Analyze your first packet with Wireshark

Scenario

In this scenario, you're a security analyst investigating traffic to a website. You'll analyze a network packet capture file that contains traffic data related to a user connecting to an internet site. The ability to filter network traffic using packet sniffers to gather relevant information is an essential skill as a security analyst.

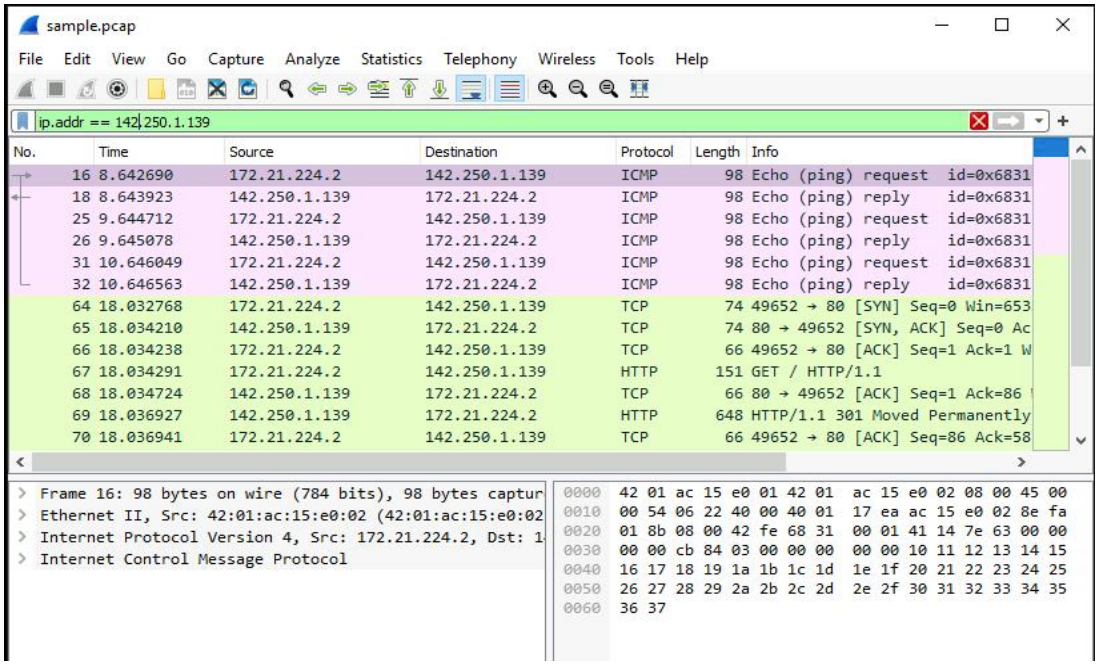
You must filter the data in order to:

- identify the source and destination IP addresses involved in this web browsing session,
- examine the protocols that are used when the user makes the connection to the website, and
- analyze some of the data packets to identify the type of information sent and received by the systems that connect to each other when the network data is captured.

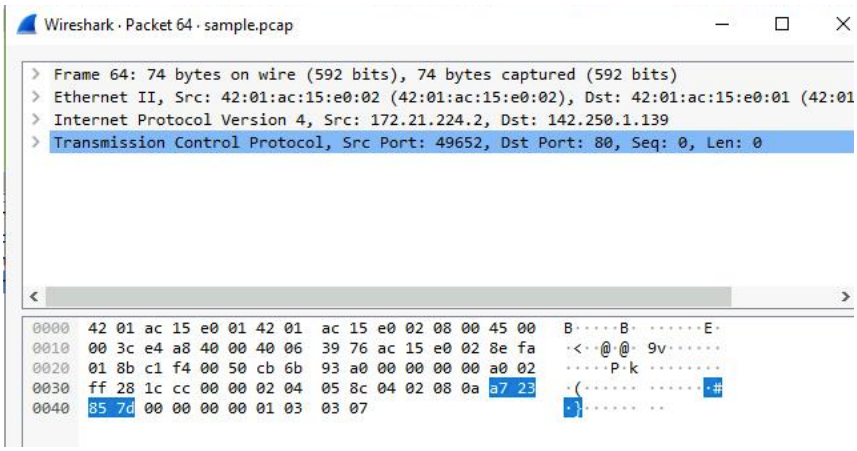
Task 1. Apply a basic Wireshark filter and inspect a packet

1.1.Filter for traffic associated with a specific IP address.

ip.addr == 142.250.1.139

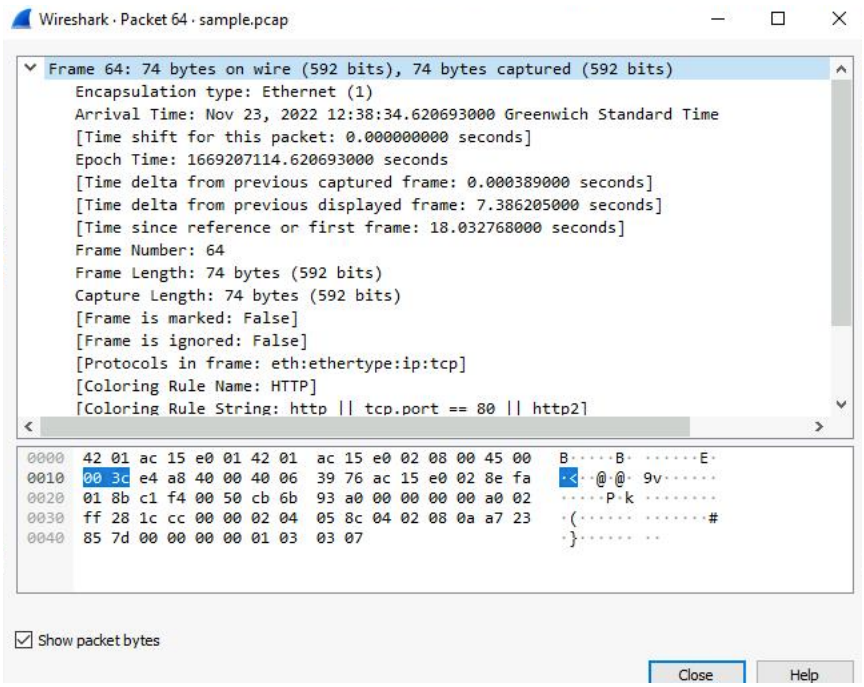


1.2.Double-click the first packet that lists **TCP** as the protocol.



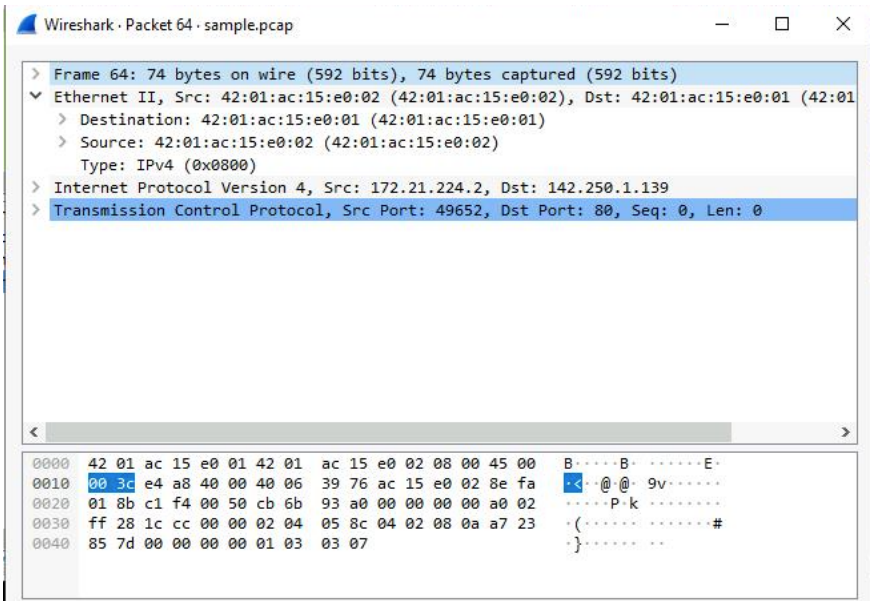
The upper section of this window contains subtrees where Wireshark will provide you with an analysis of the various parts of the network packet. The lower section of the window contains the raw packet data displayed in hexadecimal and ASCII text. There is also placeholder text for fields where the character data does not apply, as indicated by the dot (“.”).

1.3.Double-click the first subtree in the upper section. This starts with the word **Frame**.



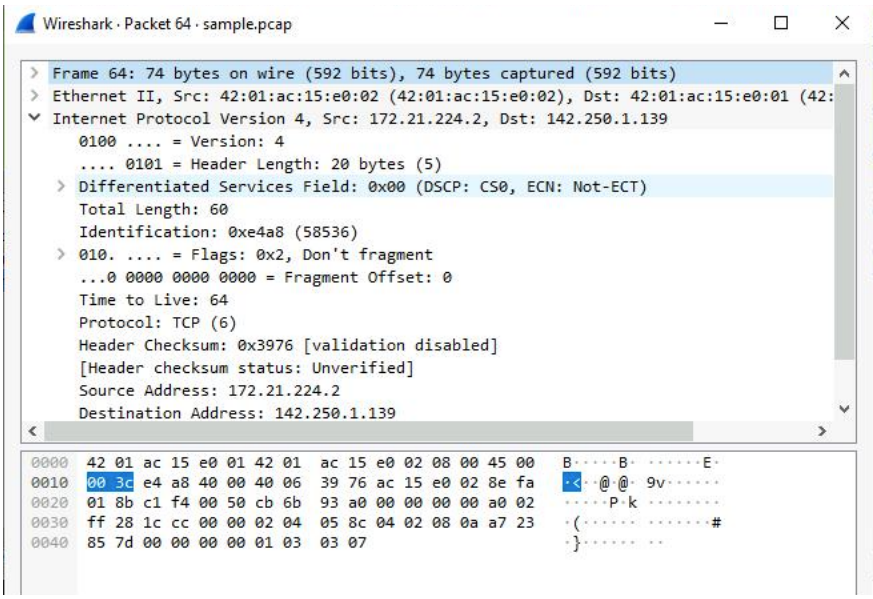
This provides you with details about the overall network packet, or frame, including the frame length and the arrival time of the packet. At this level, you’re viewing information about the entire packet of data.

1.4. Double-click **Frame** again to collapse the subtree and then double-click the **Ethernet II** subtree.



This item contains details about the packet at the Ethernet level, including the source and destination MAC addresses and the type of internal protocol that the Ethernet packet contains.

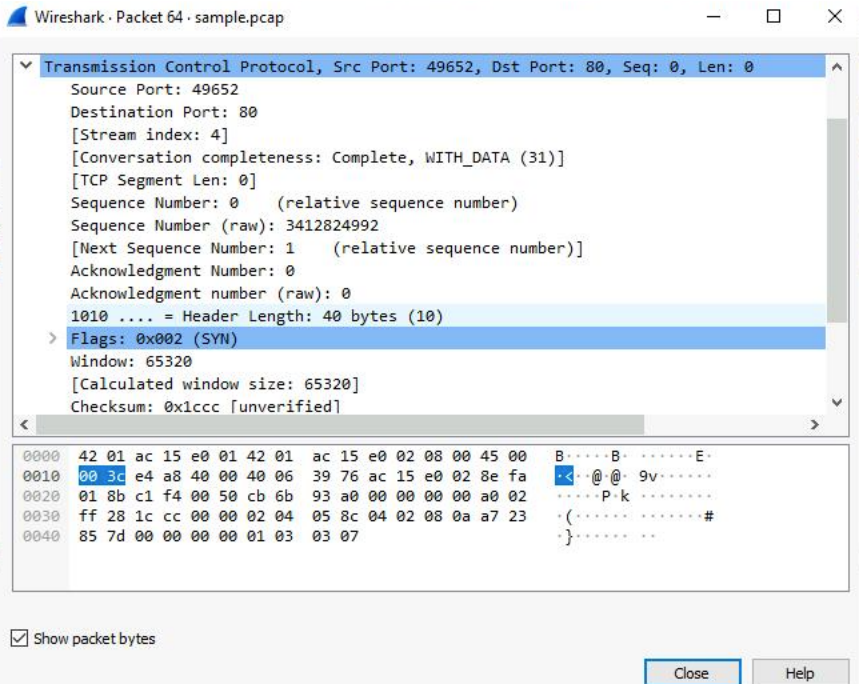
1.5. Double-click **Ethernet II** again to collapse that subtree and then double-click the **Internet Protocol Version 4** subtree.



This provides packet data about the Internet Protocol (IP) data contained in the Ethernet packet. It contains information such as the source and destination IP addresses and the Internal Protocol (for example, TCP or UDP), which is carried inside the IP packet.

The source and destination IP addresses shown here match the source and destination IP addresses in the summary display for this packet in the main Wireshark window.

1.6. Double-click **Internet Protocol Version 4** again to collapse that subtree and then double-click the **Transmission Control Protocol** subtree.

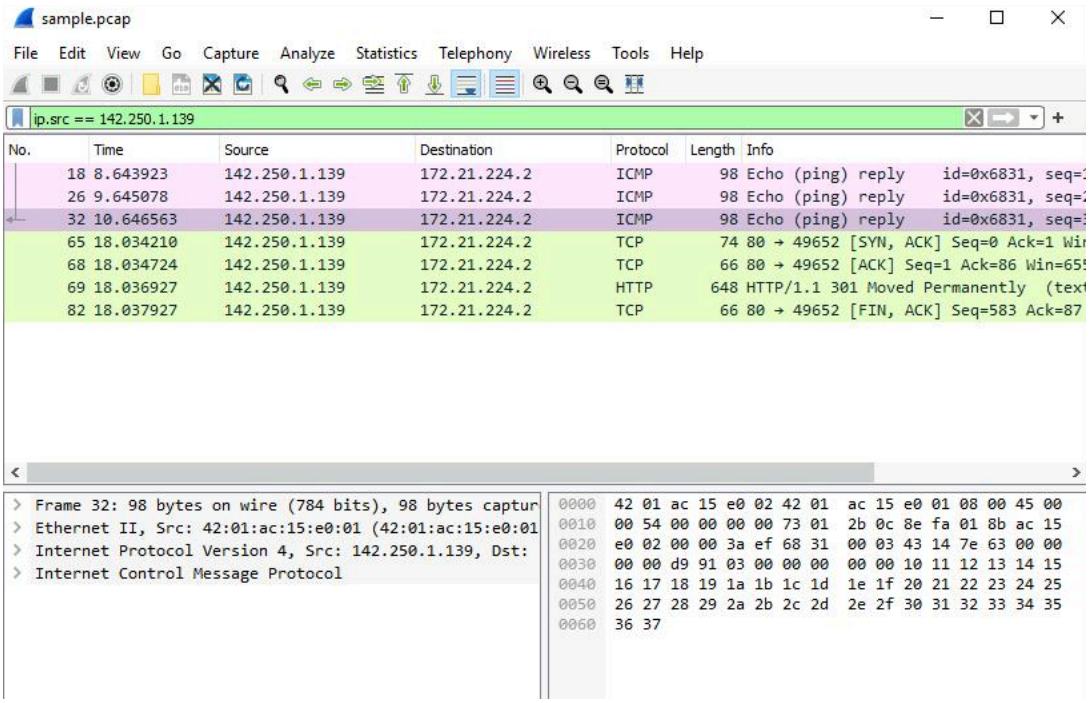


This provides detailed information about the TCP packet, including the source and destination TCP ports, the TCP sequence numbers, and the TCP flags.

Task 2. Use filters to select packets

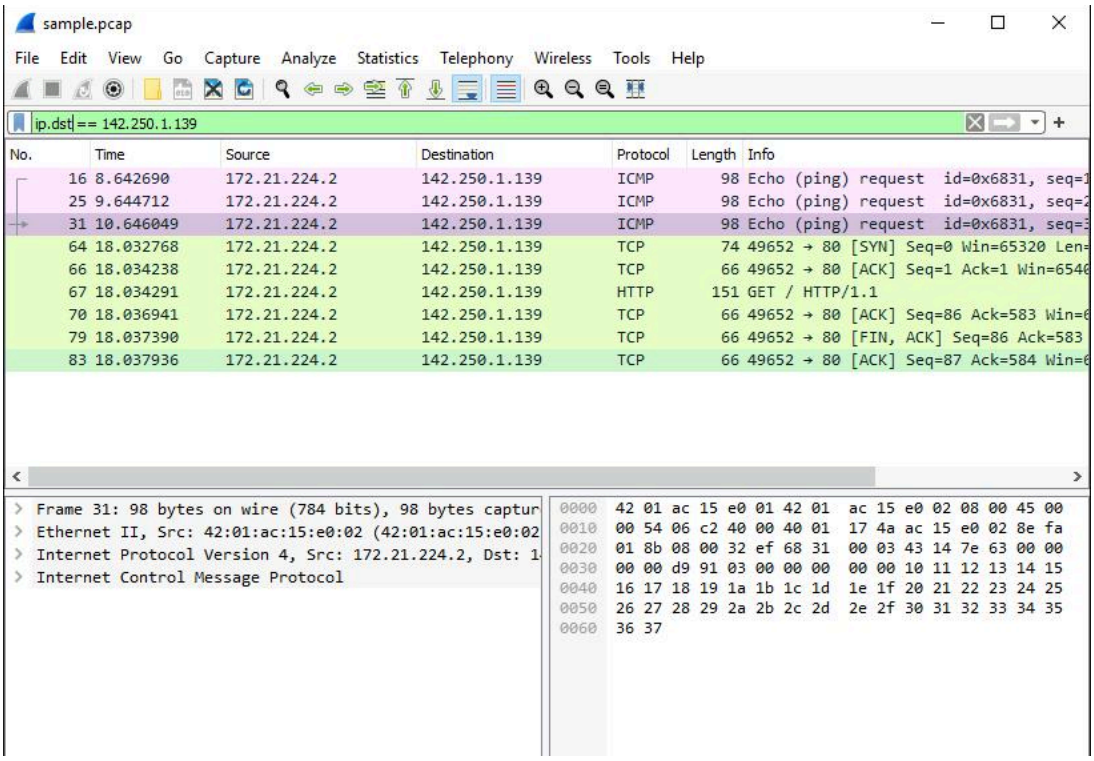
2.1. Filter to select traffic for a specific source IP address only.

ip.src == 142.250.1.139



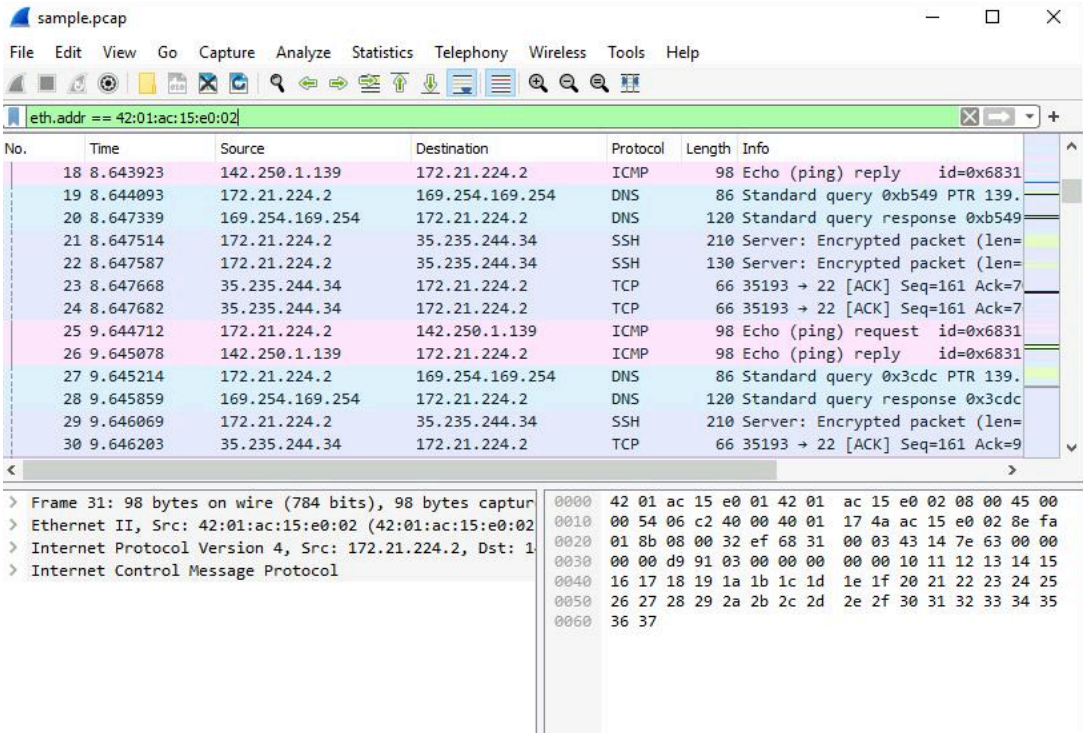
2.2. Filter to select traffic for a specific destination IP address only.

ip.dst == 142.250.1.139



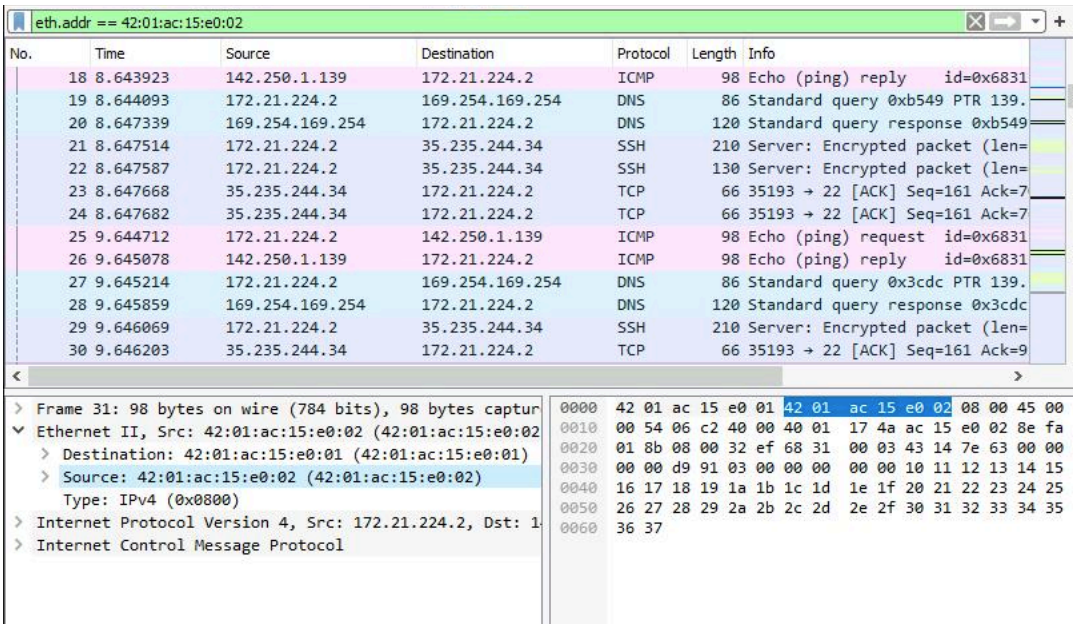
2.3. Filter to select traffic to or from a specific Ethernet MAC address. This filters traffic related to one MAC address, regardless of the other protocols involved:

eth.addr == 42:01:ac:15:e0:02

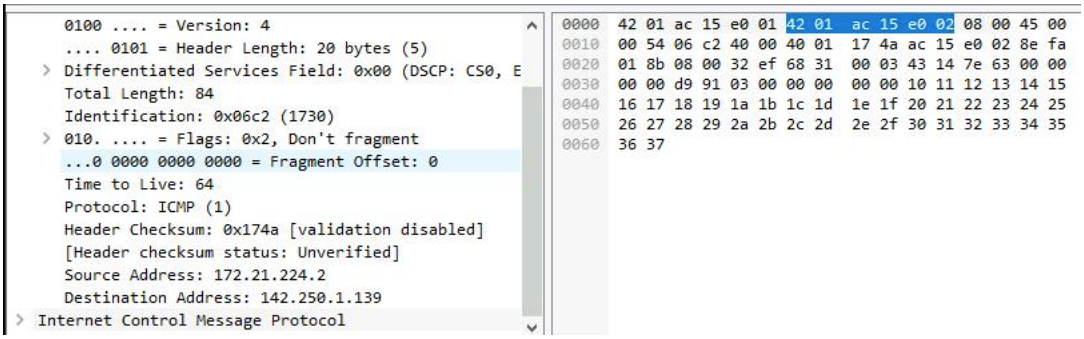


2.4. Double-click the **Ethernet II** subtree if it is not already open.

The MAC address you specified in the filter is listed as either the source or destination address in the expanded Ethernet II subtree.



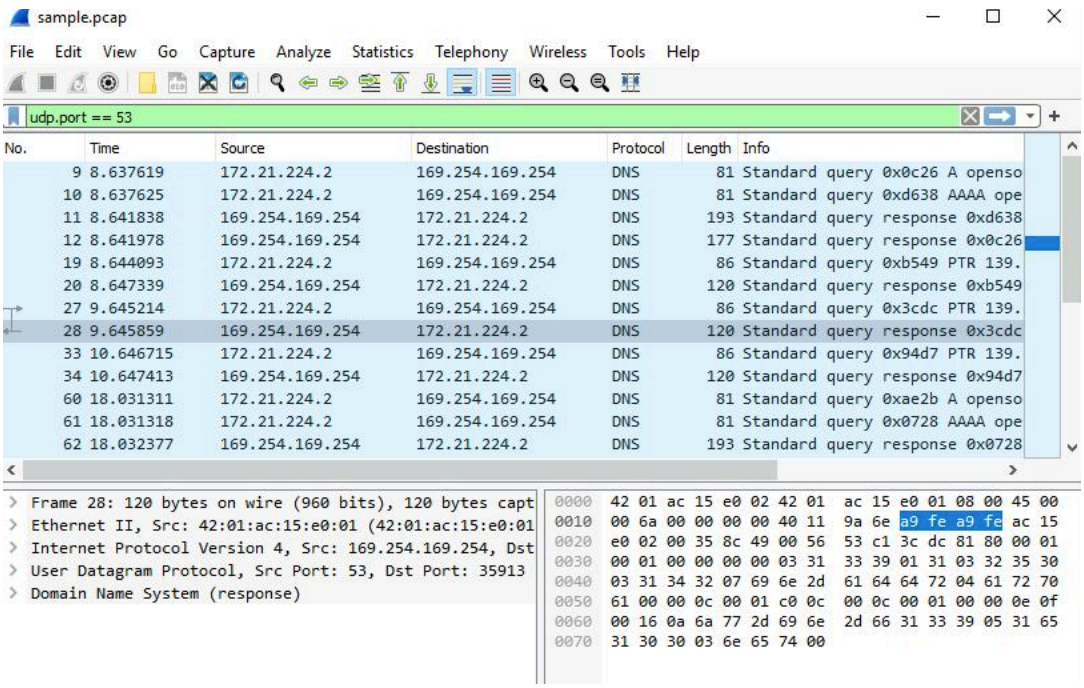
2.5. Double-click the **Internet Protocol Version 4** subtree to expand it and scroll down until the **Time to Live** and **Protocol** fields appear.



Task 3. Use filters to explore DNS packets

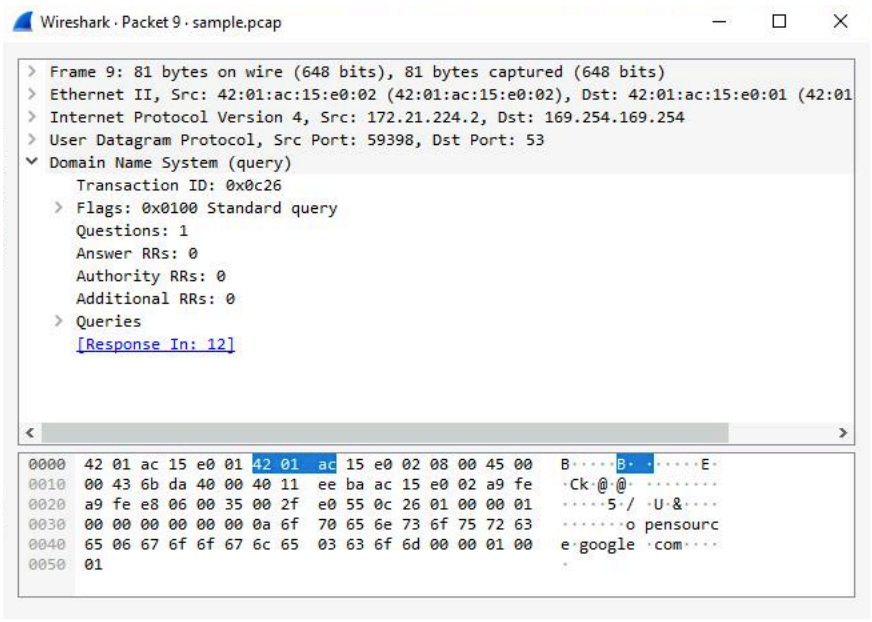
3.1. Filter to select UDP port 53 traffic. DNS traffic uses UDP port 53, so this will list traffic related to DNS queries and responses only

udp.port == 53

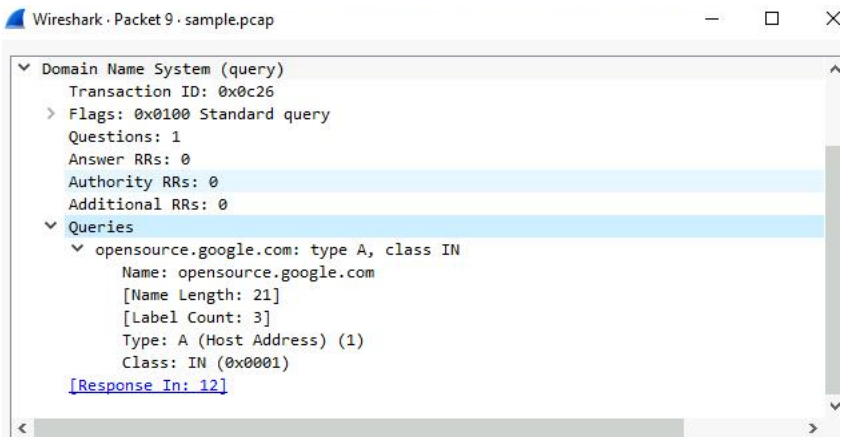


3.2. Double-click the first packet in the list to open the detailed packet window.

3.3. Scroll down and double-click the **Domain Name System (query)** subtree to expand it.



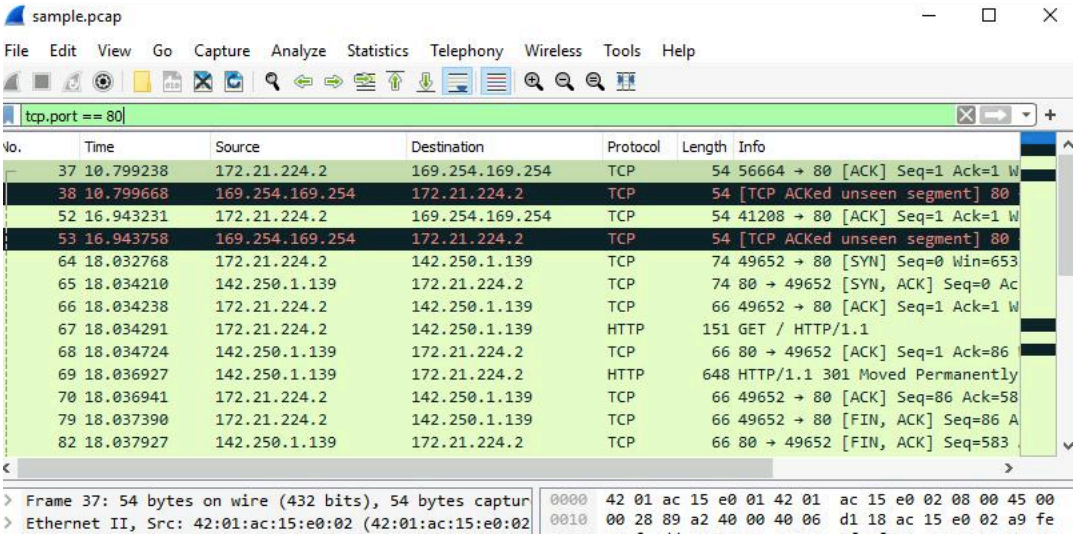
3.4. Scroll down and double-click **Queries**.



Task 4. Use filters to explore TCP packets

4.1. Filter to select TCP port 80 traffic. TCP port 80 is the default port that is associated with web traffic:

tcp.port == 80



4.2. Filter to select TCP packet data that contains specific text data.

tcp contains "curl”