Analyze a Packet With Wireshark

Skills Acquired:

- Open saved packet capture files
- View high-level packet data
- Use filters to inspect detailed packet data

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:

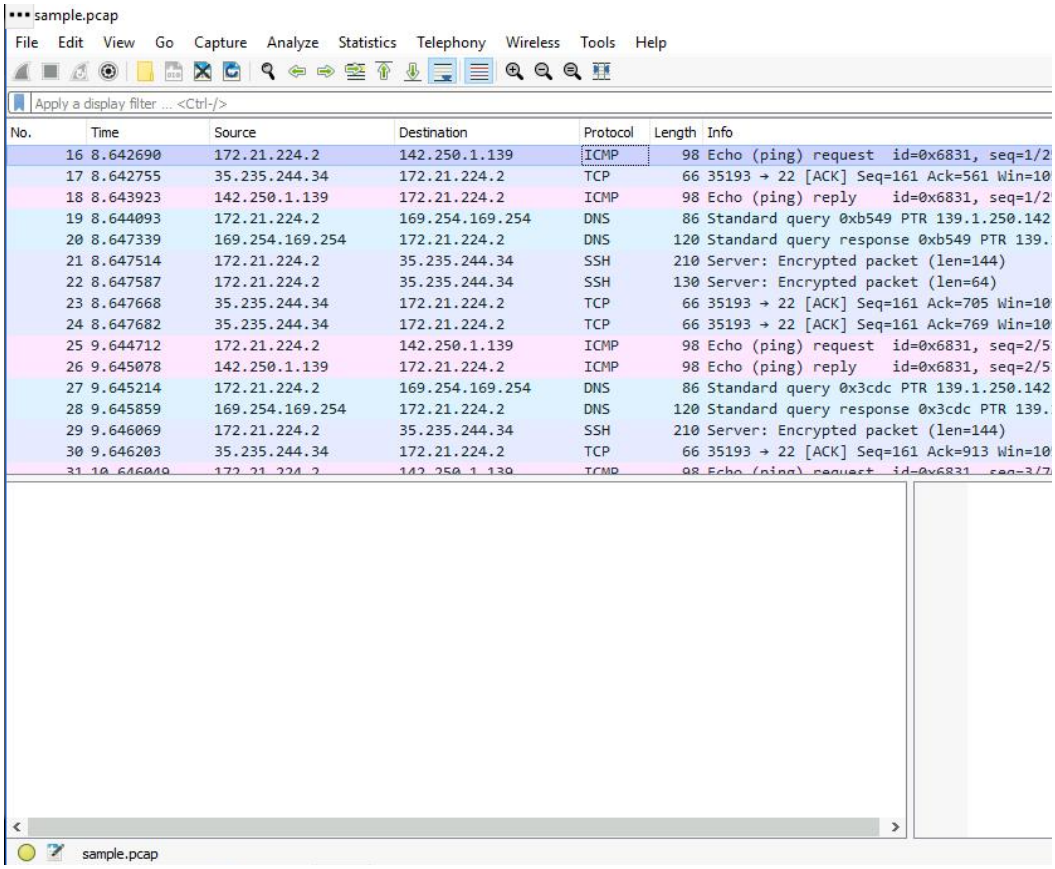
- identify the source and destination IP addresses involved in this web browsing session,
- examine the protocols that are used when the user makes the connection to the website, and
- analyze some of the data packets to identify the type of information sent and received by the systems that connect to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

Task 1. Explore data with Wireshark

1. Open the packet capture file by double-clicking the file called **sample** on the Windows desktop. Wireshark starts.
2. Double-click the Wireshark title bar next to the **sample.pcap** filename to maximize the Wireshark application window.
3. Scroll down the packet list until a packet is listed where the info column starts with the words 'Echo (ping) request'.

- **What is the protocol of the first packet in the list where the info column starts with the words 'Echo (ping) request'?**

  ICMP



Task 2. Apply a basic Wireshark filter and inspect a packet

In this task, you'll open a packet in Wireshark for more detailed exploration and filter the data to inspect the network layers and protocols contained in the packet.

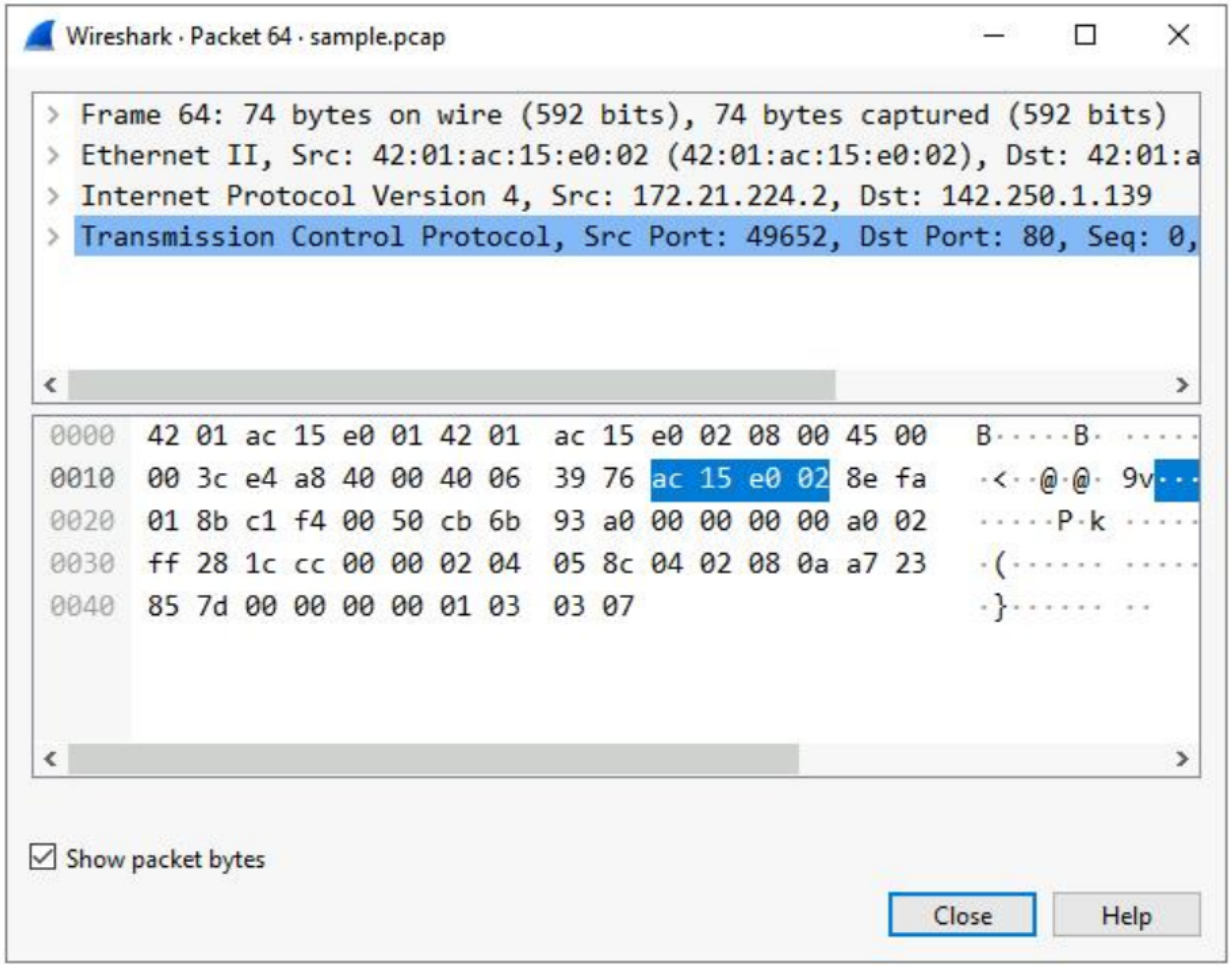
1. Enter the following filter for traffic associated with a specific IP address. Enter this into the **Apply a display filter...** text box immediately above the list of packets:

```
ip.addr == 142.250.1.139
```

2. Press **ENTER** or click the **Apply display filter** icon in the filter text box.

The list of packets displayed is now significantly reduced and contains only packets where either the source or the destination IP address matches the address you entered. Now only two packet colors are used: light pink for ICMP protocol packets and light green for TCP (and HTTP, which is a subset of TCP) packets.

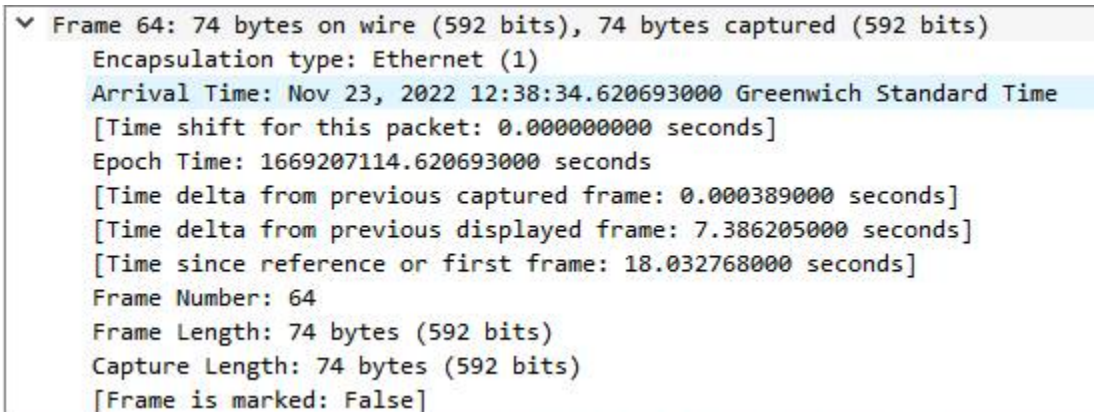
3. Double-click the first packet that lists **TCP** as the protocol.



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 (“.”).

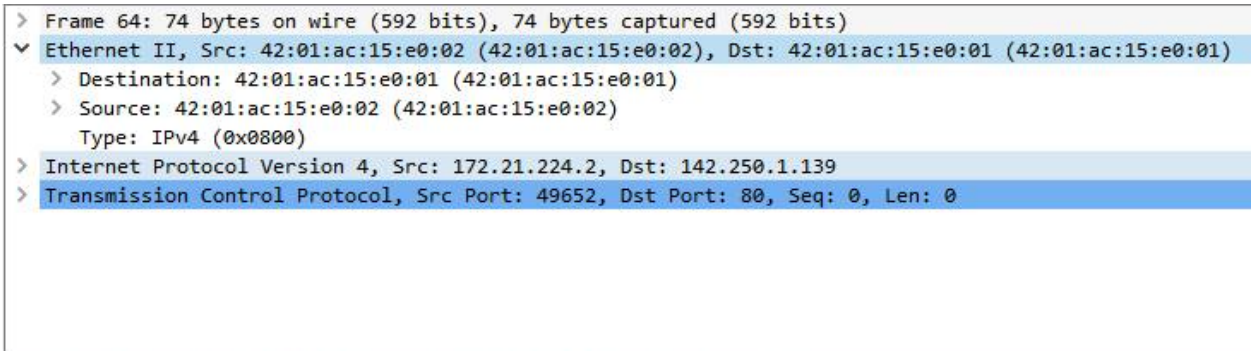
4. Double-click the first subtree in the upper section. This starts with the word **Frame**.

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.



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

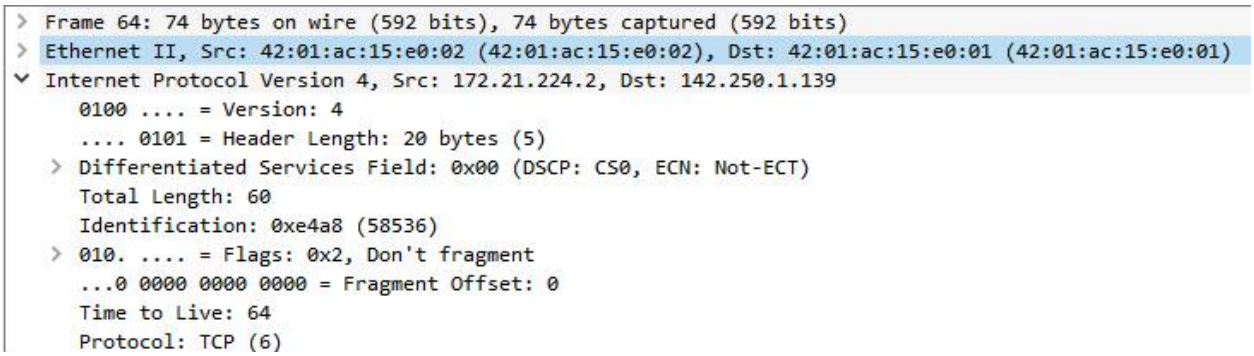
This item contains details about the packet at the Ethernet level, including the source and destination MAC addresses and the type of internal protocol that the Ethernet packet contains.



6. 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.

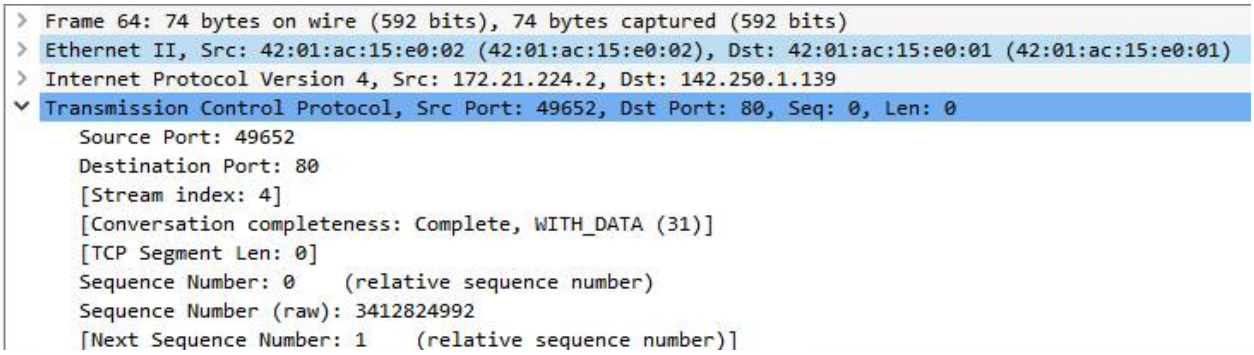


7. Double-click **Internet Protocol Version 4** again to collapse that subtree and then double-click the **Transmission Control Protocol** subtree.

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

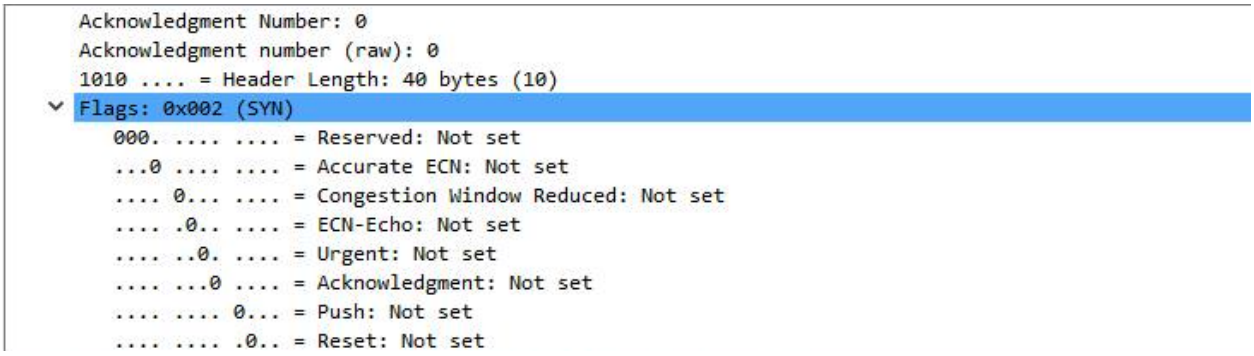
The source port and destination port listed here match the source and destination ports in the info column of the summary display for this packet in the list of all of the packets in the main Wireshark window.

- **What is the TCP destination port of this TCP packet?**



8. In the **Transmission Control Protocol** subtree, scroll down and double-click **Flags**.

This provides a detailed view of the TCP flags set in this packet.



9. Click the **X** icon to close the detailed packet inspection window.
10. Click the **X Clear display filter** icon in the Wireshark filter bar to clear the IP address filter.

All the packets have returned to the display.

Task 3. Use filters to select packets

In this task, you'll use filters to analyze specific network packets based on where the packets came from or where they were sent to. You'll explore how to select packets using either their physical Ethernet Media Access Control (MAC) address or their Internet Protocol (IP) address.

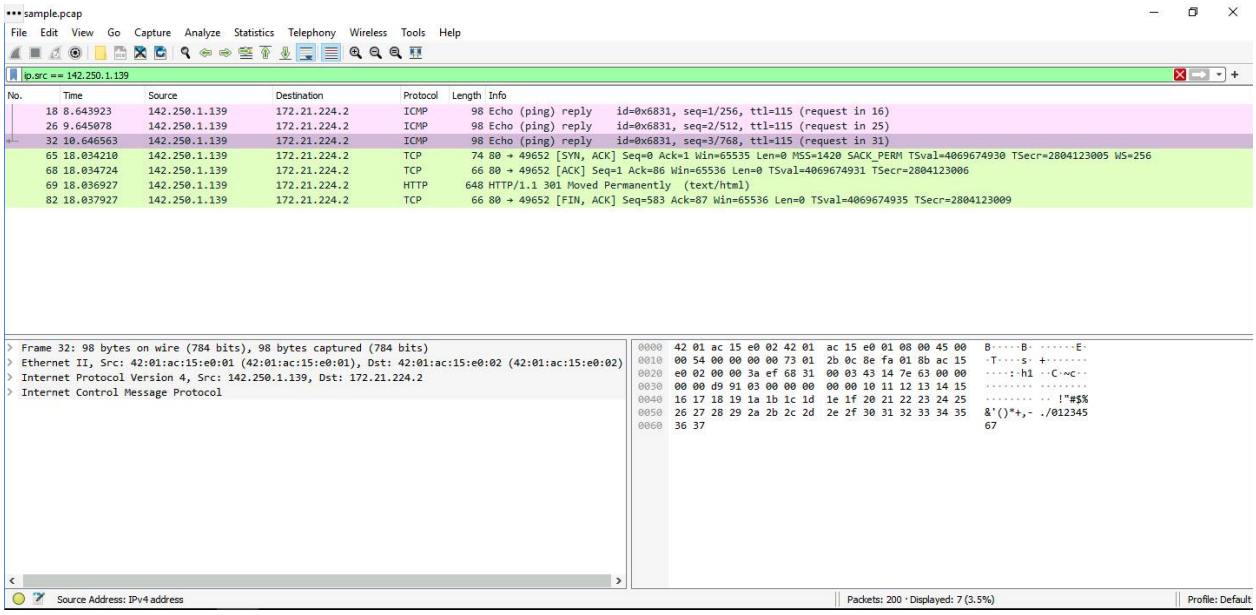
1. Enter the following filter to select traffic for a specific source IP address only.

   Enter this into the **Apply a display filter...** text box immediately above the list of packets:

```
ip.src == 142.250.1.139
```

2. Press **ENTER** or click the **Apply display filter** icon in the filter text box.

A filtered list is returned with fewer entries than before. It contains only packets that came from 142.250.1.139.

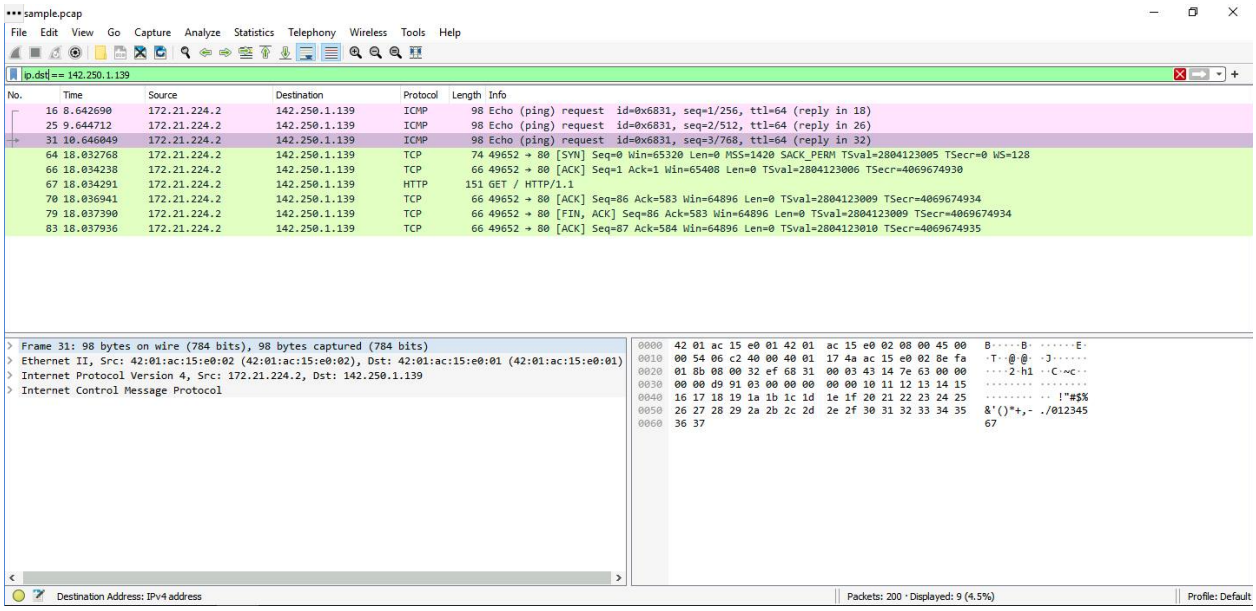


3. Click the **X Clear display filter** icon in the Wireshark filter bar to clear the IP address filter.
4. Enter the following filter to select traffic for a specific destination IP address only:

ip.dst == 142.250.1.139

5. Press **ENTER** or click the **Apply display filter** icon in the filter text box.

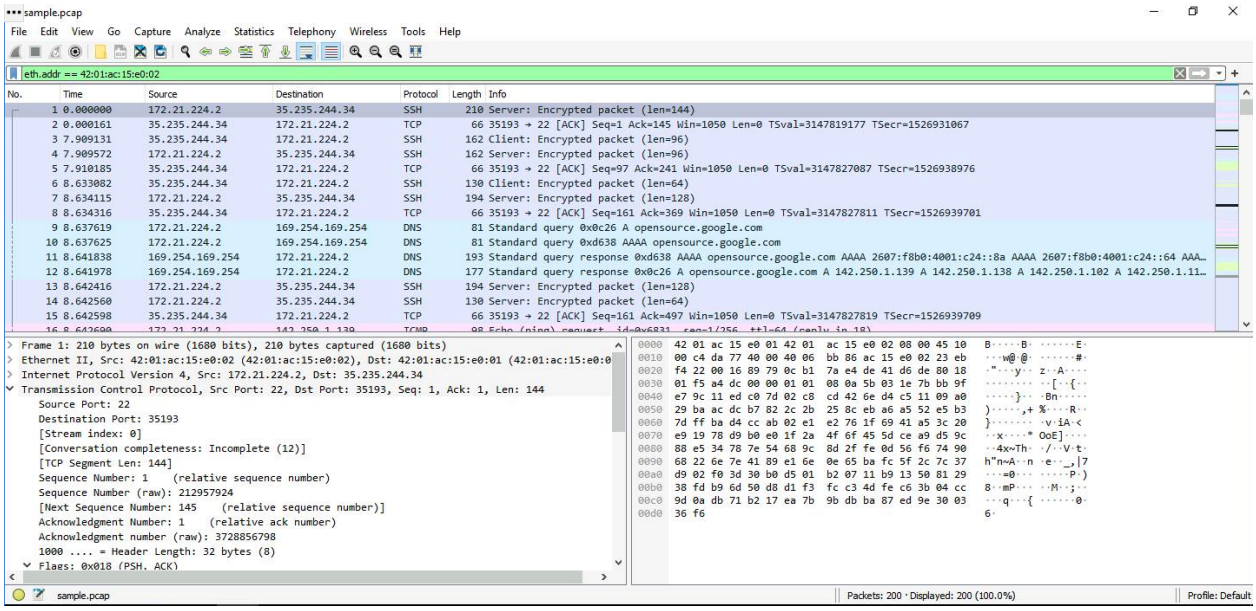
A filtered list is returned that contains only packets that were sent to 142.250.1.139.



6. Click the **X Clear display filter** icon in the Wireshark filter bar to clear the IP address filter.
7. Enter the following 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:

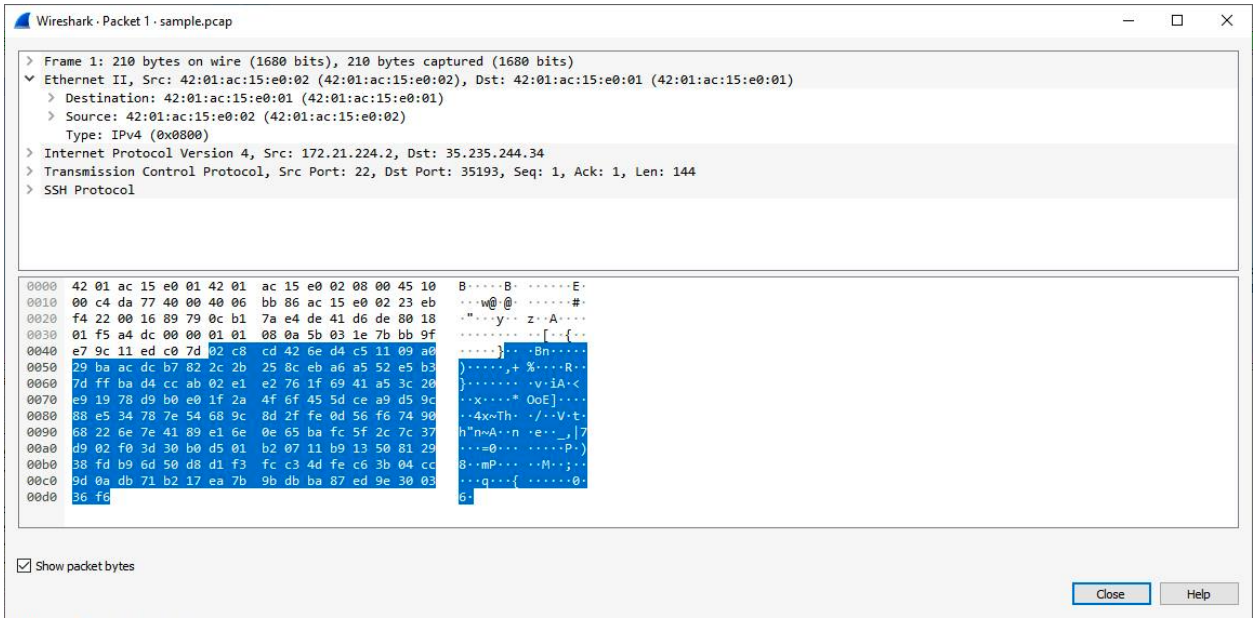
```
eth.addr == 42:01:ac:15:e0:02
```

8. Press **ENTER** or click the **Apply display filter** icon in the filter text box.



9. Double-click the first packet in the list. You may need to scroll back to display the first packet in the filtered list.
10. 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.

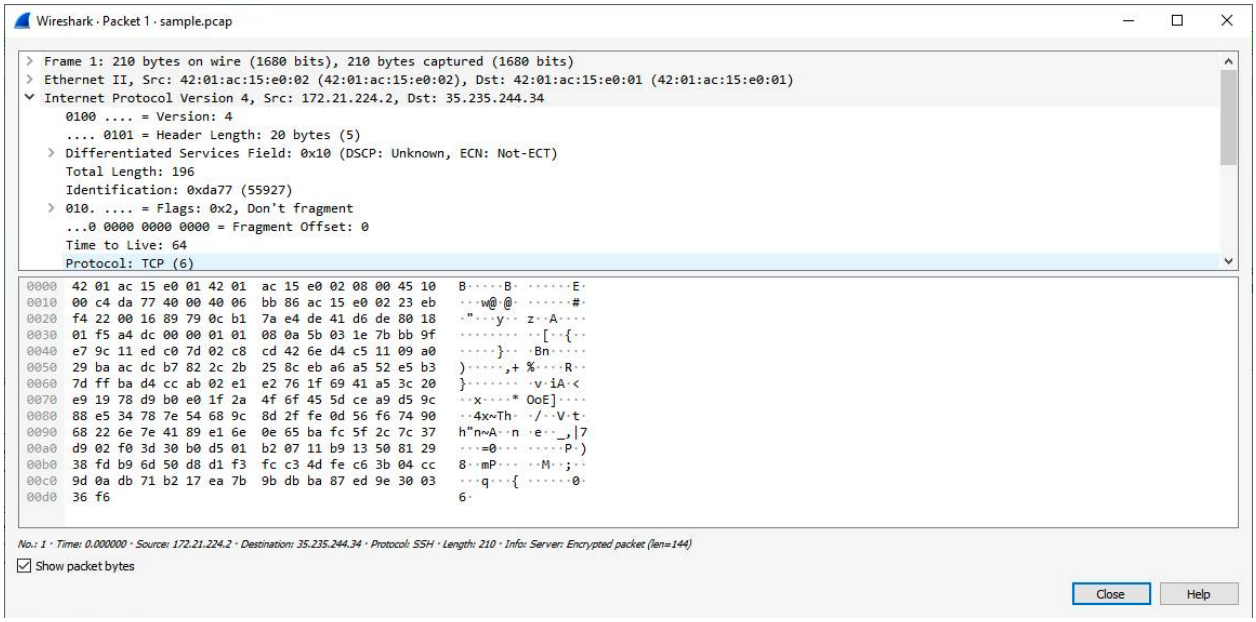


11. Double-click the **Ethernet II** subtree to close it.
12. Double-click the **Internet Protocol Version 4** subtree to expand it and scroll down until the **Time to Live** and **Protocol** fields appear.

The **Protocol** field in the **Internet Protocol Version 4** subtree indicates which IP internal protocol is contained in the packet.

- **What is the protocol contained in the Internet Protocol Version 4 subtree from the first packet related to MAC address 42:01:ac:15:e0:02?**

  TCP



12. Click the **X** icon to close the detailed packet inspection window.
13. Click the **X Clear display filter** icon in the Wireshark filter bar to clear the MAC address filter.

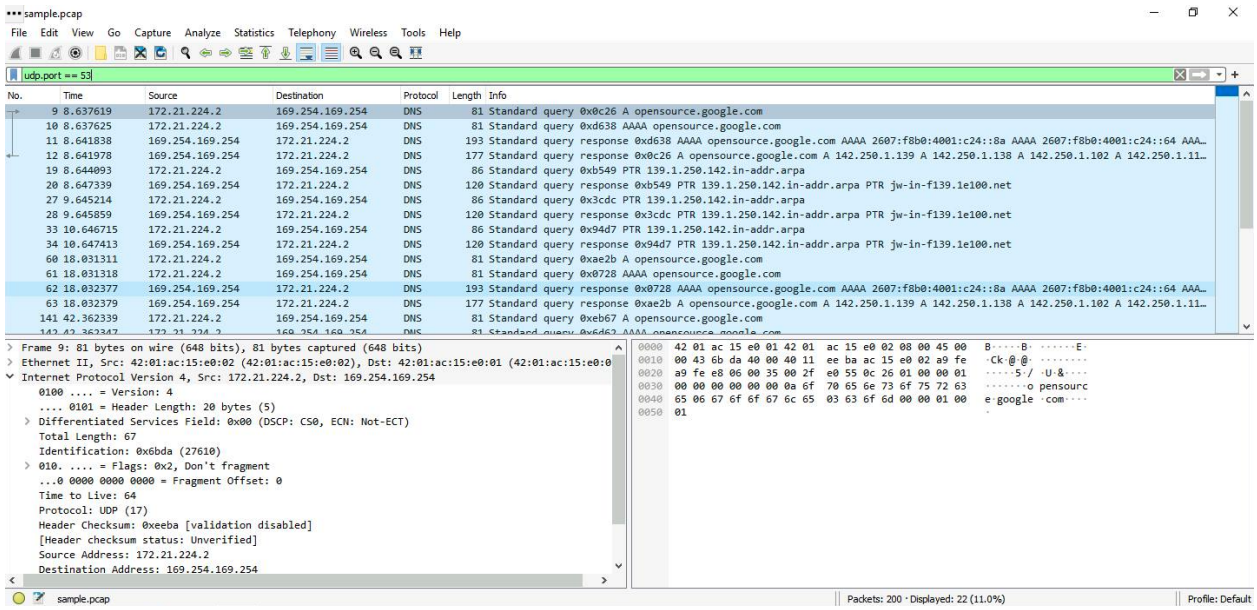
Task 4. Use filters to explore DNS packets

In this task, you'll use filters to select and examine DNS traffic. Once you've selected sample DNS traffic, you'll drill down into the protocol to examine how the DNS packet data contains both queries (names of internet sites that are being looked up) and answers (IP addresses that are being sent back by a DNS server when a name is successfully resolved).

1. Enter the following 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. Enter this into the **Apply a display filter...** text box immediately above the list of packets:

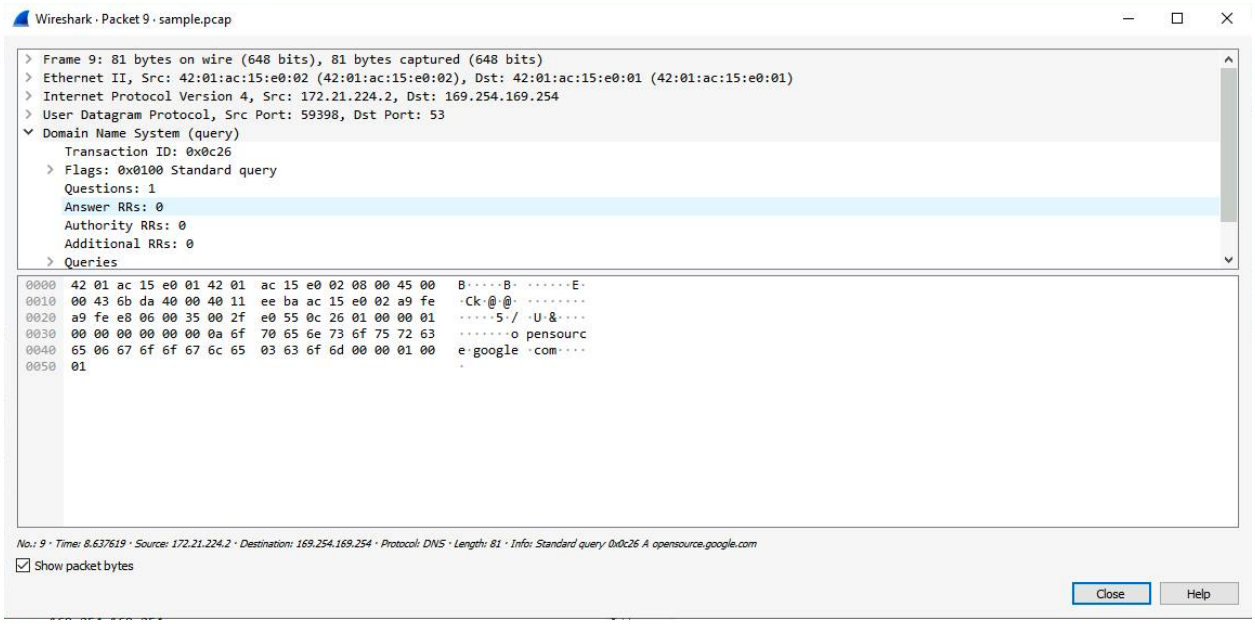
udp.port == 53

2. Press **ENTER** or click the **Apply display filter** icon in the filter text box.

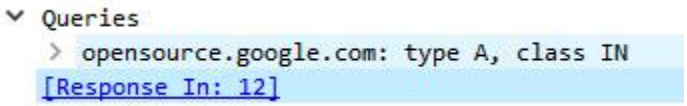


3. Double-click the first packet in the list to open the detailed packet window.

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



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



You'll notice that the name of the website that was queried is **opensource.google.com**.

6. Click the **X** icon to close the detailed packet inspection window.
7. Double-click the fourth packet in the list to open the detailed packet window.
8. Scroll down and double-click the **Domain Name System (query)** subtree to expand it.
9. Scroll down and double-click **Answers**, which is in the **Domain Name System (query)** subtree.

The Answers data includes the name that was queried (**opensource.google.com**) and the addresses that are associated with that name.

- **What is the protocol contained in the Internet Protocol Version 4 subtree from the first packet related to MAC address 42:01:ac:15:e0:02?**

  142.250.1.139

10. Click the **X** icon to close the detailed packet inspection window.
11. Click the **X Clear display filter** icon in the Wireshark filter bar to clear the filter.

Task 5. Use filters to explore TCP packets

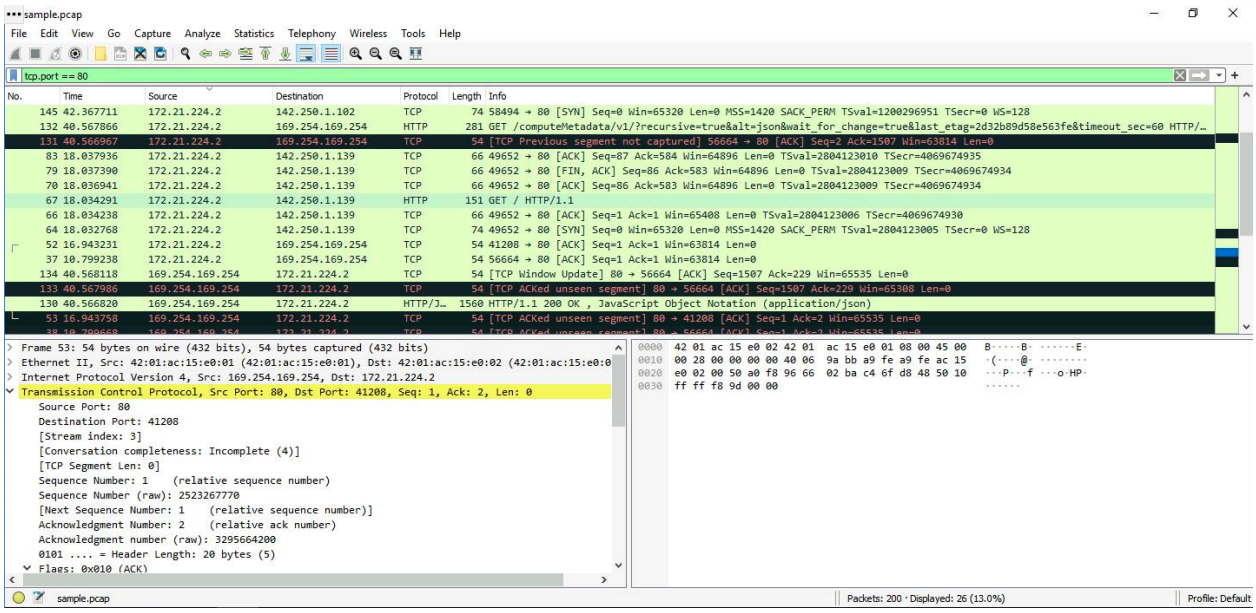
In this task, you'll use additional filters to select and examine TCP packets. You'll learn how to search for text that is present in payload data contained inside network packets. This will locate packets based on something such as a name or some other text that is of interest to you.

1. Enter the following filter to select TCP port 80 traffic. TCP port 80 is the default port that is associated with web traffic:

tcp.port == 80

2. Press **ENTER** or click the **Apply display filter** icon in the filter text box.

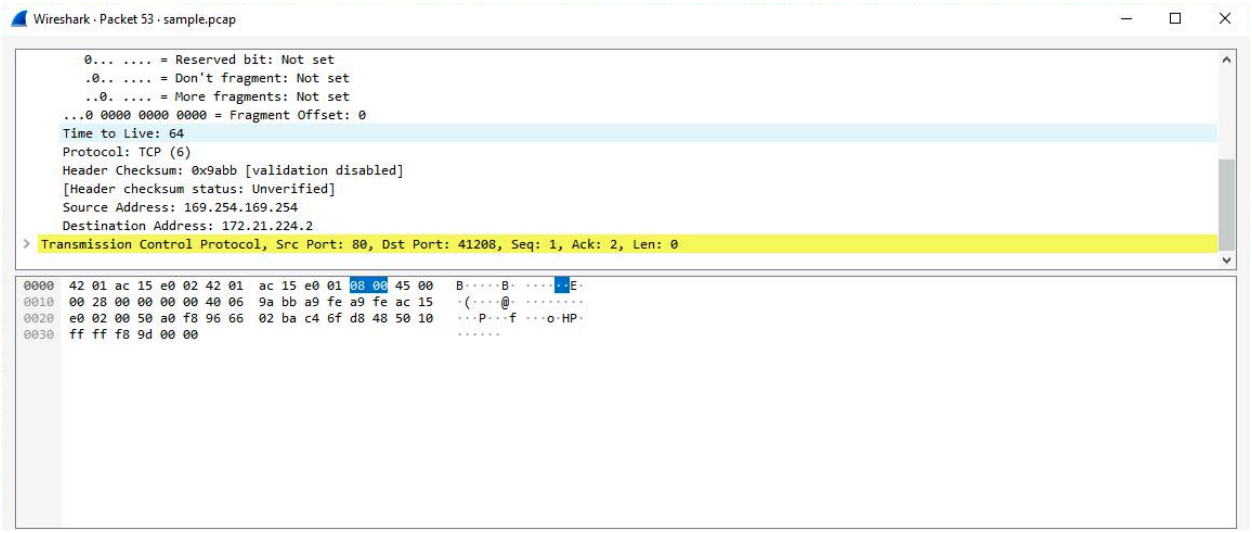
Quite a few packets were created when the user accessed the web page http://opensource.google.com.



3. Double-click the first packet in the list. The **Destination** IP address of this packet is 169.254.169.254.

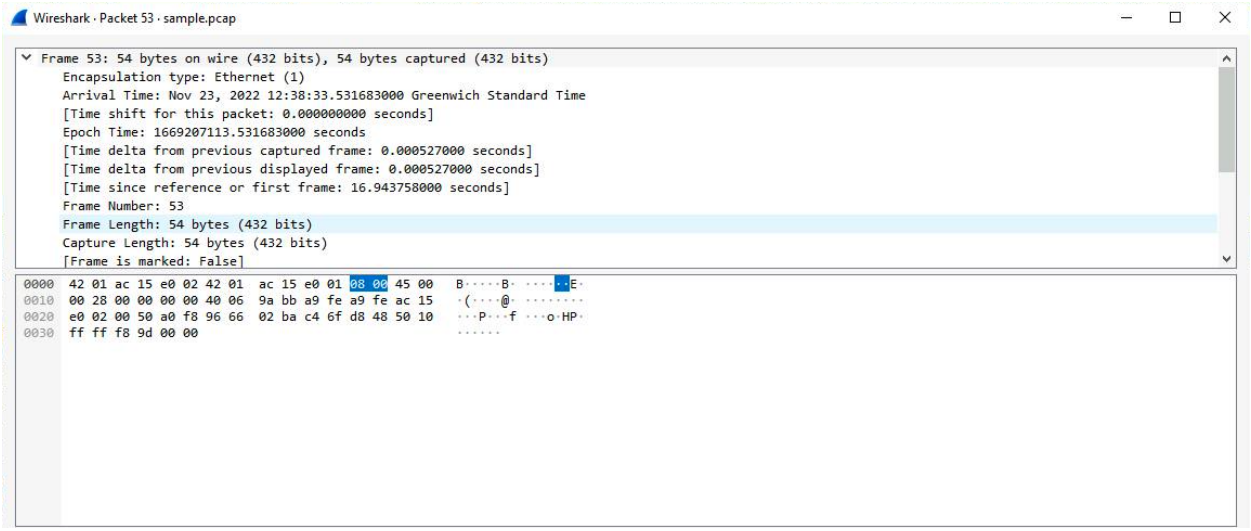
- **What is the Time to Live value of the packet as specified in the Internet Protocol Version 4 subtree?**

  64



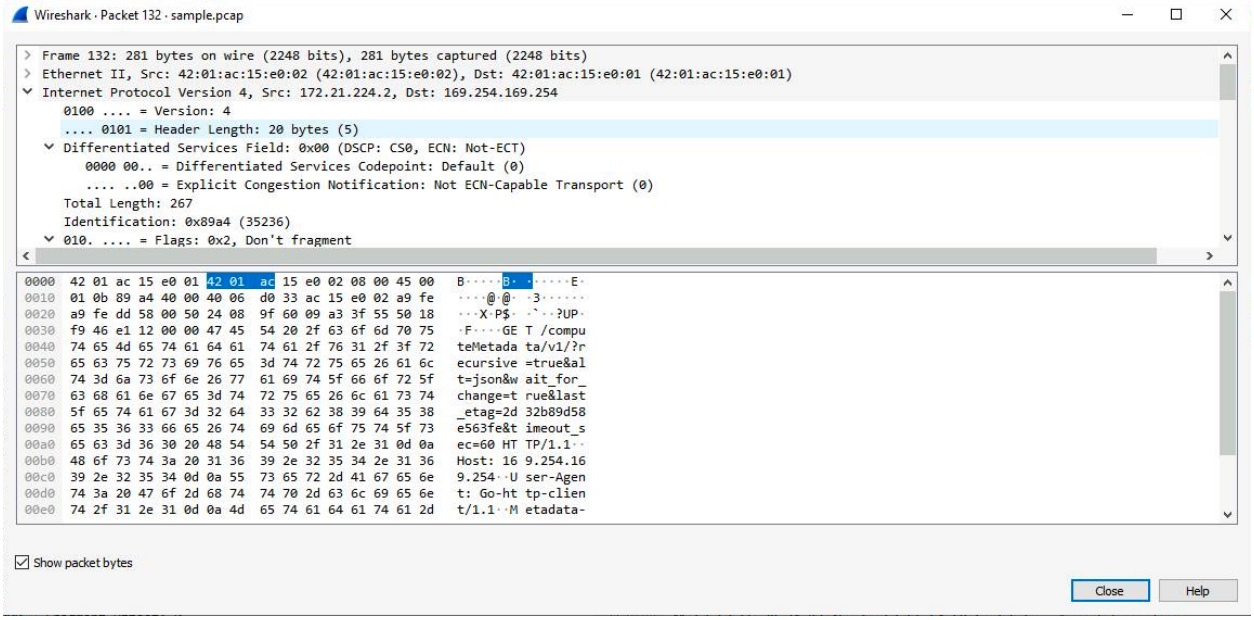
- **What is the Frame Length of the packet as specified in the Frame subtree?**

  54 bytes



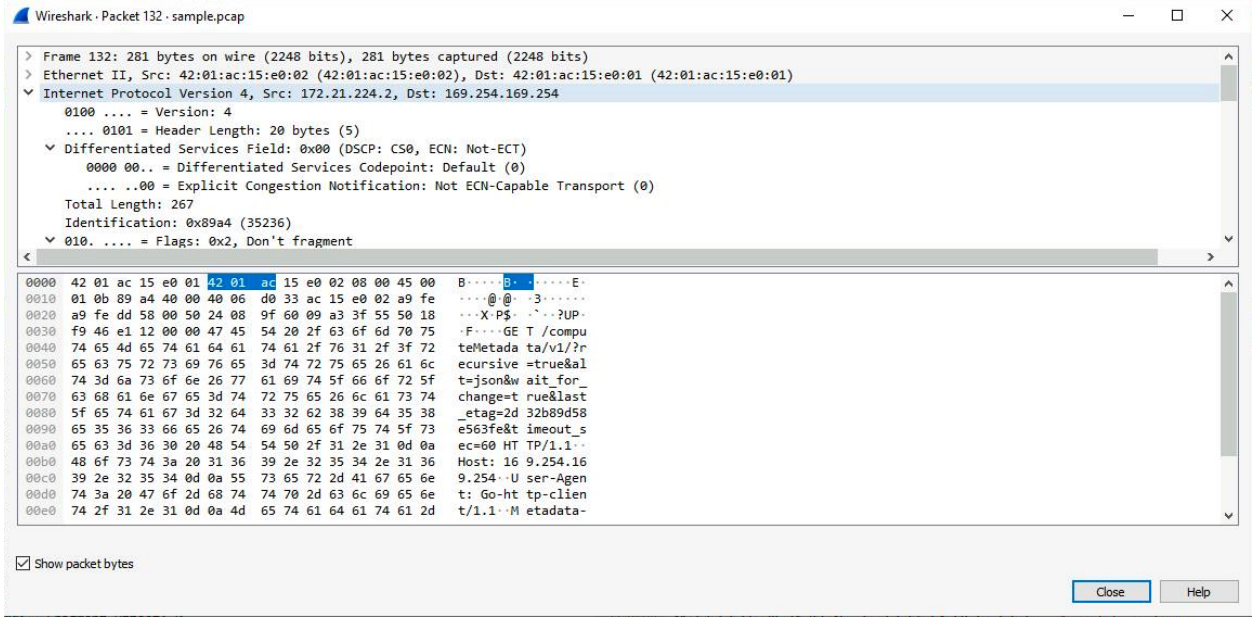
- **What is the Header Length of the packet as specified in the Internet Protocol Version 4 subtree?**

20 bytes



- **What is the Destination Address as specified in the Internet Protocol Version 4 subtree?**

169.254.169.254

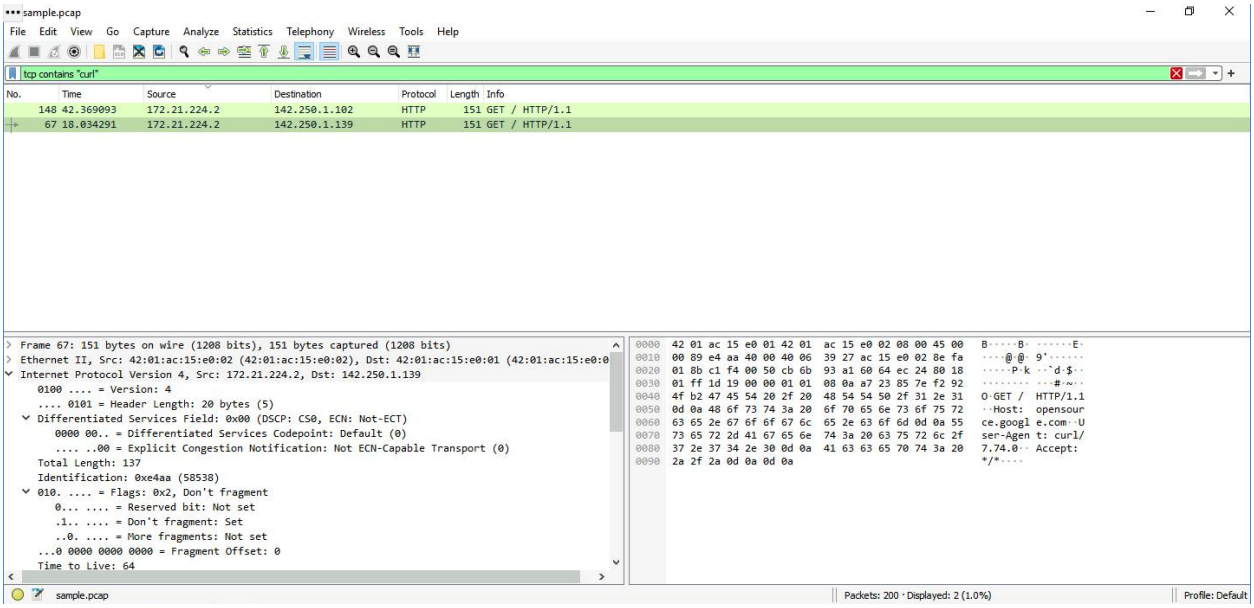


4. Click the **X** icon to close the detailed packet inspection window.
5. Click the **X Clear display filter** icon in the Wireshark filter bar to clear the filter.
6. Enter the following filter to select TCP packet data that contains specific text data.

tcp contains "curl"

7. Press **ENTER** or click the **Apply display filter** icon in the filter text box.

This filters to packets containing web requests made with the curl command in this sample packet capture file.


sample.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
tcp contains "curl"
No. Time Source Destination Protocol Length Info
148 42.369093 172.21.224.2 142.250.1.102 HTTP 151 GET / HTTP/1.1
67 18.034291 172.21.224.2 142.250.1.139 HTTP 151 GET / HTTP/1.1
Frame 67: 151 bytes on wire (1208 bits), 151 bytes captured (1208 bits)
Ethernet II, Src: 42:01:ac:15:e0:02 (42:01:ac:15:e0:02), Dst: 42:01:ac:15:e0:01 (42:01:ac:15:e0:0
Internet Protocol Version 4, Src: 172.21.224.2, Dst: 142.250.1.139
0100 .... = Version: 4
.... 0101 = Header Length: 20 bytes (5)
Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
0000 00.. = Differentiated Services Codepoint: Default (0)
.... ..00 = Explicit Congestion Notification: Not ECN-Capable Transport (0)
Total Length: 137
Identification: 0xe4aa (58538)
010. .... = Flags: 0x2, Don't fragment
0... .... = Reserved bit: Not set
.1.. .... = Don't fragment: Set
..0. .... = More fragments: Not set
...0 0000 0000 0000 = Fragment Offset: 0
Time to Live: 64
0000 42 01 ac 15 e0 01 42 01 ac 15 e0 02 08 00 45 00
0010 00 89 e4 aa 40 00 40 06 39 27 ac 15 e0 02 8e fa
0020 01 8b c1 f4 00 50 cb 6b 93 a1 60 64 ec 24 80 18
0030 01 ff 1d 19 00 00 01 01 08 0a a7 23 85 7e f2 92
0040 4f b2 47 45 54 20 2f 20 48 54 54 50 2f 31 2e 31
0050 0d 0a 48 6f 73 74 3a 20 6f 70 65 6e 73 6f 75 72
0060 63 65 2e 67 6f 6f 67 6c 65 2e 63 6f 6d 0d 0a 55
0070 73 65 72 2d 41 67 65 6e 74 3a 20 63 75 72 6c 2f
0080 37 2e 37 34 2e 30 0d 0a 41 63 63 65 70 74 3a 20
0090 2a 2f 2a 0d 0a 0d 0a
sample.pcap
Packets: 200 · Displayed: 2 (1.0%)
Profile: Default