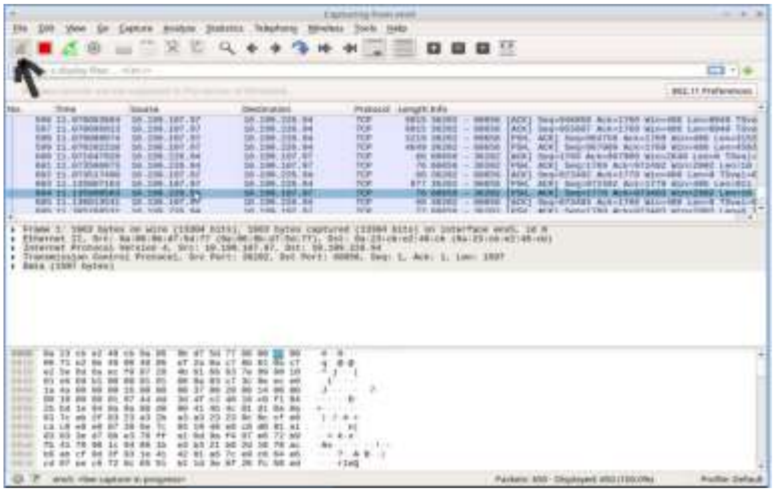
Wireshark Packet Capturing and Filtering.

Project:

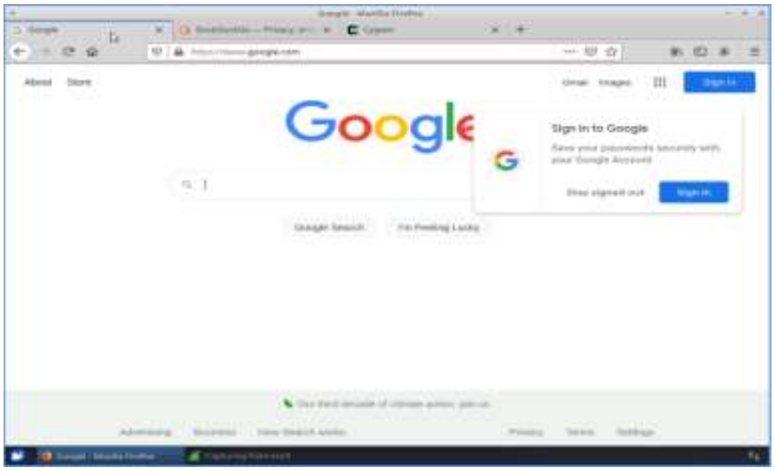
The task was to use Wireshark on Ubuntu to create a capture file and then use a display filter to list all HTTPS and HTTP packets then eliminate one IP address from the capture using a display filter. This was performed in a control environment.

1. Start a packet capture on the ethernet in Wireshark. It is important to note before starting packet capturing in Wireshark it is advisable to clear the cache in the browser you are going to use so as to get fresh website data.

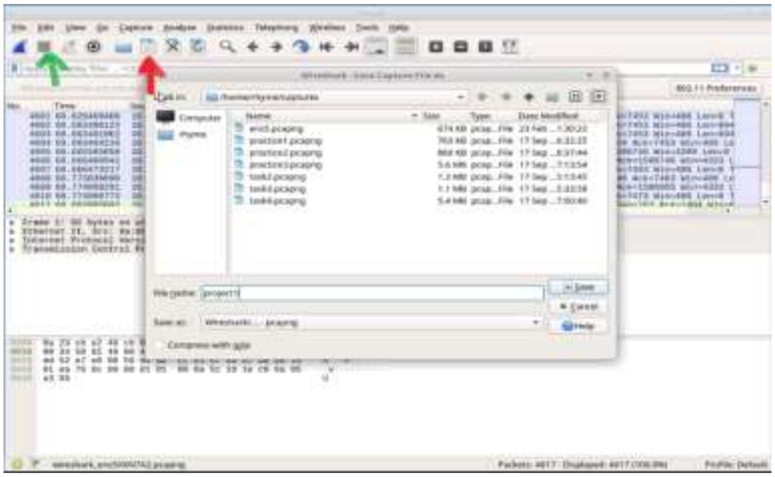


The packet capture in Wireshark begins once you tap the “blue shark fin” icon, indicated by the black arrow.

2. Visit sites (https://google.com, https://duckduckgo.com and http://cygwin.com) on the browser to create traffic.



3. Stopping the packet capture and saving it to a file (project1)

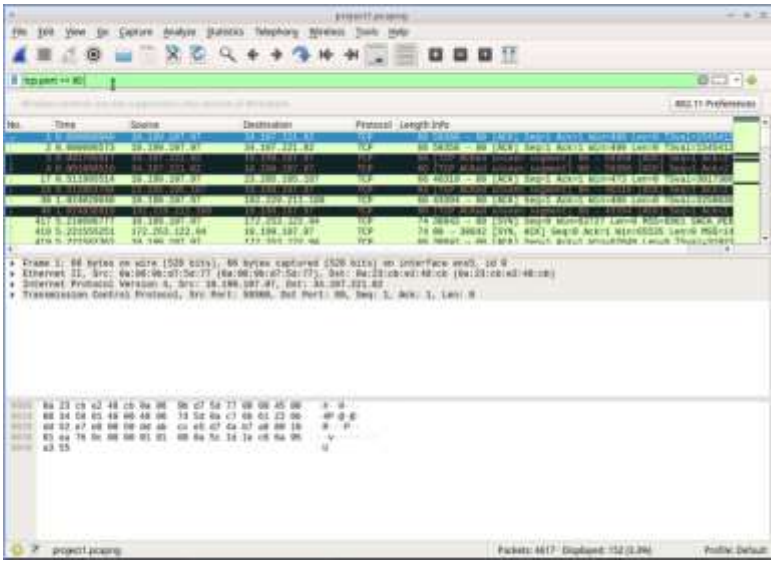


File name: project1

Save as: Wireshark/... - pcapng

The packet capture was stopped (icon indicated by the green arrow) and saved (icon indicated by the red arrow) as a file (project1).

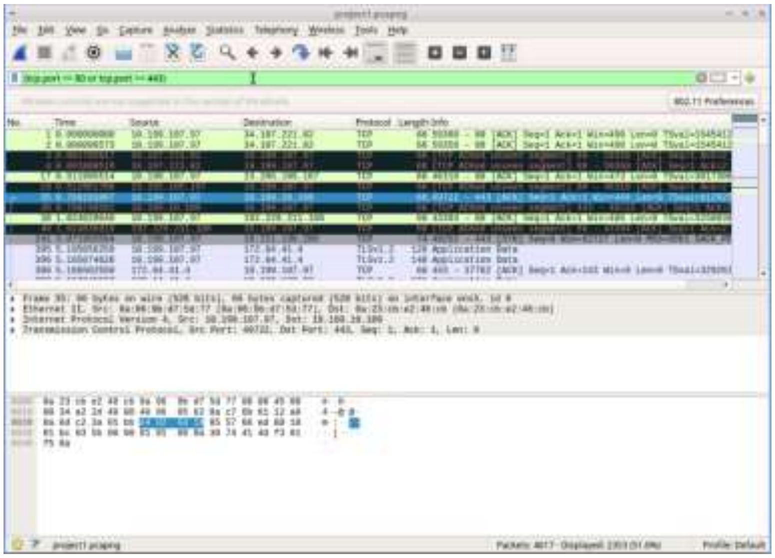
4. Creating a filter to just display port 80 TCP data on the saved file.



tcp.port == 80

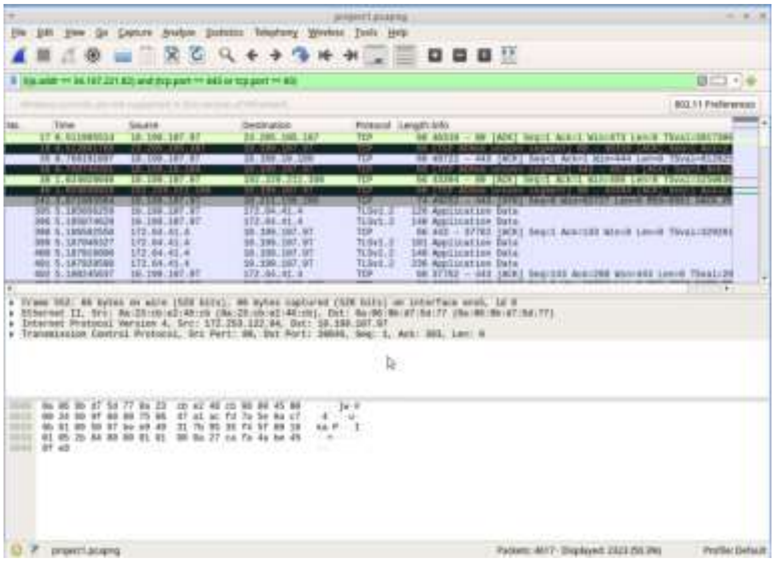
The task specified to create a filter that displayed port 80 TCP data. Port 80 is used for HTTP web traffic which in this scenario was mostly traffic for **34.107.221.82** IP address (http://cygwin.com). So the display filter was set to "**tcp.port == 80**" which will filter for any traffic that goes through port 80.

5. Creating a filter to only display HTTP and HTTPS packets



The display filter is now set to display only HTTP and HTTPS packets. We know already know that port 80 is used for HTTP web traffic and similarly port 443 is used for HTTPS web traffic. The display filter condition was set to **“(tcp.port == 80 or tcp.port == 443)”** to satisfy this request. The condition is placed in parentheses with “**OR**” so that both conditions are met.

6. Eliminating the Cygwin site visits from the displayed packets.



This task specified that we eliminate one IP address from the displayed packets. This was accomplished by using the **(!)** exclamation mark which notifies Wireshark to excluded any traffic from the specified IP address **(34.107.221.82)** and the “**AND**” was added to show what was to be included in the display, which was all other HTTP and HTTPS traffic. The resulting condition was **“!(ip.addr == 34.107.221.82) and (tcp.port == 443 or tcp.port == 80)”.**