

## 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:

1. Identify the source and destination IP addresses involved in this web browsing session.
2. Examine the protocols that are used when the user makes the connection to the website.
3. Analyze the data packet to identify the type of information sent and received by the systems that connect to each other when the network data is captured.

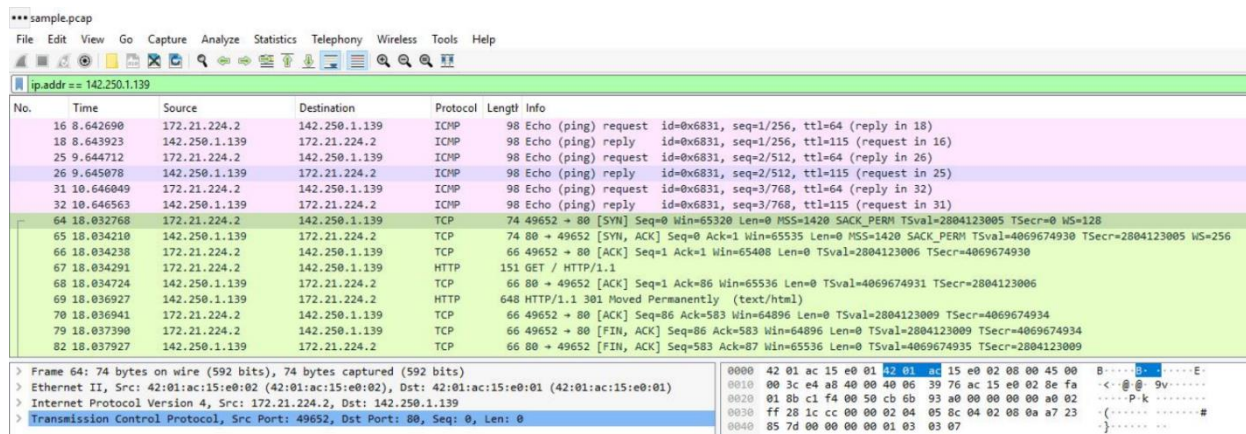
An overview of the key property columns listed for each packet:

- 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.

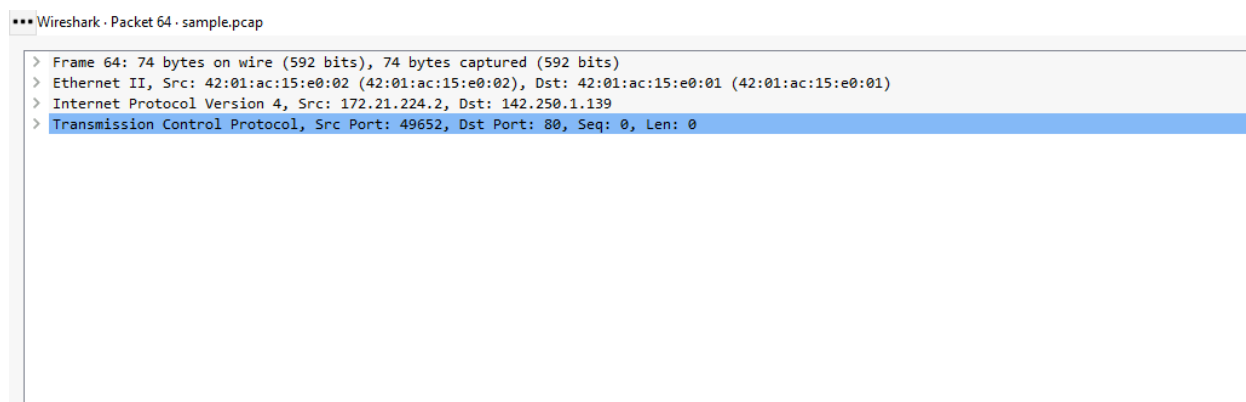
## Solutions

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

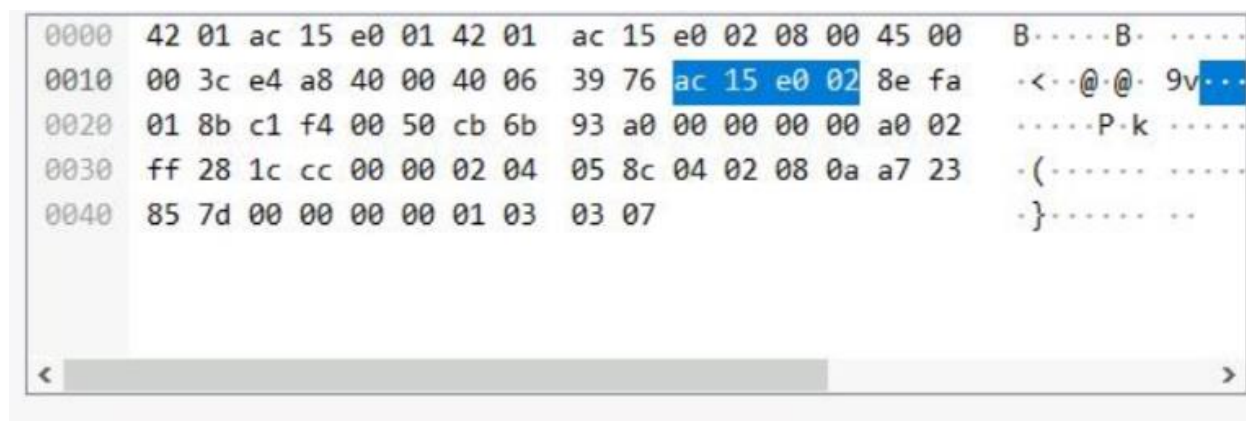
In this task, you'll open a packet in Wireshark for more detailed exploration and filter the data to inspect the network layers and protocols contained in the packet.



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.



The upper section of this window contains subtrees where Wireshark will provide you with an analysis of the various parts of the network packet.



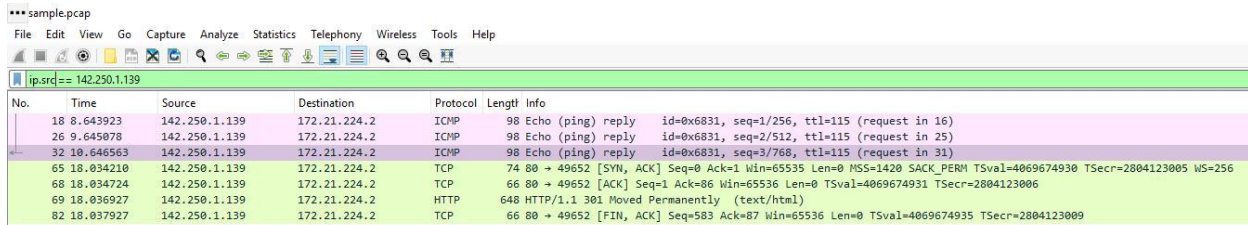
The lower section of the window contains the raw packet data displayed in hexadecimal and ASCII text. There is also placeholder text for fields where the character data does not apply, as indicated by the dot ("").

## Task 2. 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.

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

`ip.src == 142.250.1.139`

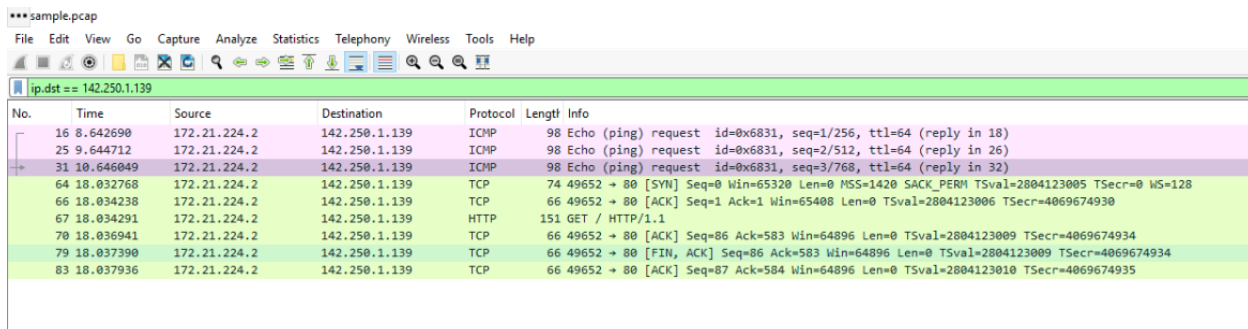


No.	Time	Source	Destination	Protocol	Length	Info
18	8.643923	142.250.1.139	172.21.224.2	ICMP	98	Echo (ping) reply id=0x6831, seq=1/256, ttl=115 (request in 16)
26	9.645078	142.250.1.139	172.21.224.2	ICMP	98	Echo (ping) reply id=0x6831, seq=2/512, ttl=115 (request in 25)
32	10.646563	142.250.1.139	172.21.224.2	ICMP	98	Echo (ping) reply id=0x6831, seq=3/768, ttl=115 (request in 31)
65	18.034210	142.250.1.139	172.21.224.2	TCP	74	80 → 49652 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0 MSS=1420 SACK_PERM TSval=4069674930 TSecr=2804123005 WS=256
68	18.034724	142.250.1.139	172.21.224.2	TCP	66	80 → 49652 [ACK] Seq=1 Ack=86 Win=65536 Len=0 TSval=4069674931 TSecr=2804123006
69	18.036927	142.250.1.139	172.21.224.2	HTTP	648	HTTP/1.1 301 Moved Permanently (text/html)
82	18.037927	142.250.1.139	172.21.224.2	TCP	66	80 → 49652 [FIN, ACK] Seq=583 Ack=87 Win=65536 Len=0 TSval=4069674935 TSecr=2804123009

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

Enter the following filter to select traffic for a specific destination IP address only:

`ip.dst == 142.250.1.139`

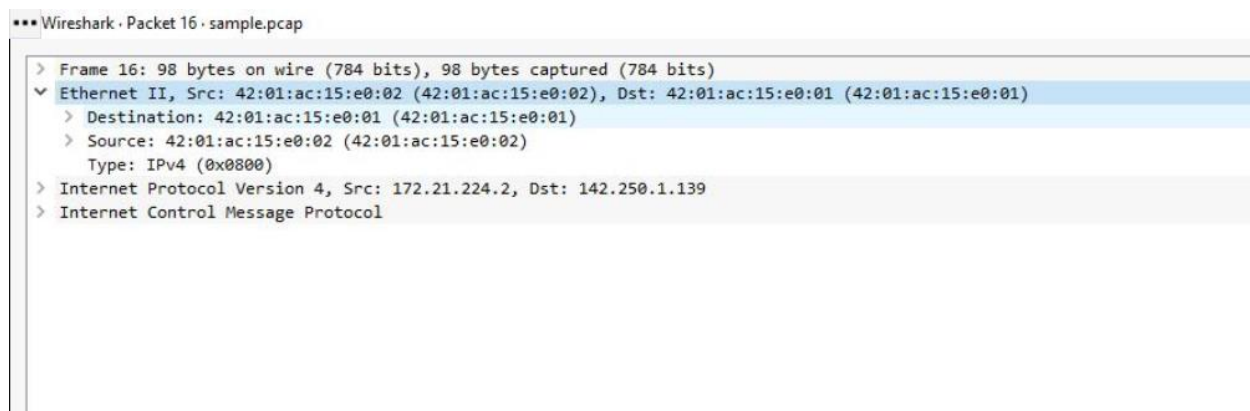


No.	Time	Source	Destination	Protocol	Length	Info
16	8.642690	172.21.224.2	142.250.1.139	ICMP	98	Echo (ping) request id=0x6831, seq=1/256, ttl=64 (reply in 18)
25	9.644712	172.21.224.2	142.250.1.139	ICMP	98	Echo (ping) request id=0x6831, seq=2/512, ttl=64 (reply in 26)
31	10.646049	172.21.224.2	142.250.1.139	ICMP	98	Echo (ping) request id=0x6831, seq=3/768, ttl=64 (reply in 32)
64	18.032768	172.21.224.2	142.250.1.139	TCP	74	49652 → 80 [SYN] Seq=0 Win=65320 Len=0 MSS=1420 SACK_PERM TSval=2804123005 TSecr=0 WS=128
66	18.034238	172.21.224.2	142.250.1.139	TCP	66	49652 → 80 [ACK] Seq=1 Ack=1 Win=65408 Len=0 TSval=2804123006 TSecr=4069674930
67	18.034291	172.21.224.2	142.250.1.139	HTTP	151	GET / HTTP/1.1
70	18.036941	172.21.224.2	142.250.1.139	TCP	66	49652 → 80 [ACK] Seq=86 Ack=583 Win=64896 Len=0 TSval=2804123009 TSecr=4069674934
79	18.037390	172.21.224.2	142.250.1.139	TCP	66	49652 → 80 [FIN, ACK] Seq=86 Ack=583 Win=64896 Len=0 TSval=2804123009 TSecr=4069674934
83	18.037936	172.21.224.2	142.250.1.139	TCP	66	49652 → 80 [ACK] Seq=87 Ack=584 Win=64896 Len=0 TSval=2804123010 TSecr=4069674935

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

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`



```

> Frame 16: 98 bytes on wire (784 bits), 98 bytes captured (784 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:e0:01)
> 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)
    Total Length: 84
    Identification: 0x0622 (1570)
  > 010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: ICMP (1)
    Header Checksum: 0x17ea [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 172.21.224.2
    Destination Address: 142.250.1.139
> Internet Control Message Protocol

```

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).

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`

\*\*\*Wireshark · Packet 9 · sample.pcap

```
> Frame 9: 81 bytes on wire (648 bits), 81 bytes captured (648 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:e0:01)
> Internet Protocol Version 4, Src: 172.21.224.2, Dst: 169.254.169.254
> User Datagram Protocol, Src Port: 59398, Dst Port: 53
▼ Domain Name System (query)
  Transaction ID: 0x0c26
  > Flags: 0x0100 Standard query
  Questions: 1
  Answer RRs: 0
  Authority RRs: 0
  Additional RRs: 0
  ▼ Queries
    > opensource.google.com: type A, class IN
    [Response In: 12]
```

\*\*\*Wireshark · Packet 12 · sample.pcap

```
> Frame 12: 177 bytes on wire (1416 bits), 177 bytes captured (1416 bits)
> Ethernet II, Src: 42:01:ac:15:e0:01 (42:01:ac:15:e0:01), Dst: 42:01:ac:15:e0:02 (42:01:ac:15:e0:02)
> Internet Protocol Version 4, Src: 169.254.169.254, Dst: 172.21.224.2
> User Datagram Protocol, Src Port: 53, Dst Port: 59398
▼ Domain Name System (response)
  Transaction ID: 0x0c26
  > Flags: 0x8180 Standard query response, No error
  Questions: 1
  Answer RRs: 6
  Authority RRs: 0
  Additional RRs: 0
  ▼ Queries
    > opensource.google.com: type A, class IN
  ▼ Answers
    > opensource.google.com: type A, class IN, addr 142.250.1.139
    > opensource.google.com: type A, class IN, addr 142.250.1.138
    > opensource.google.com: type A, class IN, addr 142.250.1.102
    > opensource.google.com: type A, class IN, addr 142.250.1.113
    > opensource.google.com: type A, class IN, addr 142.250.1.100
    > opensource.google.com: type A, class IN, addr 142.250.1.101
    [Request In: 9]
  [Time: 0.004359000 seconds]
```

The IP address 142.250.1.139 is displayed in the expanded Answers section for the DNS query for **opensource.google.com**.



## Task 4. 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.

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

\*\*\*sample.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port == 80

No.	Time	Source	Destination	Protocol	Length	Info
79	18.037390	172.21.224.2	142.250.1.139	TCP	66	49652 → 80 [FIN, ACK] Seq=86 Ack=583 Win=64896 Len=0 TSval=2804123009 TSecr=4069674934
82	18.037927	142.250.1.139	172.21.224.2	TCP	66	80 → 49652 [FIN, ACK] Seq=583 Ack=87 Win=65536 Len=0 TSval=4069674935 TSecr=2804123009
83	18.037936	172.21.224.2	142.250.1.139	TCP	66	49652 → 80 [ACK] Seq=87 Ack=584 Win=64896 Len=0 TSval=2804123010 TSecr=4069674935
130	40.566820	169.254.169.254	172.21.224.2	HTTP/1.1	1560	HTTP/1.1 200 OK, JSON (application/json)
131	40.566967	172.21.224.2	169.254.169.254	TCP	54	[TCP Previous segment not captured] 56664 → 80 [ACK] Seq=2 Ack=1507 Win=63814 Len=0
132	40.567866	172.21.224.2	169.254.169.254	HTTP	281	GET /compute/metadata/v1/?recursive=true&alt=json&wait_for_change=true&last_etag=2d32b89d58e563fe&timeout_sec=60 HTTP/1.1
133	40.567986	169.254.169.254	172.21.224.2	TCP	54	80 → 56664 [ACK] Seq=1507 Ack=229 Win=65536 Len=0
134	40.568118	169.254.169.254	172.21.224.2	TCP	54	[TCP Window Update] 80 → 56664 [ACK] Seq=1507 Ack=229 Win=65536 Len=0
145	42.367711	172.21.224.2	142.250.1.102	TCP	74	58494 → 80 [SYN] Seq=0 Win=65536 Len=0 MSS=1420 SACK_PERM TSval=1200296951 TSecr=0 WS=128
146	42.368889	142.250.1.102	172.21.224.2	TCP	74	80 → 58494 [SYN, ACK] Seq=0 Ack=1 Win=65536 Len=0 MSS=1420 SACK_PERM TSval=1996174824 TSecr=1200296951 WS=256
147	42.369019	172.21.224.2	142.250.1.102	TCP	66	58494 → 80 [ACK] Seq=1 Ack=1 Win=65408 Len=0 TSval=1200296953 TSecr=1996174824
148	42.369093	172.21.224.2	142.250.1.102	HTTP	151	GET / HTTP/1.1
149	42.369285	142.250.1.102	172.21.224.2	TCP	66	80 → 58494 [ACK] Seq=1 Ack=86 Win=65536 Len=0 TSval=1996174825 TSecr=1200296953
150	42.370267	142.250.1.102	172.21.224.2	HTTP	657	HTTP/1.1 301 Moved Permanently (text/html)
151	42.370278	172.21.224.2	142.250.1.102	TCP	66	58494 → 80 [ACK] Seq=86 Ack=592 Win=64896 Len=0 TSval=1200296954 TSecr=1996174826

> Frame 37: 54 bytes on wire (432 bits), 54 bytes captured (432 bits) on interface 0  
> 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:e0:01)  
> Internet Protocol Version 4, Src: 172.21.224.2, Dst: 169.254.169.254  
> Transmission Control Protocol, Src Port: 56664, Dst Port: 80, Seq: 1, Ack: 1, Len: 0

0000 42 01 ac 15 e0 01 42 01 ac 15 e0 02 00 00 45 00 B.....E.  
0010 00 89 e4 aa 40 00 40 06 d1 18 ac 15 e0 02 a9 fe .(. @ 9'.....  
0020 a9 fe dd 58 00 50 24 08 9f 5f 09 a3 39 73 50 10 ....X P\$ .....9sP.  
0030 f9 46 e0 2f 00 00 ..F./..

Enter the following filter to select TCP packet data that contains specific text data.

tcp contains "curl"

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp contains "curl"

No.	Time	Source	Destination	Protocol	Length	Info
67	18.034291	172.21.224.2	142.250.1.139	HTTP	151	GET / HTTP/1.1
148	42.369093	172.21.224.2	142.250.1.102	HTTP	151	GET / HTTP/1.1

> Frame 67: 151 bytes on wire (1208 bits), 151 bytes captured (1208 bits) on interface 0  
> 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:e0:01)  
> Internet Protocol Version 4, Src: 172.21.224.2, Dst: 142.250.1.139  
> Transmission Control Protocol, Src Port: 49652, Dst Port: 80, Seq: 1, Ack: 1, Len: 85  
> Hypertext Transfer Protocol

0000 42 01 ac 15 e0 01 42 01 ac 15 e0 02 00 00 45 00 B.....E.  
0010 00 89 e4 aa 40 00 40 06 39 27 ac 15 e0 02 8e fa .....@ 9'.....  
0020 01 8b c1 f4 00 50 cb 0b 93 a1 60 64 ec 24 00 18 ....P.k...d \$..  
0030 01 ff 1d 19 00 00 01 01 00 0a 07 23 05 7e f2 92 .....#.....  
0040 4f b2 47 45 54 20 2f 20 48 54 50 2f 31 2e 31 O.GET / HTTP/1.1  
0050 0d 0a 48 6f 73 74 3a 20 6f 70 65 6e 73 6f 75 72 ..Host: opensour  
0060 63 65 2e 67 6f 6f 67 6c 65 2e 63 6f 6d 0d 0a 55 ce.googl e.com..U  
0070 73 65 72 2d 41 67 65 6e 74 3a 20 63 75 72 6c 2f ser-Agen t: curl/  
0080 37 2e 37 34 2e 30 0d 0a 41 63 63 65 70 74 3a 20 7.74.0... Accept:  
0090 2a 2f 2a 0d 0a 0d 0a \*/\*.....