**WIRESHARK AND ITS FUNDAMENTALS**

**What is Wireshark?**

In our increasingly connected world, understanding the inner workings of computer networks has become essential. Whether it's diagnosing network issues, optimizing performance, or ensuring security, having the ability to capture, analyse , and interpret network traffic is a valuable skill for network administrators, developers, and security professionals alike.

Wireshark is a powerful, open-source network protocol analyser that has revolutionized the way we analyse and understand network communications. Wireshark enables you to look into the network traffic, providing you with a detailed view of packets traversing your network. It allows you to dissect protocols, spot anomalies and gain insights into how data is exchanged between systems.

**Uses of Wireshark**

Wireshark has many uses, including [troubleshooting networks](https://www.comptia.org/content/guides/a-guide-to-network-troubleshooting) that have performance issues. Cybersecurity professionals often use Wireshark to trace connections, view the contents of suspect network transactions and identify bursts of network traffic. It’s a major part of any IT pro’s toolkit – and hopefully, the IT pro has the knowledge to use it.

**Features of Wireshark**

The following are some of the features that Wireshark provides:

1) Capture packet data from a network interface

2) Open files containing packet data

3) Comprehensive Display Filters

4) Filter Protocols

5) Statistical Analysis

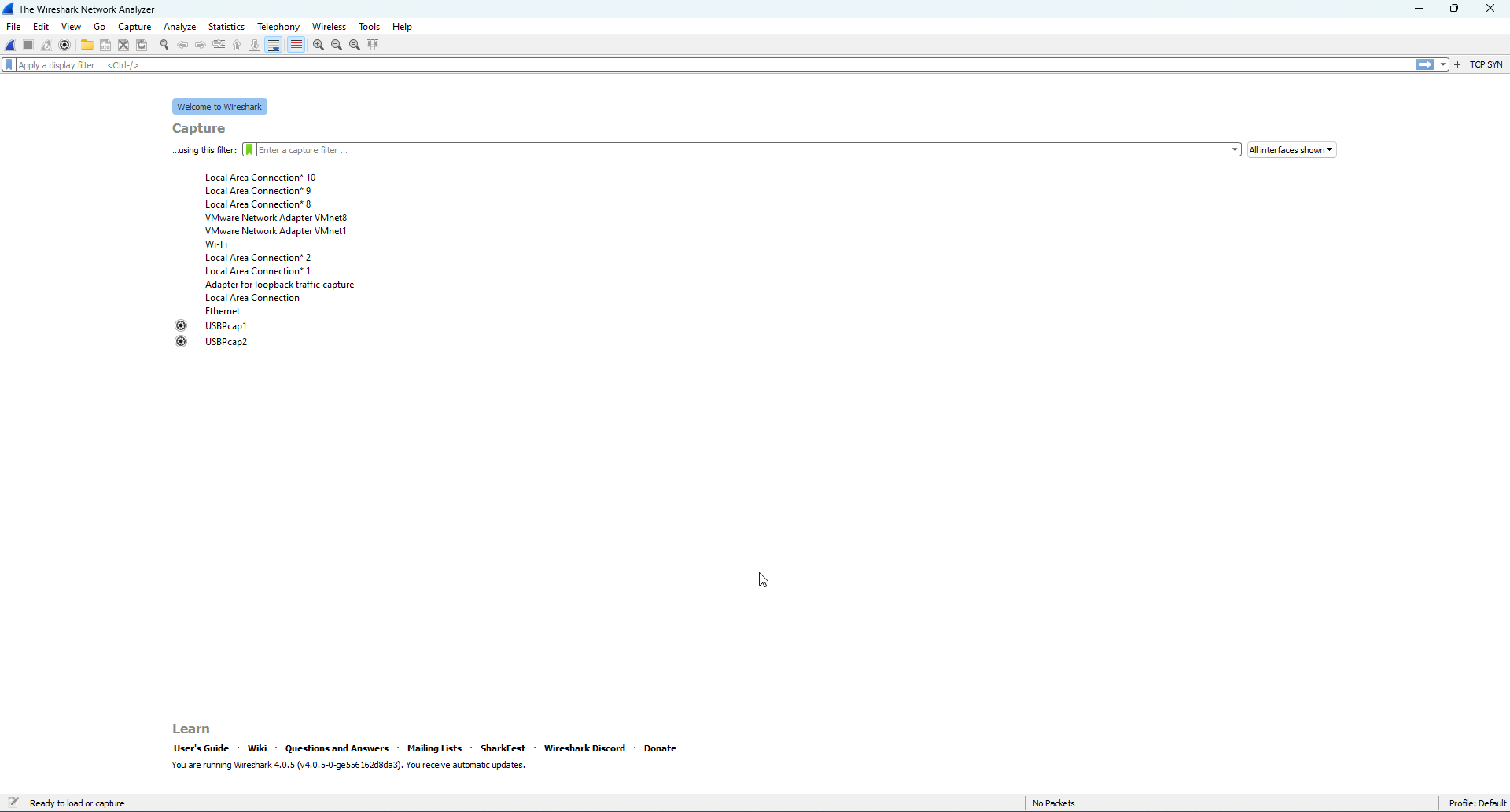
6) Mapping of IP address locations (using tools like GeoIP)

7) Real-Time Monitoring

Apart from these Wireshark also provides many tools to get a better understanding about the network traffic and helps to gain information about the servers, protocols, information and communication happening in the network.

**Fundamentals of Wireshark**

Home Page of Wireshark



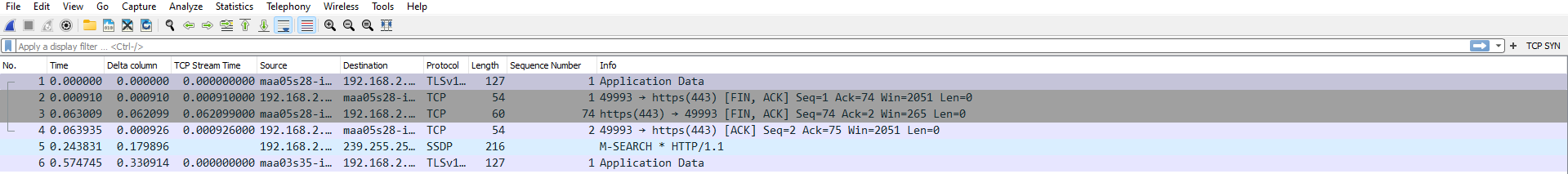
The home page of Wireshark displays the menu bar, display filter, capture filter, make new profiles and also customise it, shows various interfaces such as LAN, Wi-fi, VMware etc.

Capture filter can be used to capture live packets from our networks.

By clicking at each interface we can reach our display page where all the packet communications on that interface can be seen and analysis can be performed using network traffic.

Default Wireshark Column Display

Menu Bar



Column

Display

DisplayFilter

The above picture displays the columns which are there on Wireshark during packet capturing, the display filter and the menu bar which consists of file, edit, view etc options. Each column gives unique information about the packet that was captured or information about communication with different protocols.

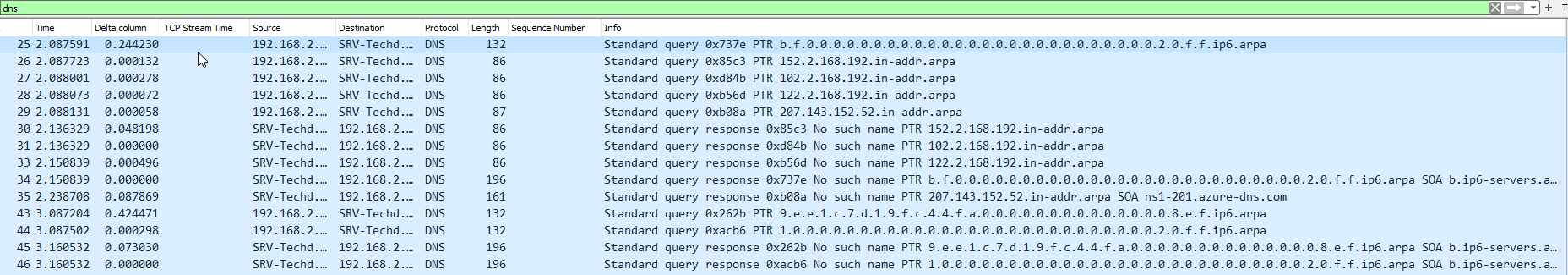
Functions of each column and options:

1) Menu Bar: Menu bar consists of various options such as File, Edit, View, Go, Capture, Analyze etc.

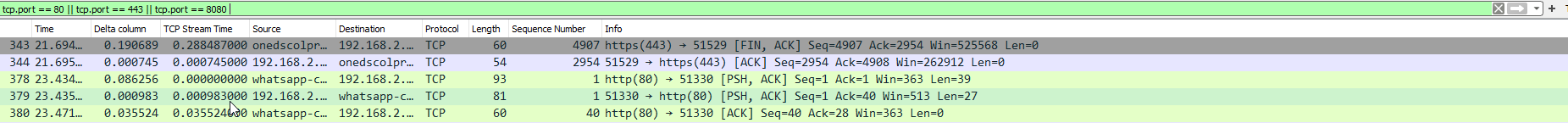
2) Display Filter: This filter allows the user to see the specific connections and protocols during network capturing. It enables you to filter and display only the packets that match certain conditions, making it easier to focus on relevant information. By using display filters, you can narrow down the packet list to show specific protocols, source or destination IP addresses, port numbers, or any other field value within the captured packets.

Some of the examples to how to use display filter are:

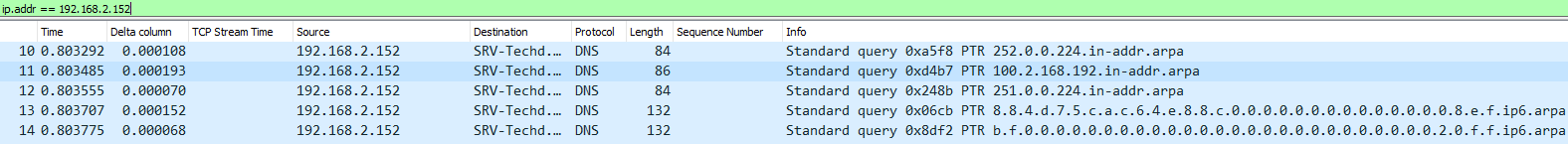
* In this example we use the display filter to display only those communications which involves DNS (domain name server) protocol.



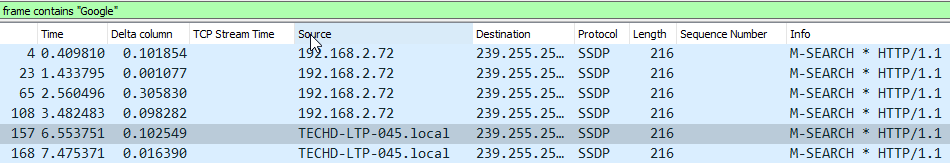
* **tcp.port == 80 || tcp.port == 443 || tcp.port ==** 8080. This filter is used to capture TCP communication but on different ports. In this example, this filter is used to capture TCP communication between HTTP(port 80), HTTPS(port 443) and web servers(port 8080).



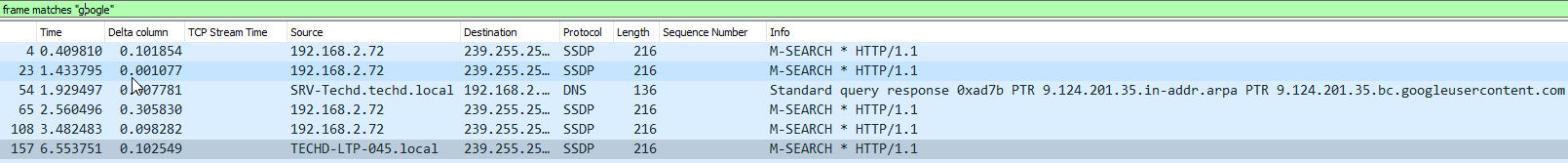
* ip.addr == 192.168.2.152 filter helps to display all the communications to and from the ip address 192.168.2.152 above internet protocol( tcp and udp).



* The "contains" operator allows a filter to search for a sequence of characters, expressed as a string, or bytes. This filter is case sensitive.

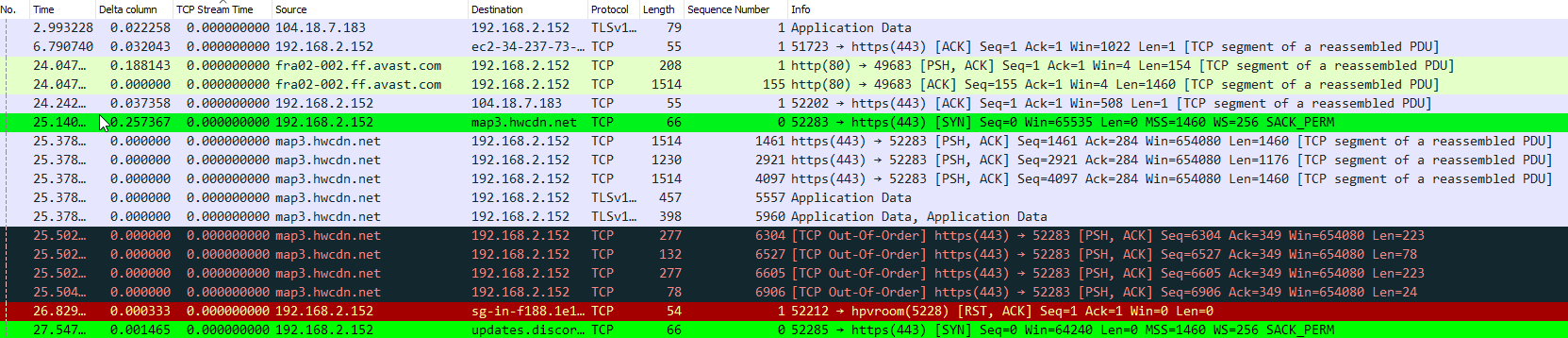


The "matches" or "~" operator allows a filter to apply to a specified Perl-compatible regular expression (PCRE2). The regular expression must be a double quoted string. This filter is case insensitive.



Apart from all these examples, the display filter can also be used for various other purposes such as using comparison operators we can filter between communications (for example all tcp.port > 1024), we can filter on the vendor portion of an ethernet address using the slice operator etc.

3) Column Display: The default column display in Wireshark provides a wealth of information and is a very useful tool for data filtration and command and control of the activity. We can also customise the display so as to provide a better view of the activity in the network.

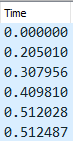


The default columns in the column display are:

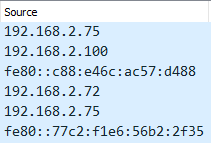
1. **No**. -Frame number from the beginning of the pcap. The first frame is always 1.



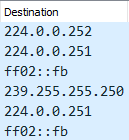
1. **Time** – This column gives information about the seconds which are converted down to the nanosecond from the first frame of the pcap. The first frame is always 0.000000.



1. **Source** – This column provides information about the source address, commonly an IPv4, IPv6, or Ethernet address.

****

1. **Destination** – This column provides information about the destination address where the packet is headed , commonly an IPv4, IPv6, or Ethernet address.



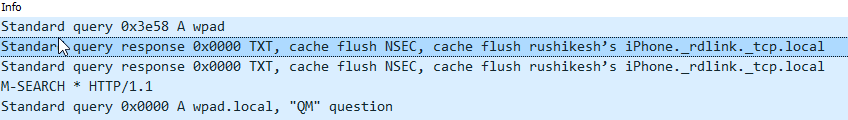
1. **Protocol** – This column provides information about the protocol used in the Ethernet frame, IP packet, or TCP segment (ARP, DNS, TCP, HTTP, etc.).



1. **Length** - This column provides information about the length of the frame in bytes.



1. **Info -** This column provides information about the queries and the communication happening between packets, network ,servers etc.



To understand lets take an example of the query :

*Standard query response 0x0000 TXT, cache flush NSEC, cache flush rushikesh’s iPhone.\_rdlink.\_tcp.local. What is the meaning of this info on wireshark*

In this query:

* "Standard query response 0x0000": This indicates that it is a DNS (Domain Name System) response to a standard query. The "0x0000" represents the query ID, which is a unique identifier for the query-response pair.
* 2. "TXT": This refers to the resource record type in the DNS response. TXT records are typically used to store text-based information associated with a domain or hostname.
* 3. "Cache flush NSEC": NSEC (Next-Secure) is a type of DNS resource record used for providing authenticated denial of existence. In this case, the "cache flush" indicates that the NSEC record should be flushed from the DNS cache.
* 4. "Cache flush rushikesh’s iPhone.\_rdlink.\_tcp.local": This represents the fully qualified domain name (FQDN) for the specific resource being

queried. It indicates that the DNS cache should be flushed for the resource named "rushikesh’s iPhone.\_rdlink.\_tcp.local"

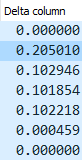
Apart from the default columns we can also add some columns so as to help us provide a better understanding about the network activity which is going on.

Examples of such columns are:

1. **Delta Time Column –** The delta time column in Wireshark provides the time difference between the current packet and the previous packet in the capture file. It measures the time elapsed between the arrival of each packet on the network.

We can add this column by following the above steps:

Edit > Preferences > Appearances > Columns > + button to add column with their title, type etc.

****

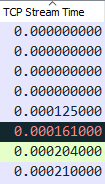
1. **Time since previous frame in this TCP stream column -** Since multiple connections are getting established in the network, this column can be helpful to separate the particular connection and will let us know the time before its previous frame which has passed in their connection.

We can add this column by following the above steps:

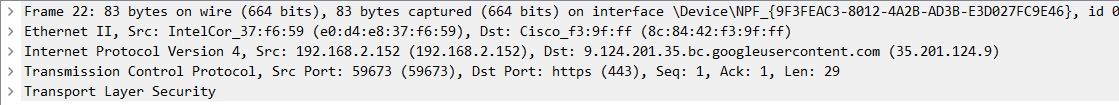
Frame Details > Transmission Control Protocol frame > [Timestamps] >

Time since previous frame in this TCP stream > Right click > Apply as

Column.

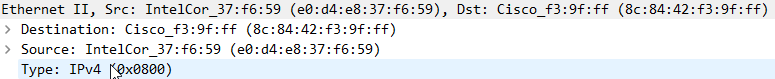


4) Packet Details: This window shows the protocols and protocol fields of the packet selected in the “Packet List” .

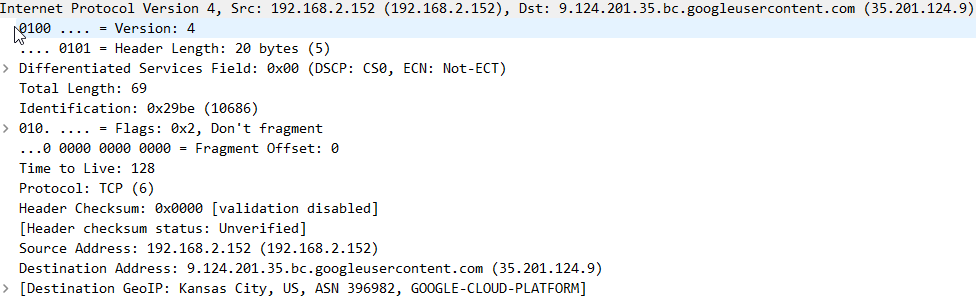


Expanding a packet allows you to view its details and layers:

**Ethernet:** The Ethernet layer contains information about MAC addresses (source and destination) and the Ethernet protocol used (e.g., Ethernet II, IEEE 802.3).

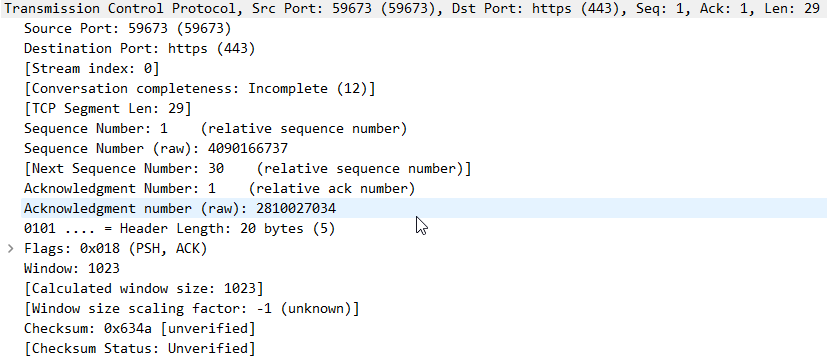


**IP:** The IP layer provides information about the IP addresses of the source and destination hosts, the IP version (IPv4 or IPv6), and other IP-related details.



This layer also provides information about each packet’s information details such as Version, Header length, Total Length, Differentiated Services field etc.

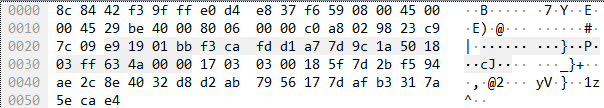
**Transport Layer:** Depending on the protocol used, this layer could be TCP (Transmission Control Protocol) or UDP (User Datagram Protocol). It includes details such as source and destination ports, sequence numbers, acknowledgment numbers, etc.



**Application Layer:** This layer represents the specific application protocol being used, such as HTTP, DNS, FTP, etc. It includes protocol-specific information and can provide insights into the data being exchanged at the application level.



4) **Hexadecimal View:** The hexadecimal view in Wireshark provides a detailed representation of the actual binary data contained within a captured packet. It displays the packet's contents as a series of hexadecimal values.



Some additional features of Wireshark:

1. **Colorize Filters:** There are already some coloring rules which are set up by Wireshark as default but we can add more coloring rules for connections for our understanding using Wireshark features.

For example, if we want to colorize the TCP SYN connection as green.

To perform this, we must follow the following steps:

View > Coloring Rules > Click + button > edit Name (for eg. TCP SYN) and edit filter ( for this example i.e. tcp.flags.syn == 1) > Background > Choose green colour > Press Ok.



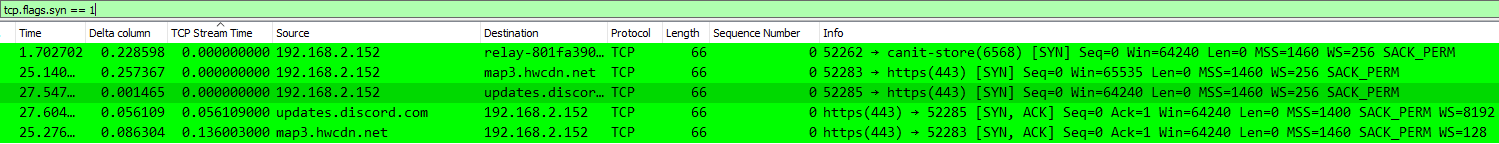


As you can see above all the TCP SYN connections are displayed in green colour.

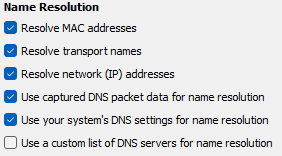
1. **Saving Display Filter Buttons:** We can use Wireshark features to create buttons for connections and make easier for us to see those connections just by clicking at it. This really helps us during network analysis.  
   For example, if we want to set up a button for all TCP SYN connections.

To perform this, we must follow the following steps:

Click the top right + button > Edit name and Filter (TCP SYN and tcp.flags.syn = = 1) > Press OK.



1. **Name Resolution:** Name resolution in Wireshark refers to the process of translating network addresses (IP addresses, MAC addresses, port numbers, etc.) into more meaningful names, such as domain names or hostnames. It helps to enhance readability of the network traffic by replacing numeric addresses with recognizable names.



**Resolve MAC addresses –** This section in name resolution provides the organisational unique identifier rather than a hexadecimal MAC address.

**Resolve transport names –** This resolution is used by Wireshark to compare the port values in the trace file with the well-known port numbers and hence replace the port values with the protocols.

**Resolve network (IP) addresses –** This resolution is used resolve IPv4, IPv6 and IPX addresses into host names. The next set of boxes are used to determine how the name resolution should be performed.

**Use captured DNS packet data for name resolution –** This section uses the address/name pairs found in captured DNS packets for name resolution.

**Use your system’s DNS settings for name resolution –** This section uses the system’s configured name resolver (usually DNS) for name resolution.

**Use a custom list of DNS servers for name resolution -** This section uses DNS servers to resolve network names if TRUE. If FALSE, default information is used.

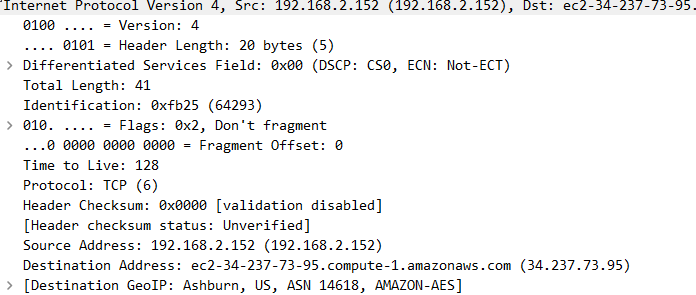
1. **MAPPING IP ADDRESS TO LOCATIONS:** This module uses all the IPs in the pcap file in Wireshark and identify from where they are coming from and also drop all those locations on the map. This feature is useful when we are performing forensics for cyber security.

We can perform this feature with the help of GeoIP tool databases i.e. GeoLite2 Country, City and ASN databases.

To show locations we should follow the following steps in Wireshark:

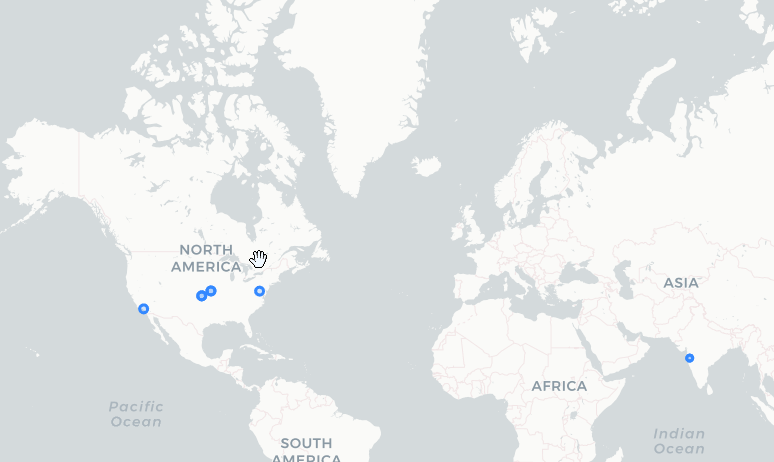
Edit > Preferences > Name resolution > Max Mind Directories > Choose the folder where you have stored the databases > Press Ok.

After performing these steps, you can see the location of the frames in the Internet Protocol section in the Packet filter section.



To show these locations on a Map we perform the following steps:

Statistics > Endpoints > Map > Open in Browser.

****

1. **TCP Handshake Protocol –** TCP uses a three-way handshake to establish a reliable connection. The connection is full duplex, and both sides synchronize (SYN) and acknowledge (ACK) each other. The exchange of these four flags is performed in three steps: SYN, SYN-ACK, ACK. It also uses a 4-way handshake to terminate connection.

The clients sends a SYN packet to the server to establish connection between them. The value of SYN in this packet will be SYN == 1.

****

The server receives the SYN packet and sends ACK to client to send a message that it has received the SYN packet and is ready to make a connection. During this the client can send data to the server and the server also sends a SYN packet to client to allow it.



After receiving the data successfully, the client will then send an ACK flag as packet ACK: 1 to the server as a confirmation.



Now to terminate the connection, TCP uses a 4- way handshake.

After the transmission of data is completed, the clients sends FIN packet to the server to terminate the connection and waits until server acknowledges it.



The server then acknowledges the closing request for the client side and sends an ACK packet. Till now, the client is in waiting state which can be said as FIN\_WAIT\_2.

The server then sends the FIN packet to terminate the connection.



After receiving the FIN and the ACK packet from the server, the client sends an acknowledgement which means that the connection termination process is completed.

