtered. For example, if we apply a filter for HTTP, only the interfaces with the HTTP will be listed.



If you want to filter according to the source, right-click on the source you want to filter and select 'Apply as Filter' and choose '...and filter.'

Steps for the permanent colorization are: click on the 'View' option on the menu bar and select 'Coloring Rules.' The table will appear like the image shown below:



For the network administrator job, advanced knowledge of Wireshark is considered as the requirements. So, it is essential to understand the concepts of the software. It contains these 20 default coloring rules which can be added or removed according to the requirements. Select the option 'View' and then choose 'Colorize Packet List,' which is used to toggle the color on and off. Whenever we type any commands in the filter command box, it turns green if your command is correct. It turns red if it is incorrect or the Wireshark does not recognize your command.

If you have a Linux system, you'd install Wireshark using the following sequence (notice that you'll need to have root permissions):

$ sudo apt-get install wireshark

$ sudo dpkg-reconfigure wireshark-common

$ sudo usermod -a -G wireshark $USER

$ newgrp wireshark

Once you have completed the above steps, you then log out and log back in, and then start Wireshark:

$ wireshark

Wireshark tries to help you identify packet types by applying common-sense color coding. The table below describes the default colors given to major packet types.

Color in Wireshark	Packet Type
Light purple	TCP
Light blue	UDP
Black	Packets with errors
Light green	HTTP traffic
Light yellow	Windows-specific traffic, including Server Message Blocks (SMB) and NetBIOS

Color in Wireshark	Packet Type
Dark yellow	Routing
Dark gray	TCP SYN, FIN and ACK traffic

In Wireshark, just go to Statistics >> I/O Graph, and you'll see a graph similar to the one below:

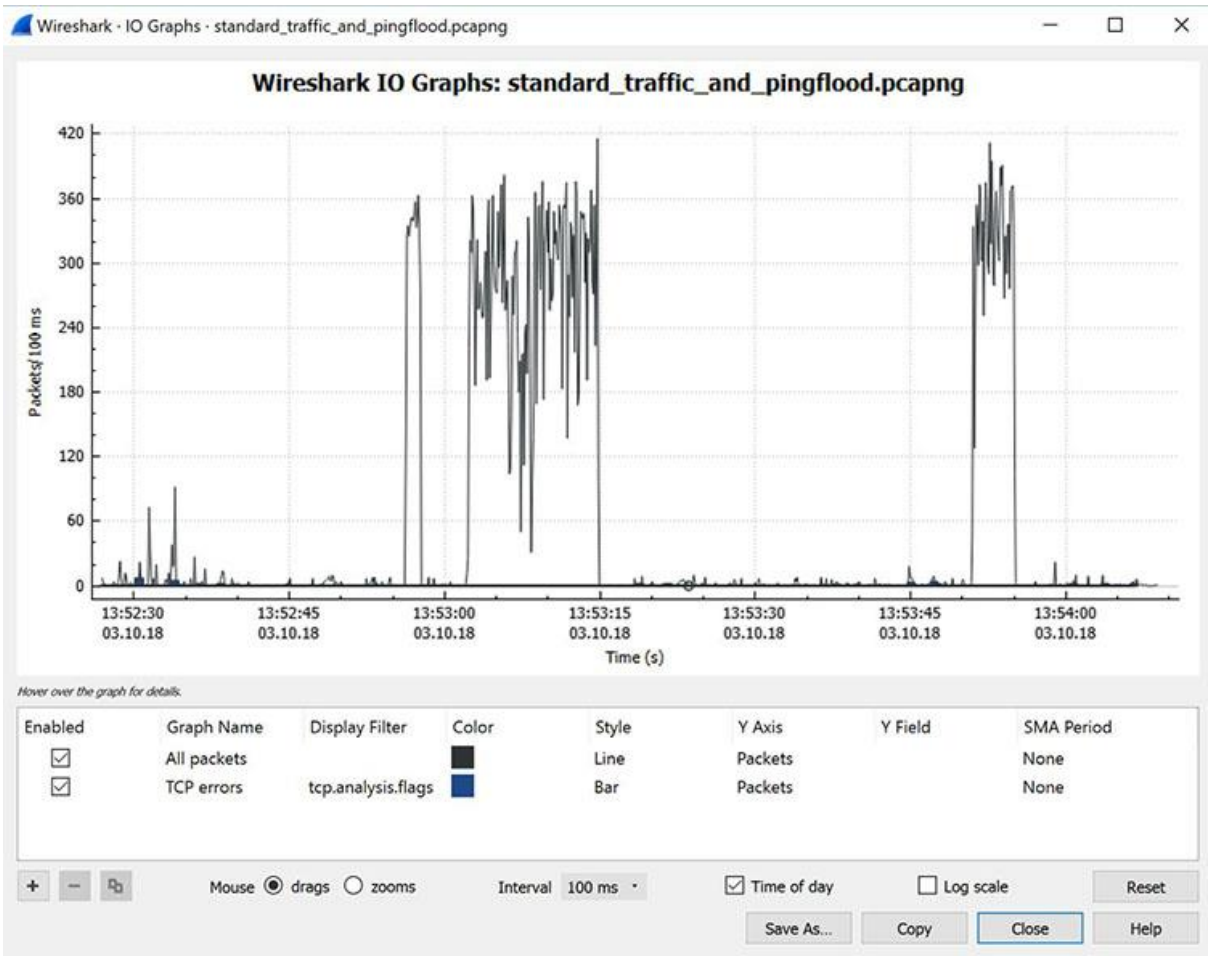


Figure : Viewing the input/output traffic graph in Wireshark

This particular graph is showing typical traffic generated by a home office. The spikes in the graph are bursts of traffic that were caused by generating a Distributed Denial of Service (DDoS) attack using a few Linux systems.

In this case, three major traffic bursts were generated. Many times, cybersecurity pros use Wireshark as a quick and dirty way to identify traffic bursts during attacks.

It's also possible to capture the amount of traffic generated between one system and another. If you go to Statistics and then select Conversations, you will see a summary of conversations between end points, as shown below in Figure.

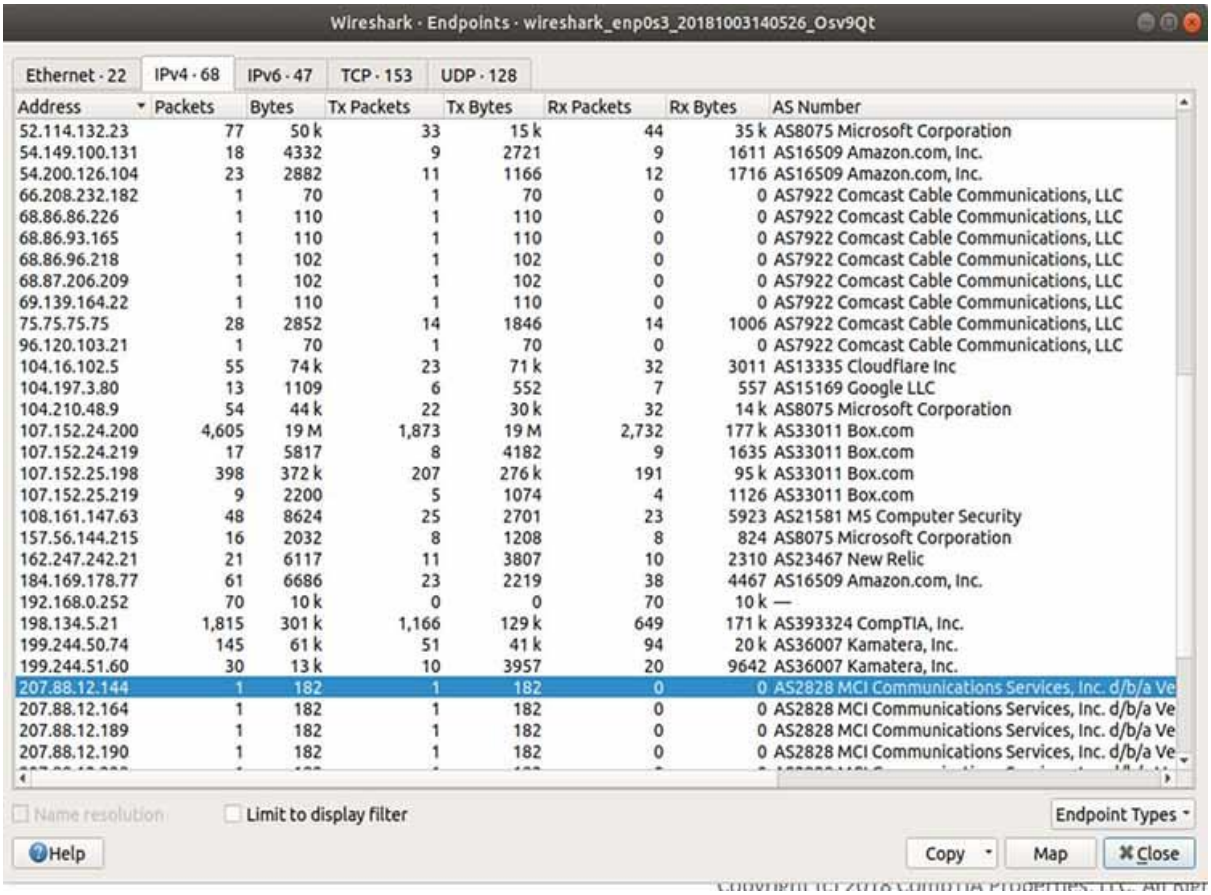


Figure: Viewing endpoint conversations in Wireshark

In the above case, Wireshark was used to see if an old piece of equipment from MCI communications that was running on a client's network could be traced.

It turned out that the client didn't know this device was even on the network. Thus, it was removed, helping to make the network a bit more secure. Notice, also, that this

network connection is experiencing a lot of traffic to Amazon (administering a server in AWS at the time) and Box.com (using Box for system backup at the time).

In some cases, it is even possible to use Wireshark to identify the geographic location of source and destination traffic. If you click on the Map button at the bottom of the screen , Wireshark will show you a map, providing its best guess of the location of the IP addresses you've identified.

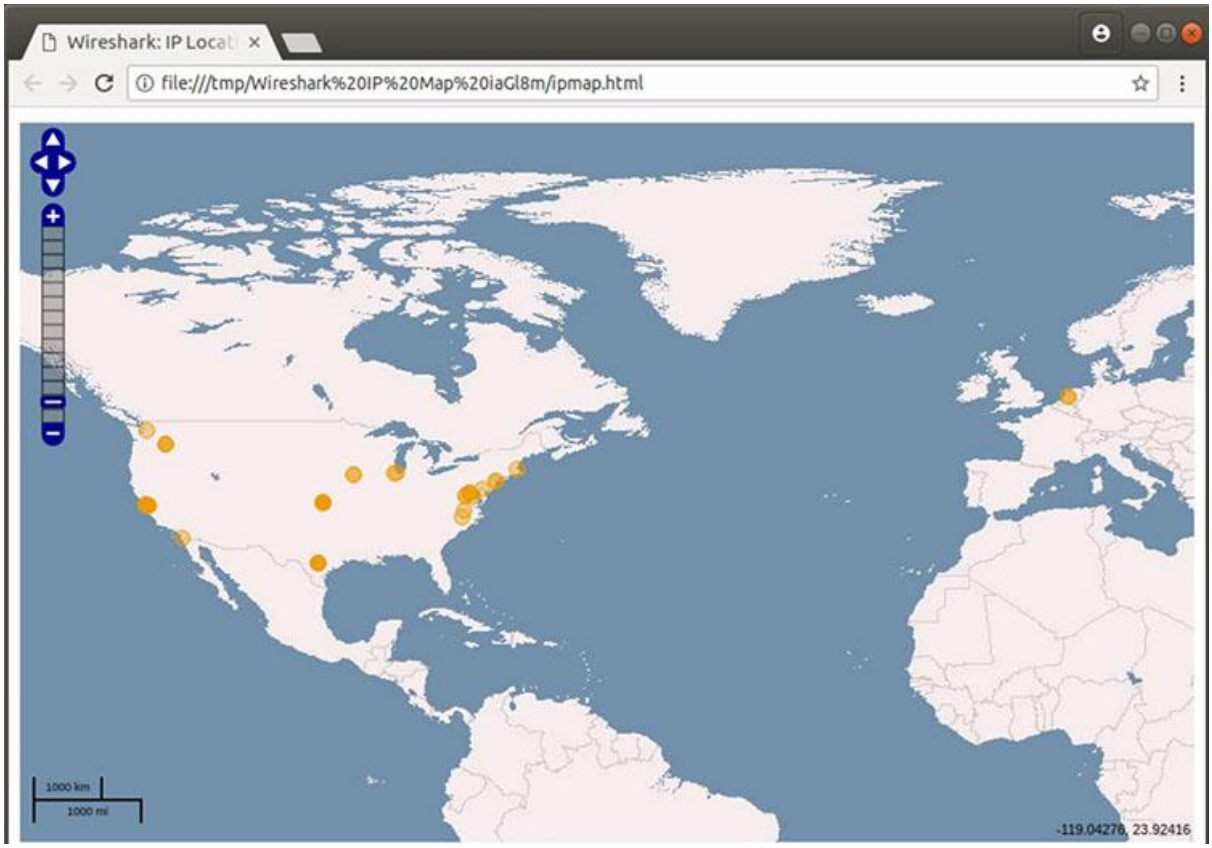


Figure: Viewing geographic estimations in Wireshark

Because IPv4 addresses can be easily spoofed, you can't rely completely on this geographical information. But it can be fairly accurate.

You can apply Wireshark filters in two ways:

1. In the Display Filter window, at the top of the screen
2. By highlighting a packet (or a portion of a packet) and right-clicking on the packet

Wireshark filters use key phrases, such as the following:

`ip.addr`	Specifies an IPv4 address
`ipv6.addr`	Specifies an IPv6 address
`src`	Source - where the packet came from
`dst`	Destination - where the packet is going

You can also use the following values:

`&&`	Means “and,” as in, “Choose the IP address of 192.168.2.1 and 192.168.2.2”
`==`	Means “equals,” as in “Choose only IP address 192.168.2.1”
`!`	Means “not,” as in, do not show a particular IP address or source port

Valid filter rules are always colored green. If you make a mistake on a filter rule, the box will turn a vivid pink.

For example, let’s say you want to see packets that have only the IP address of 18.224.161.65 somewhere inside. You would create the following command line, and put it into the Filter window:

ip.addr == 18.224.161.65

Figure shows the results of adding that filter:

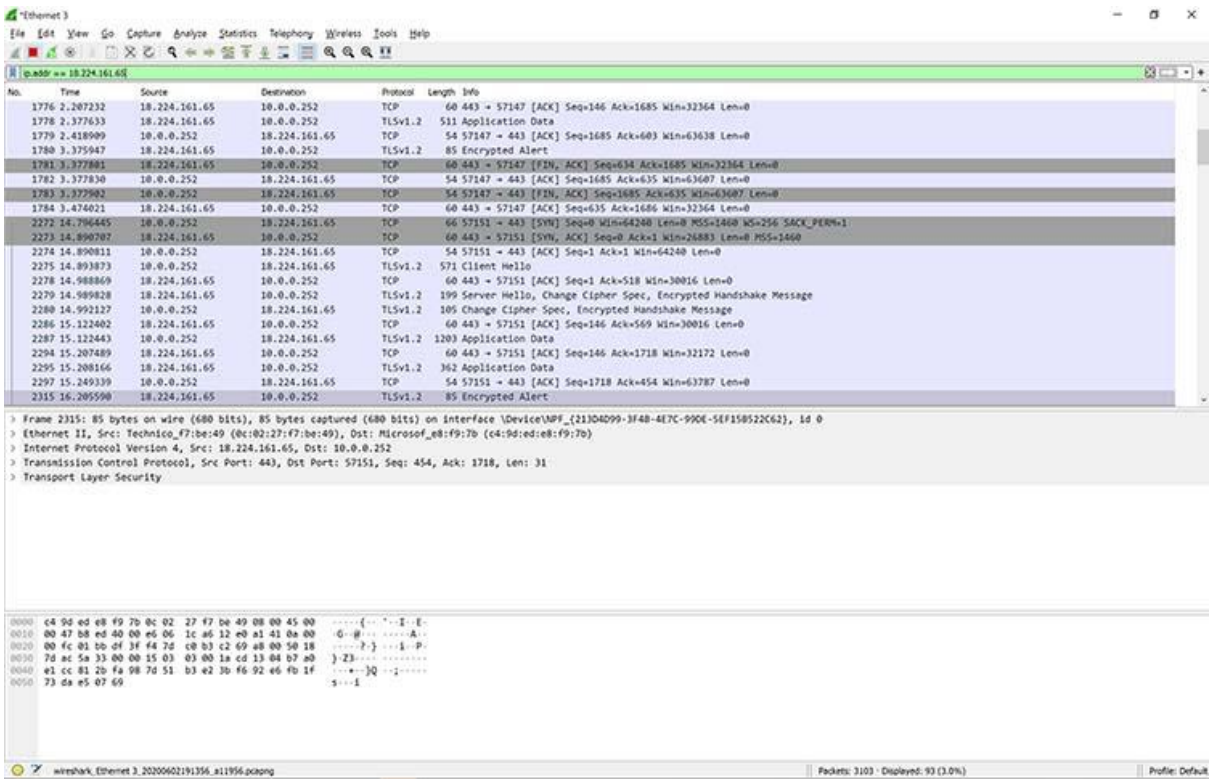


Figure: Applying a filter to a capture in Wireshark

Alternatively, you can highlight the IP address of a packet and then create a filter for it. Once you select the IP address, right-click, and then select the Apply As Filter option.

You'll then see a menu of additional options. One of those is called Selected. If you choose Selected, then Wireshark will create a filter that shows only packets with that IP address in it.

You can also decide to filter out a specific IP address using the following filter, also shown in Figure:

!ip.addr==18.224.161.65

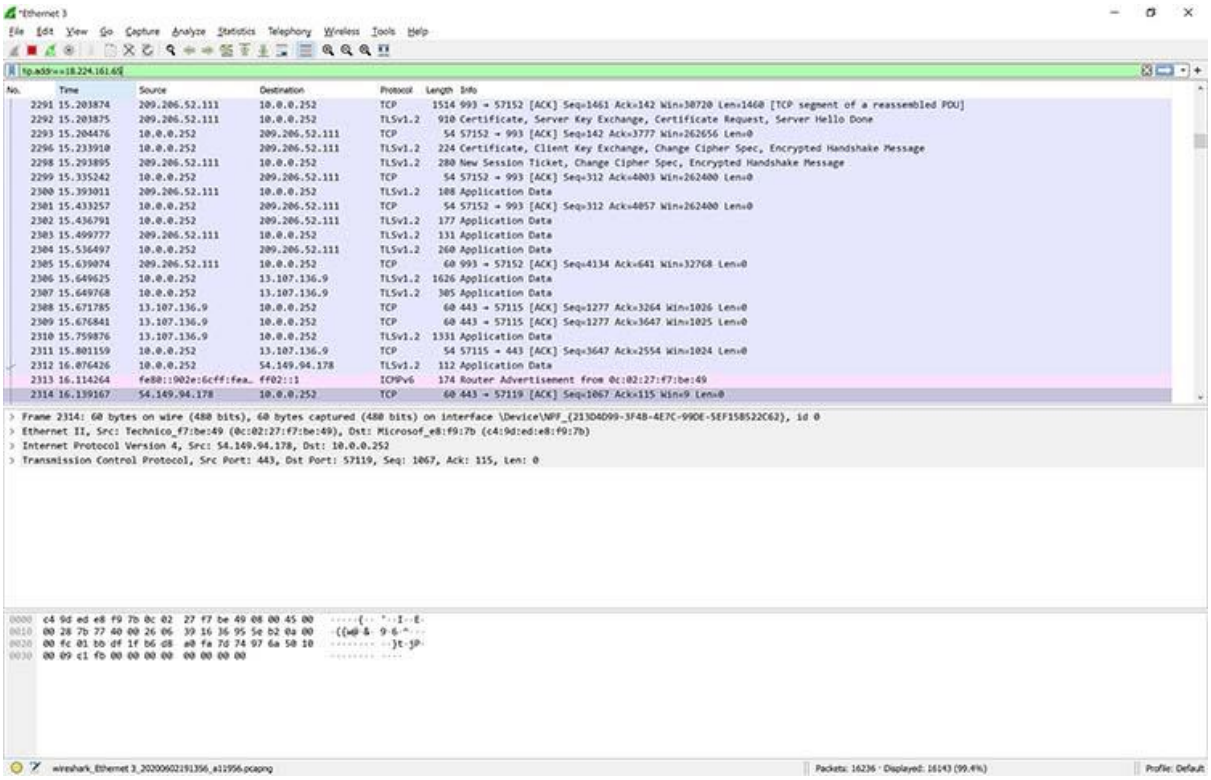


Figure: Filtering out a specific IP address in Wireshark

You're not limited to just IPv4 addresses. For example, if you want to see if a particular computer is active and using an IPv6 address on your network, you can open up a copy of Wireshark and apply the following rule:

ipv6.dst == 2607:f8b0:400a:15::b

This same rule is shown in Figure.

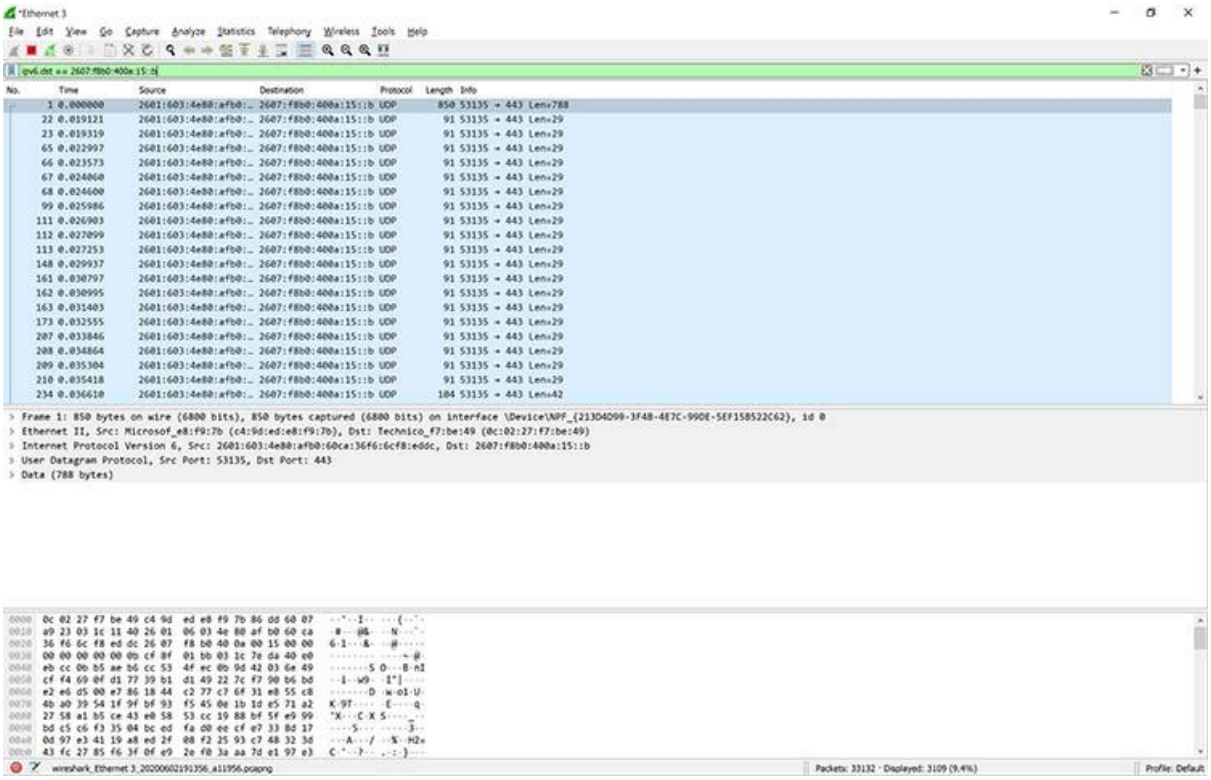


Figure: Applying an IPv6 filter in Wireshark

Clearly, this system is alive and well, talking on the network. There are so many possibilities.

Additional filters include:

`tcp.port==8080`	Filters packets to show a port of your own choosing – in this case, port 8080
`!(ip.src == 162.248.16.53)`	Shows all packets except those originating from 162.248.16.53
`!(ipv6.dst == 2607:f8b0:400a:15::b)`	Shows all packets except those going to the IPv6 address of 2607:f8b0:400a:15::b

`ip.addr == 192.168.4.1 && ip.addr == 192.168.4.2`	Shows both 192.168.4.1 and 192.168.4.2
`http.request`	Shows only http requests – useful when troubleshooting or visualizing web traffic