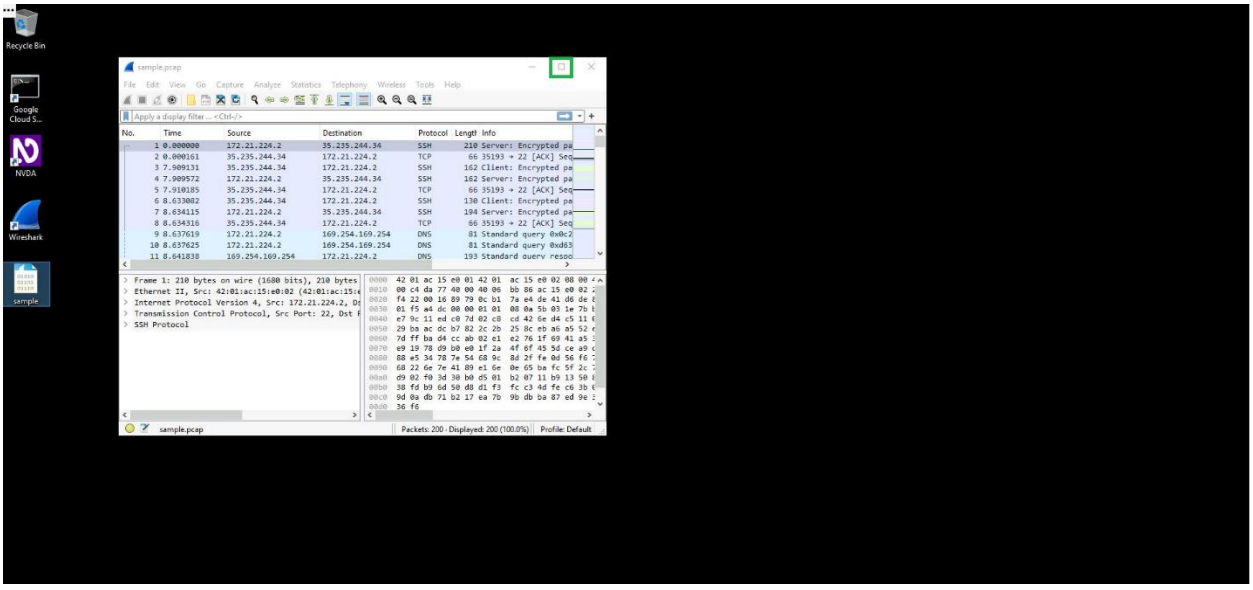
Packet Analysis using Wireshark

In this project I boot up a Windows VM which has Wireshark installed and go through the various ways that internet traffic can be analyzed using this application. This packet analyzer is particularly user-friendly because, unlike Tcpdump, it has a graphical user interface (GUI).

Exploring data with Wireshark

After booting up the Windows VM, We click on the *.pcap* sample packet capture file that that has been prepared on the desktop.

This then opens up the Wireshark application window which can be maximized for a better experience.



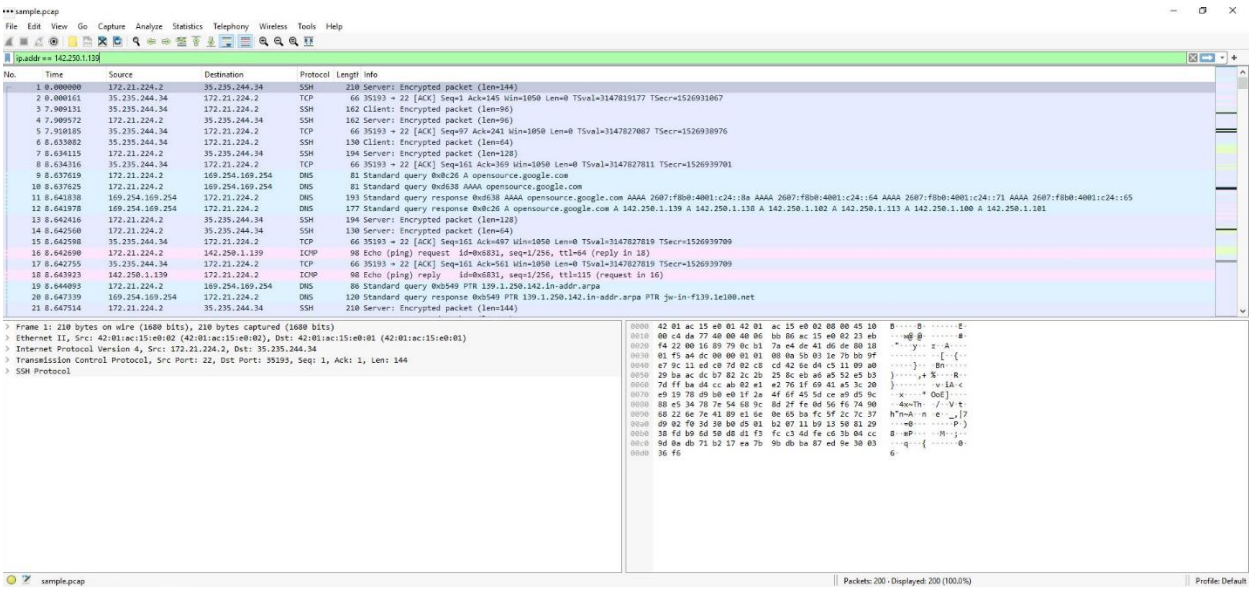
The key property columns listed for each packet can be examined:

- **No.** : The index number of the packet in this packet capture file
- **Time**: The timestamp of the packet
- **Source**: The source IP address
- **Destination**: The destination IP address
- **Protocol**: The protocol contained in the packet
- **Length**: The total length of the packet
- **Info**: Some information about the data in the packet (the payload) as interpreted by Wireshark

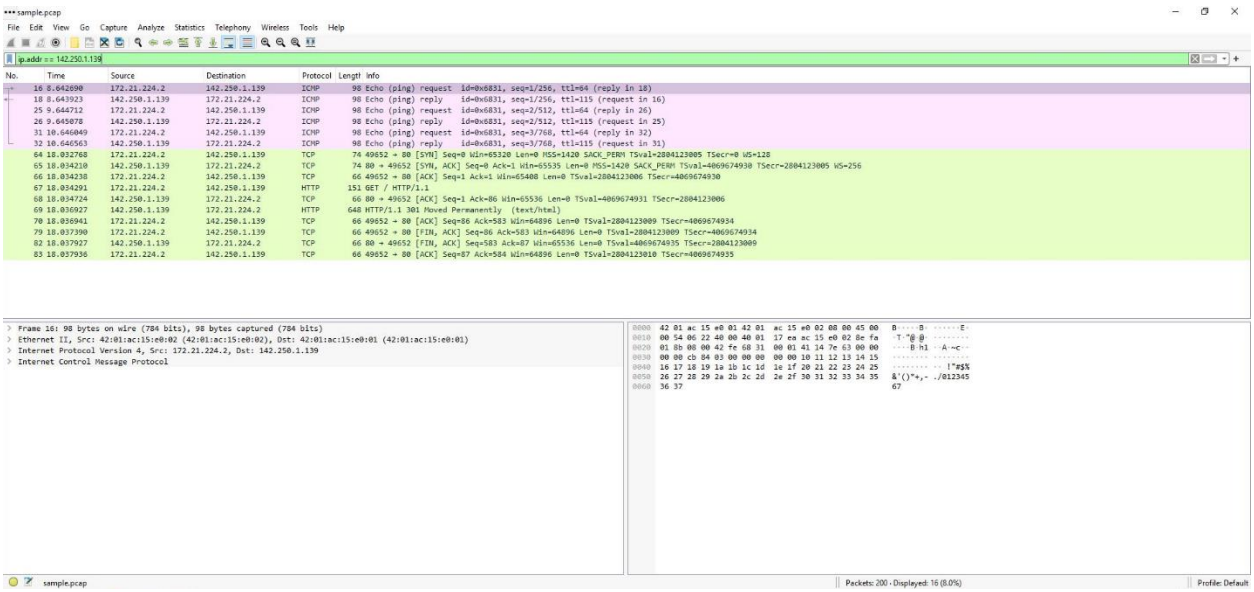
Packets are also color-coded based on protocol type. These coloring rules are customizable and changes can be both temporary (valid until the application is closed) or permanent.

Applying Wireshark filters and inspecting packets

The first filter that could be applied to the data is *ip.addr ==*. In this example we will type "ip.addr == 142.250.1.139" into the "**Apply a display filter …**" text box.

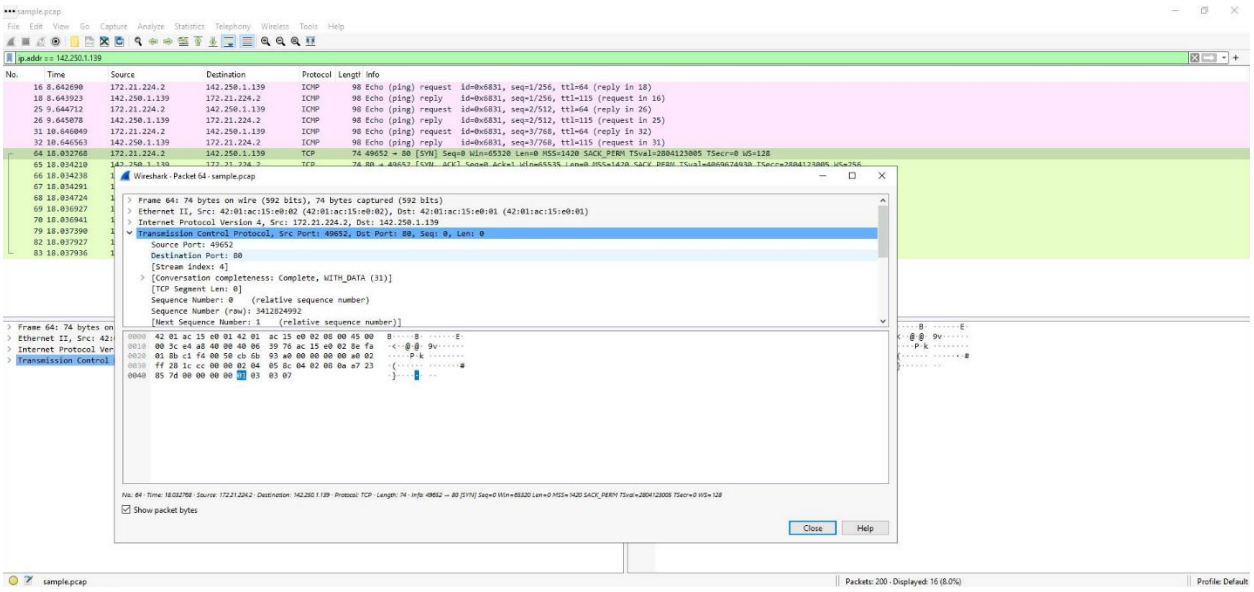


This results in a list of only the packets which have the same source or destination IP address as the one that was entered after the double equal sign.



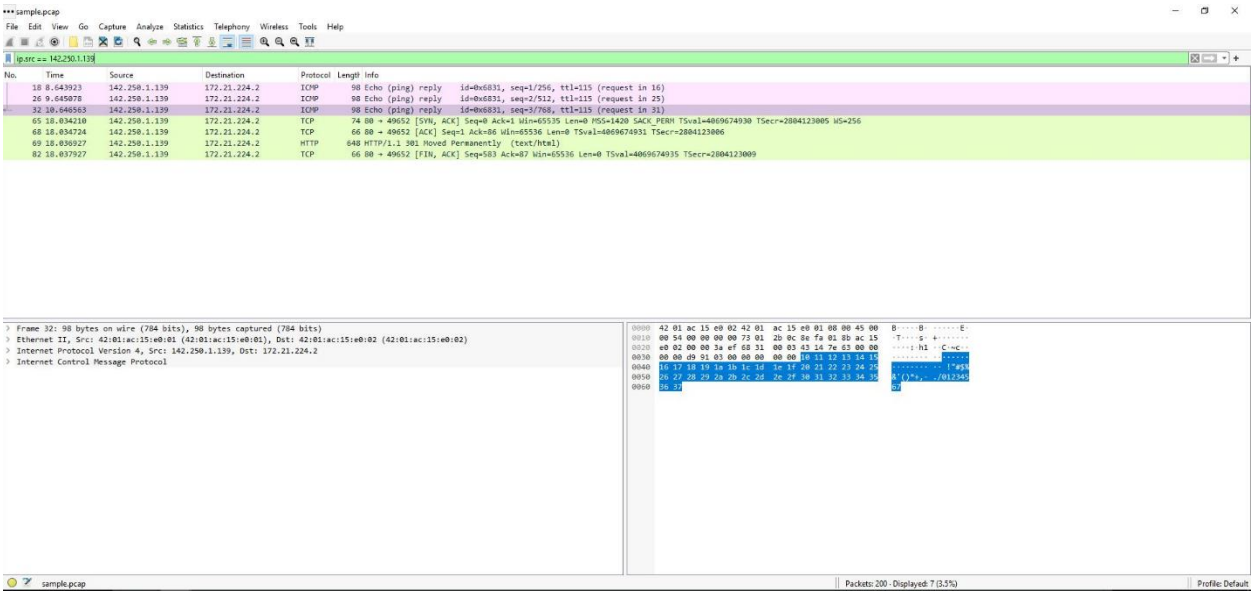
Now only ICMP and TCP traffic is visible (HTTP runs over TCP, which is why it also shows up and is under the same color rule).

If, for example, the first TCP packet is selected and double-clicked, the TCP destination port of the packet can be seen under the **Transmission Control Protocol** subtree. In this case the destination port is 80.

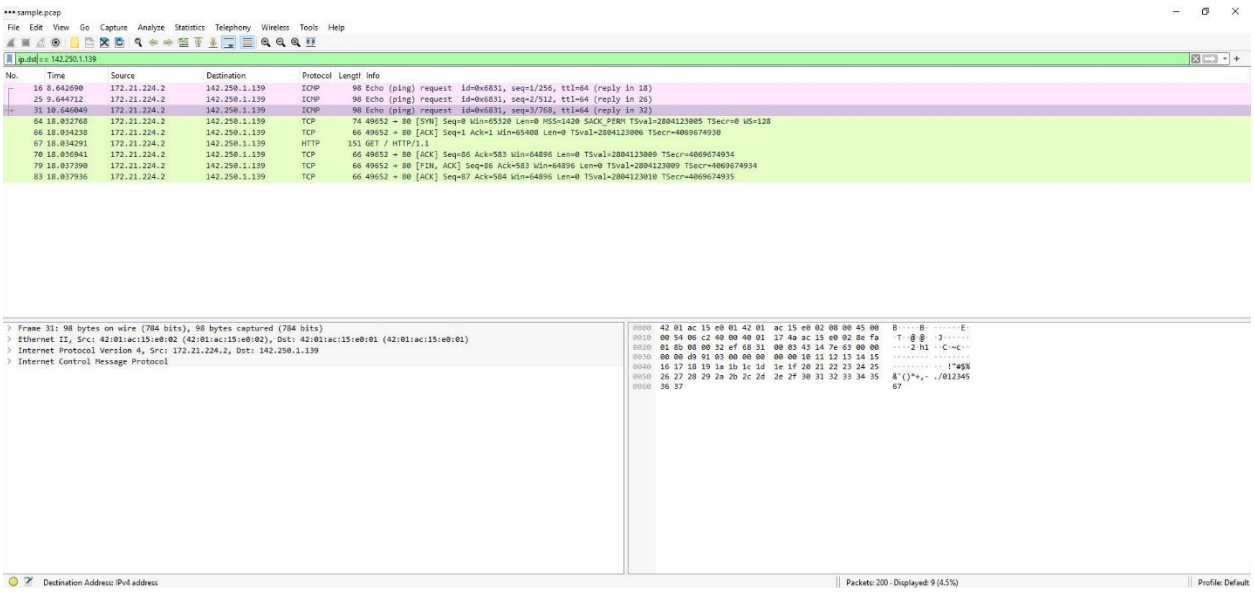


Selecting packets

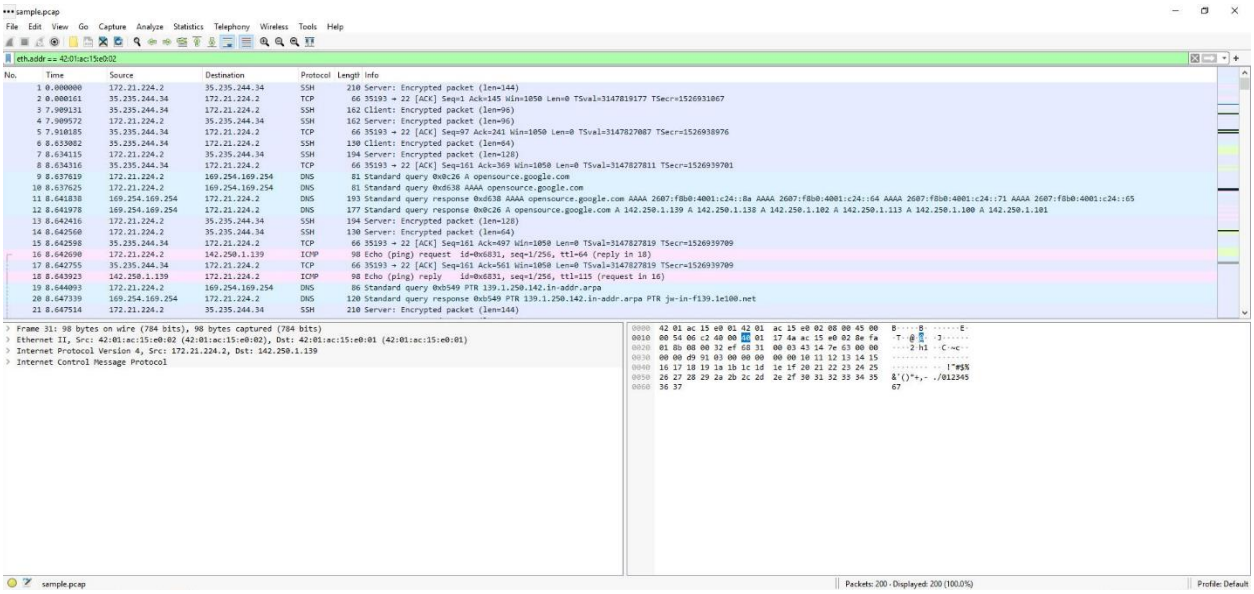
For more granular search results we could specifically filter by source or destination IP address. Applying the “ip.src == 142.250.1.139” display filter returns only entries for packets that came from the IP 142.250.1.139.



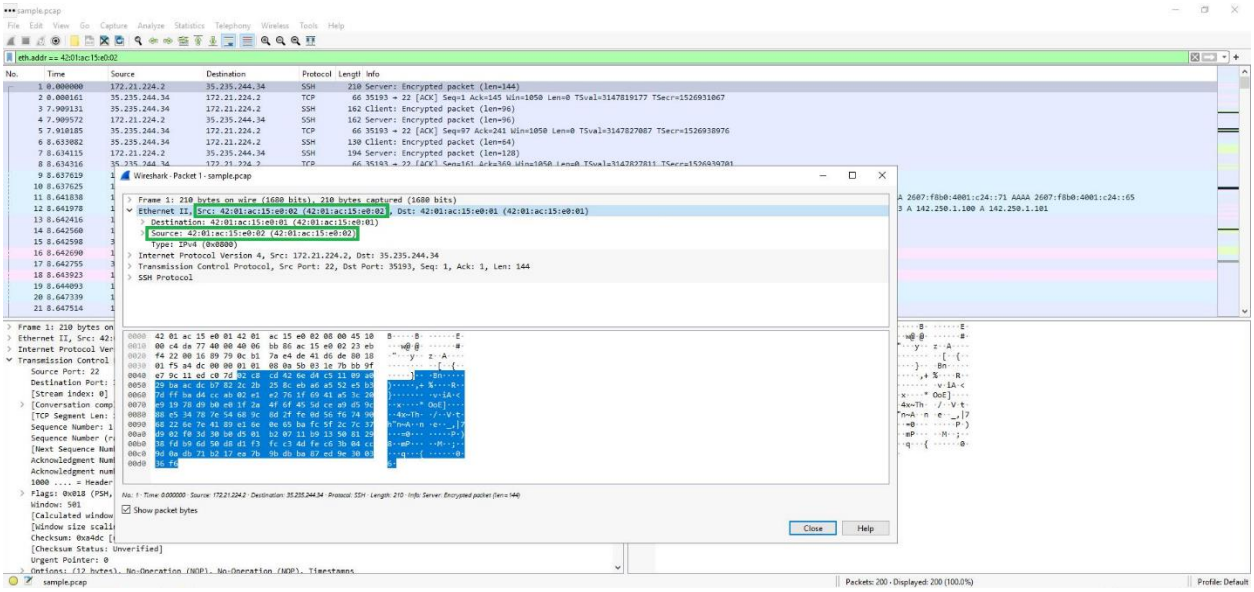
Likewise, applying the “ip.dst == 142.250.1.139” display filter will return only entries for packets that were sent to the IP 142.250.1.139.



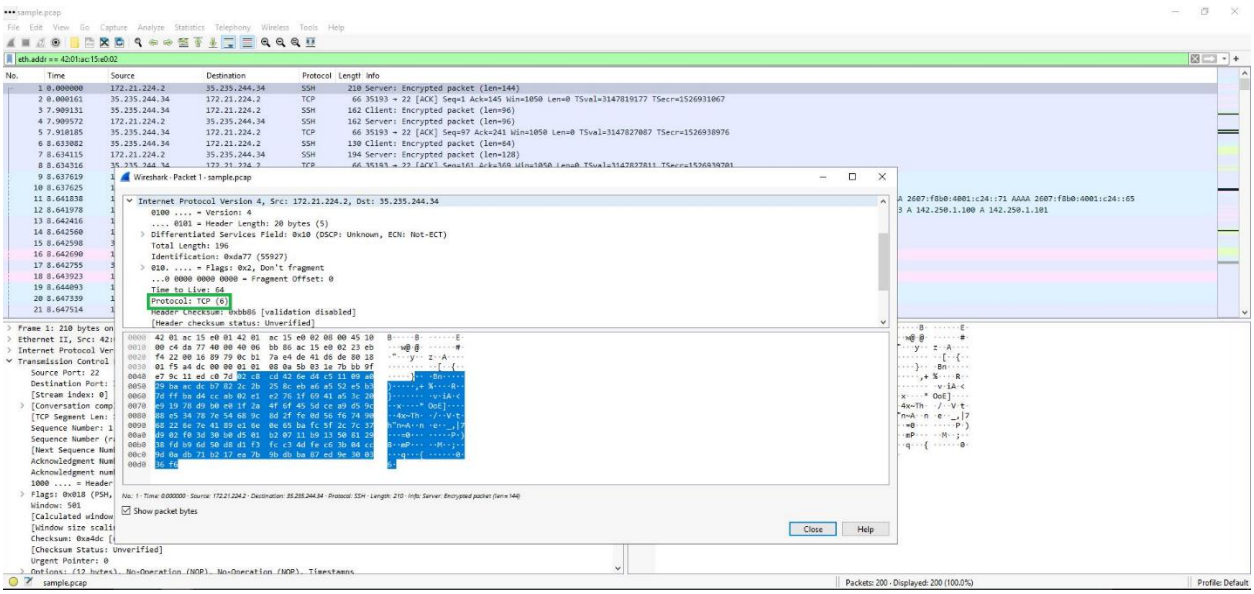
Results can also be filtered based a specific MAC address. Typing “eth.addr == 42:01:ac:15:e0:02” into the search box will result in all traffic related to this MAC address, regardless of the protocols which are involved.



Here, if we open the first packet in the results and expand the **Ethernet II** subtree, we can see that the specified MAC address is the source of the packet.



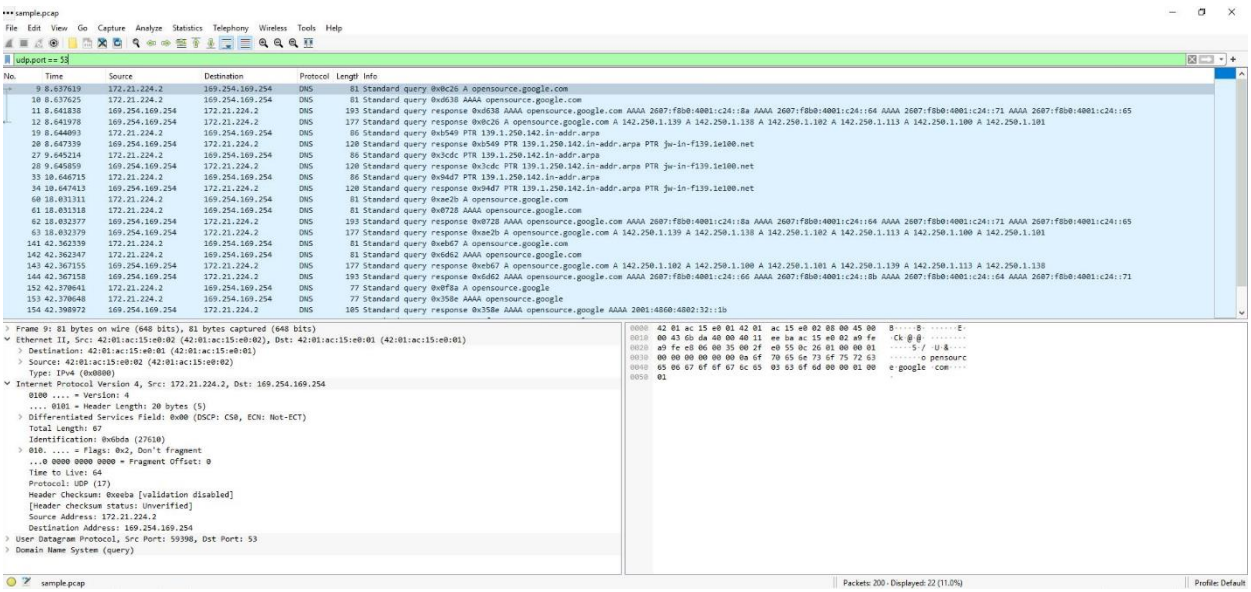
Additionally, expanding the **Internet Protocol Version 4** subtree reveals the internal protocol contained in the packet.



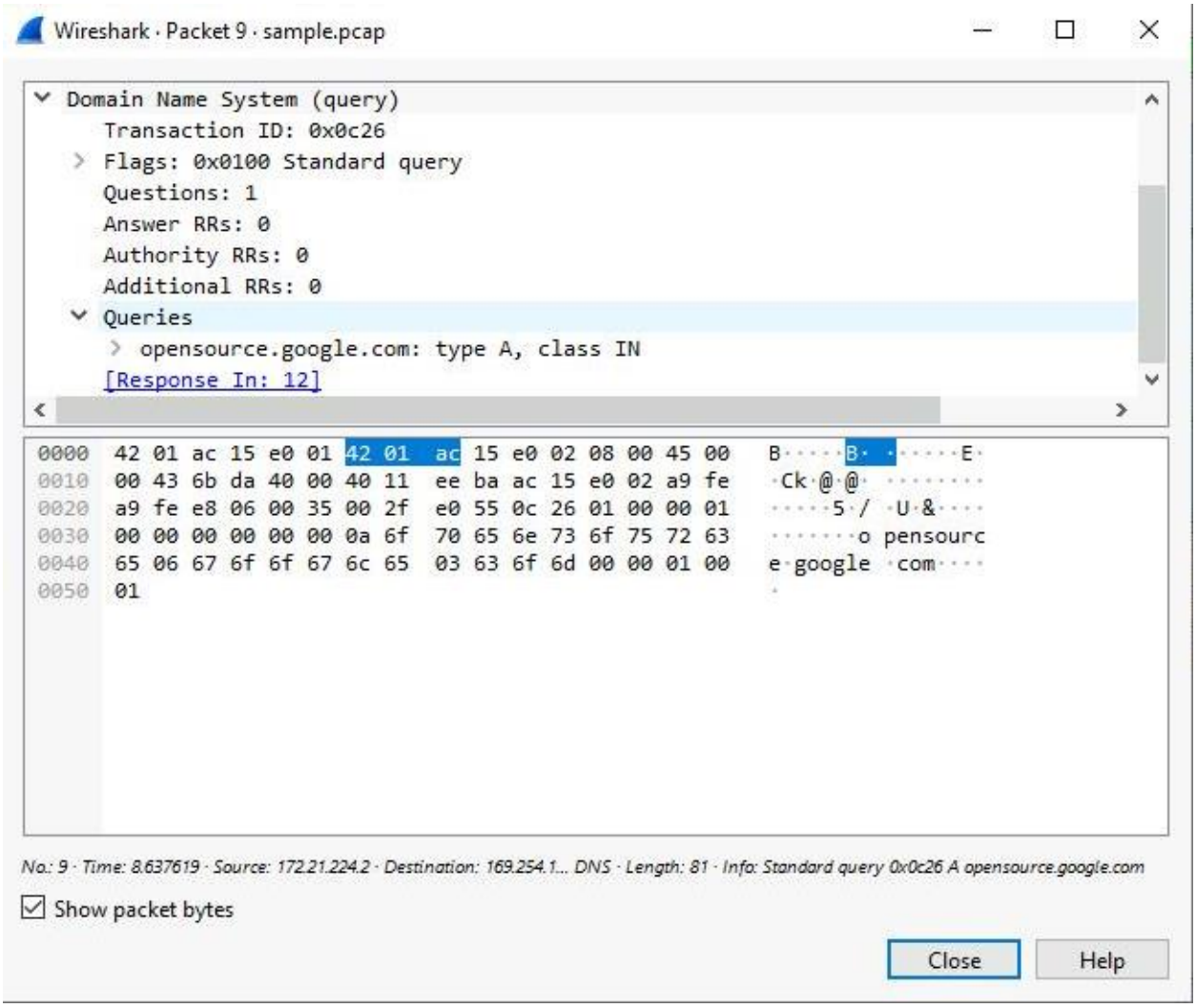
Exploring DNS packets

DNS traffic can also be examined with Wireshark. In this part of the project, I will demonstrate how to find DNS packet data such as queries (names of websites that are being looked up) and answers (IP addresses that are being sent back by a DNS server when a name is successfully resolved).

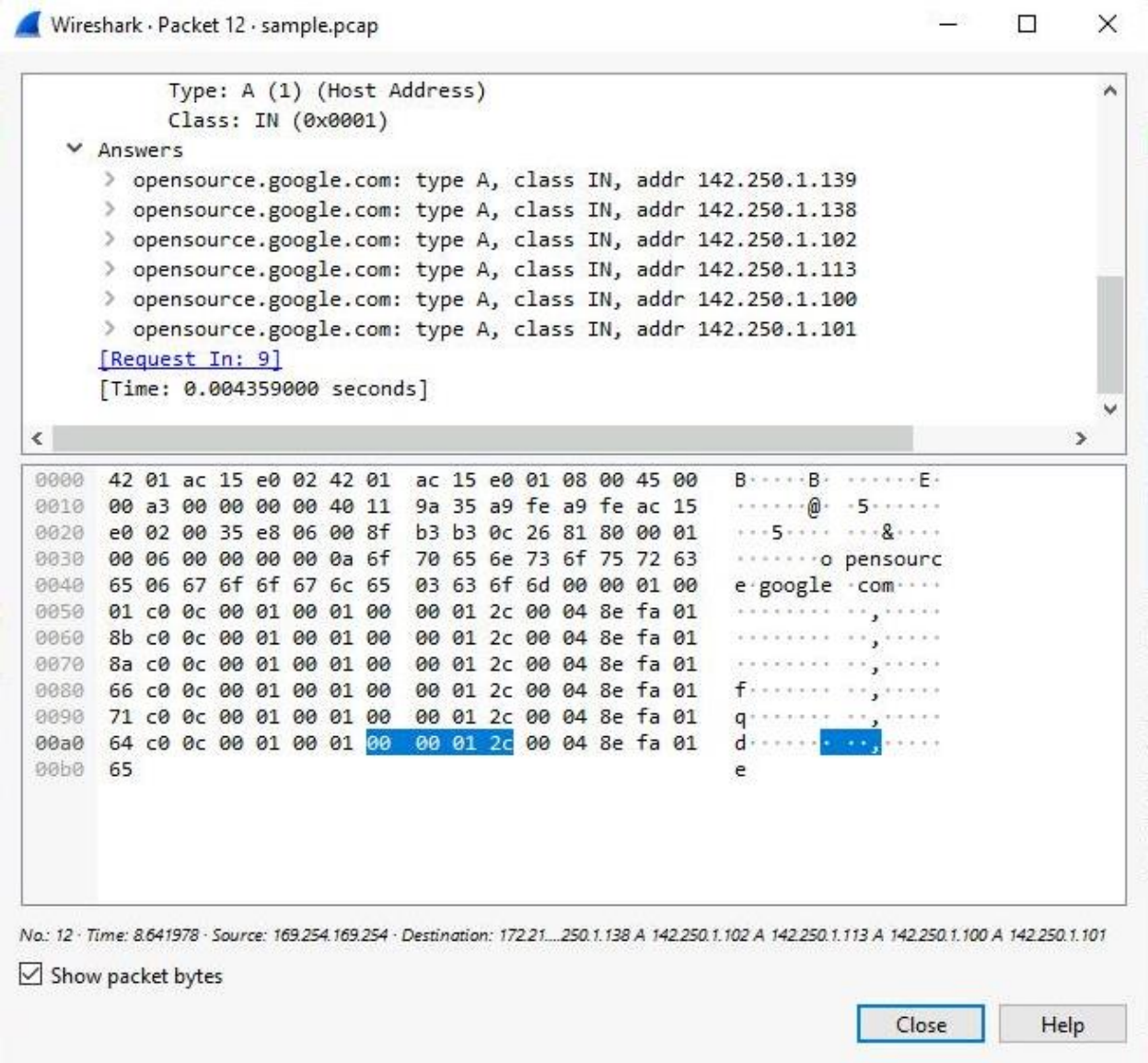
Since DNS traffic is over UDP port 53, the filter that needs to be applied here will be “udp.port == 53”. This lists only DNS queries and responses.



If the first packet is opened, under the **Domain Name System (query)** subtree, if **Queries** is clicked on we see that the name of the website which was queries is **opensource.google.com**.

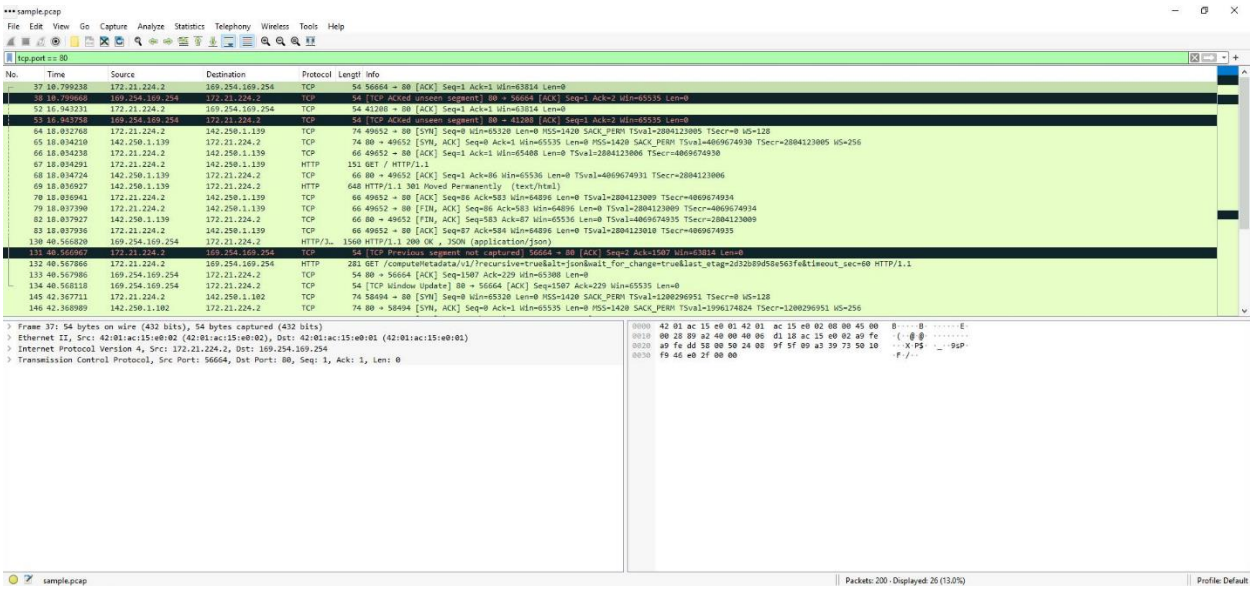


If the same is done when examining the 4th packet in the list and we scroll down and double-click on **Answers** under the **Domain Name System (query)**, we can see the name that was queried (**opensource.google.com**) and the addresses that are associated with that name.



Exploring TCP packets

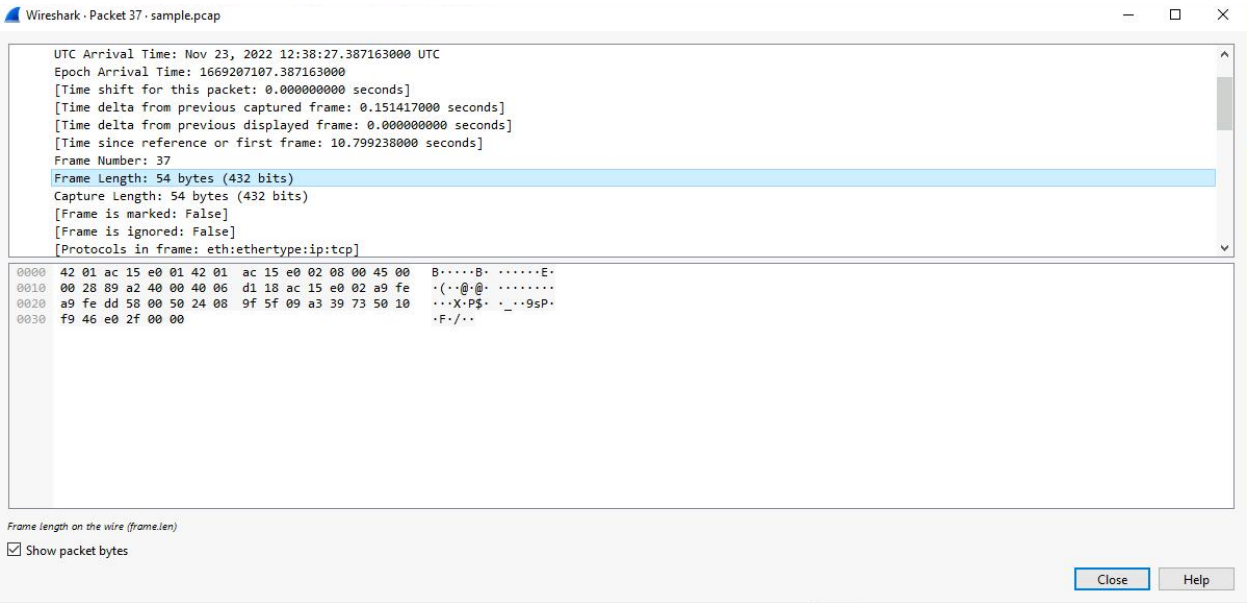
To examine TCP traffic, we start by applying the display filter “tcp.port == 80”.



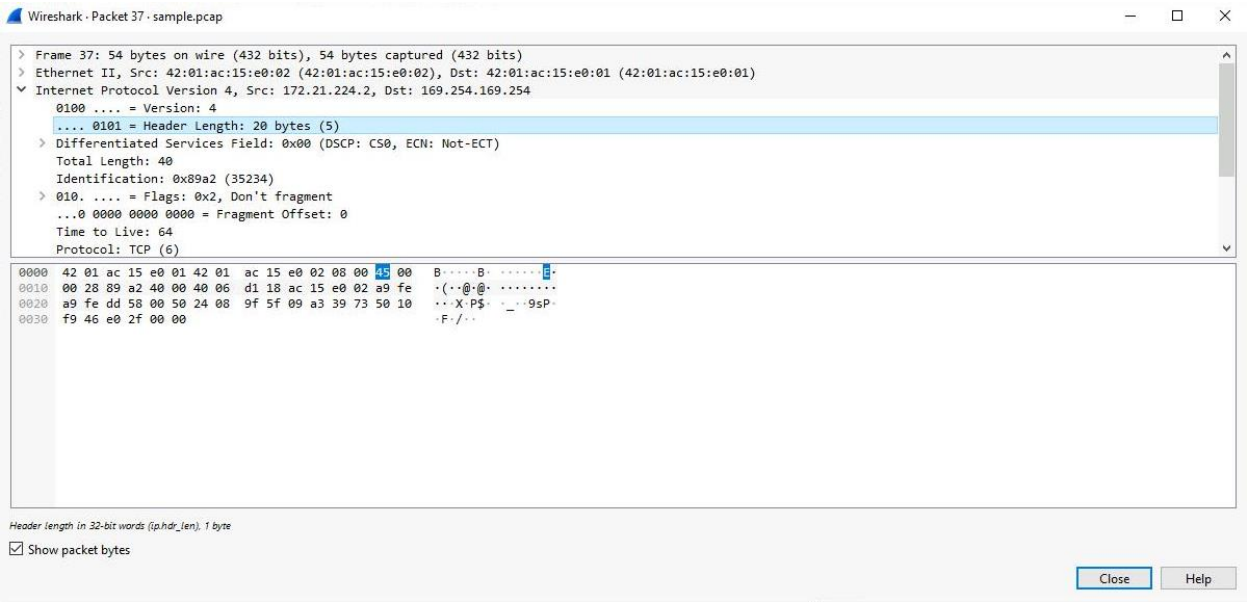
After clicking on the first packet with a destination IP of **169.254.169.254**, we can find various information about the packet such as:

1. The **Time to Live** value which can be found under the **Internet Protocol Version 4** subtree

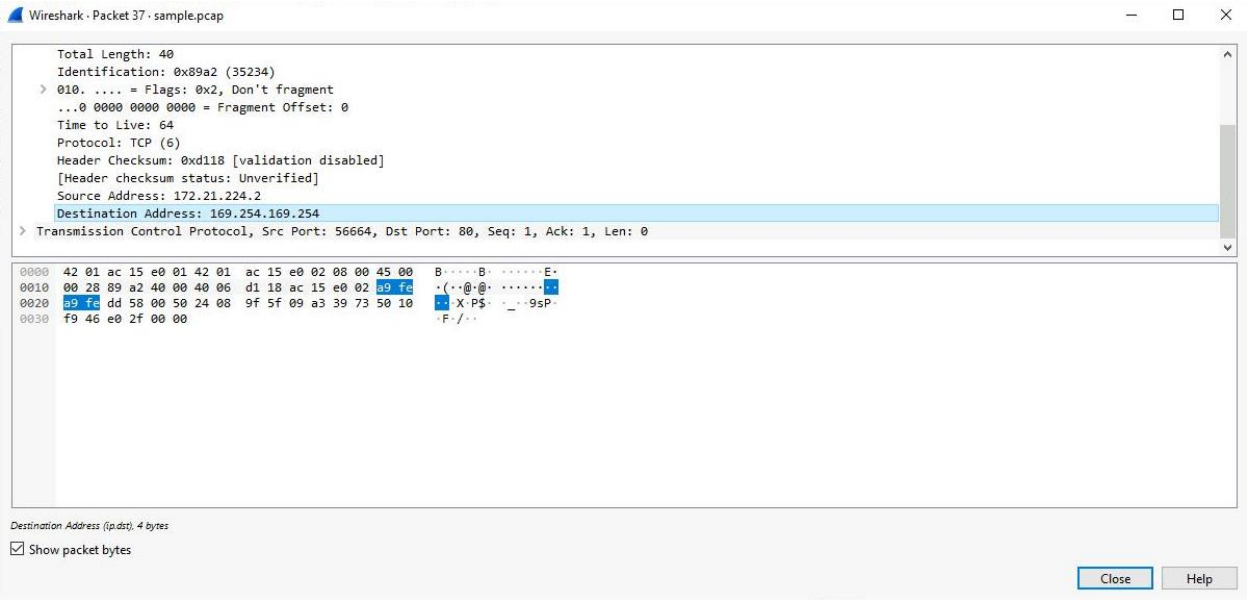
2. The **Frame Length**, found under the **Frame** subtree



3. The **Header Length**, found under the **Internet Protocol Version 4** subtree



4. We can see the Destination IP address here as well by looking under the **Internet Protocol Version 4** subtree



Wireshark also allows us to filter results based on specific text data. This can be done by typing “tcp contains” followed by the word we are searching for written within quotation marks. In this example, we are looking for web requests made with the “*curl*” command.