Wireshark Packet Analysis

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, and
3. 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.

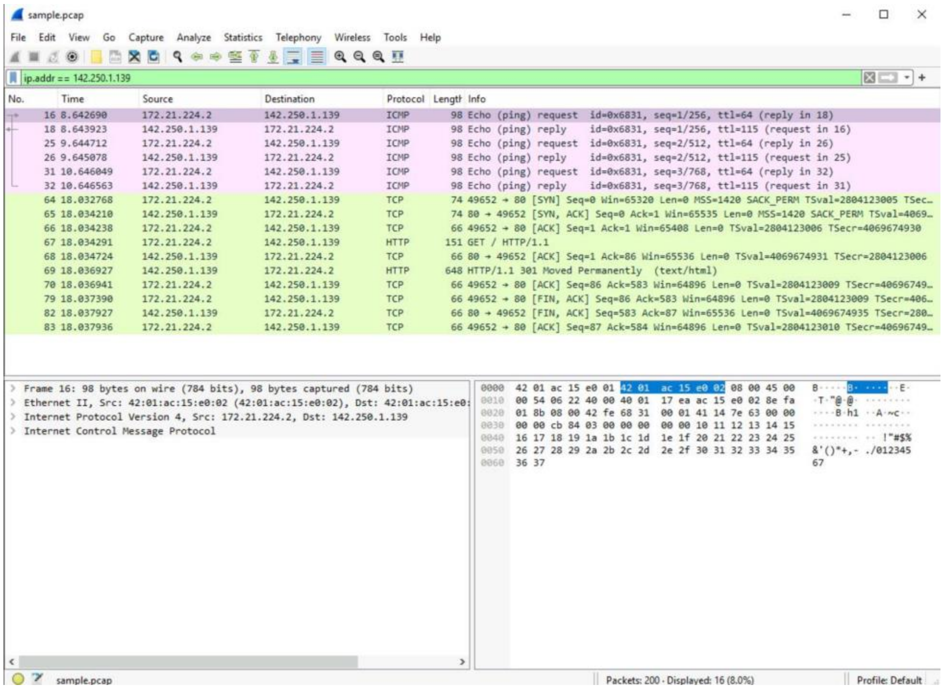
An overview of the key property columns listed for each packet:

- **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 infomation about the data in the packet (the payload) as interpreted by Wireshark.

Solutions

1. Identify the source and destination IP addresses involved in this web browsing session.

On the title bar, type ip.addr == 142.250.1.139 to filter for traffic associated with a specific IP address. Select the first packet that contains TCP on the info field. addr means either the source or the destination IP.



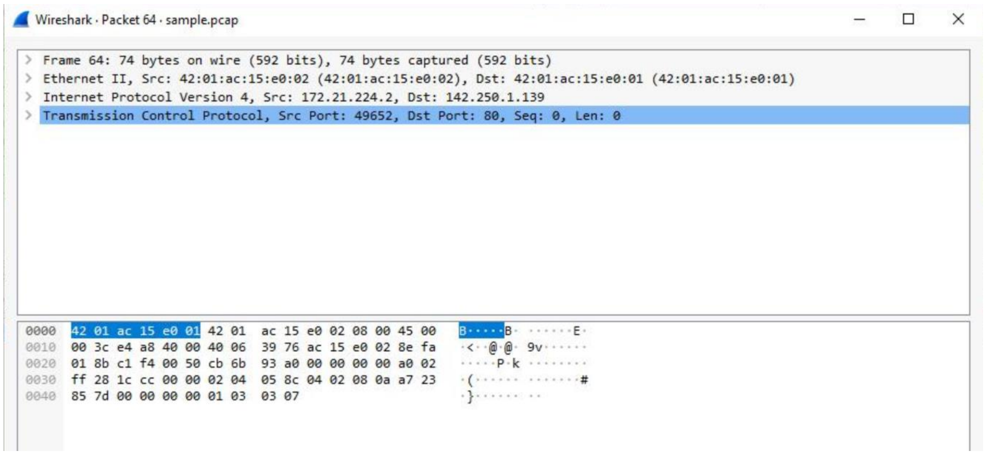
sample.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ip.addr == 142.250.1.139

```
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)
   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)
   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)
   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)
   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)
   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)
   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 TSec…
   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=4069…
   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
   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)
   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=40696749…
   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=406…
   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=280…
   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=40696749…
```

```
> 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:
> Internet Protocol Version 4, Src: 172.21.224.2, Dst: 142.250.1.139
> Internet Control Message Protocol
```

```
0000  42 01 ac 15 e0 01 42 01  ac 15 e0 02 08 00 45 00   B·····B· ······E·
0010  00 54 06 22 40 00 40 01  17 ea ac 15 e0 02 8e fa   ·T·"@·@· ········
0020  01 8b 08 00 42 fe 68 31  00 01 41 14 7e 63 00 00   ····B·h1 ··A·~c··
0030  00 00 cb 84 03 00 00 00  00 00 10 11 12 13 14 15   ········ ········
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ········ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                              67
```

sample.pcap
Packets: 200 · Displayed: 16 (8.0%)
Profile: Default

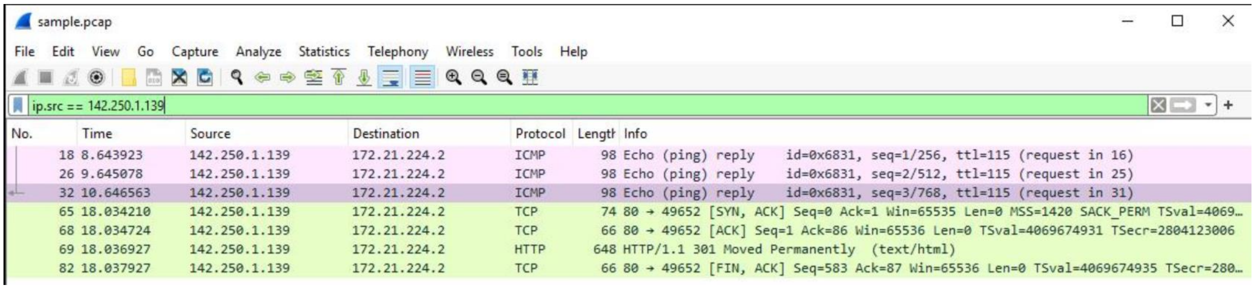


Wireshark · Packet 64 · sample.pcap

```
> Frame 64: 74 bytes on wire (592 bits), 74 bytes captured (592 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
> Transmission Control Protocol, Src Port: 49652, Dst Port: 80, Seq: 0, Len: 0
```

```
0000  42 01 ac 15 e0 01 42 01  ac 15 e0 02 08 00 45 00   B·····B· ······E·
0010  00 3c e4 a8 40 00 40 06  39 76 ac 15 e0 02 8e fa   ·<··@·@· 9v······
0020  01 8b c1 f4 00 50 cb 6b  93 a0 00 00 00 00 a0 02   ·····P·k ········
0030  ff 28 1c cc 00 00 02 04  05 8c 04 02 08 0a a7 23   ·(······ ·······#
0040  85 7d 00 00 00 00 01 03  03 07                     ·}······ ··
```

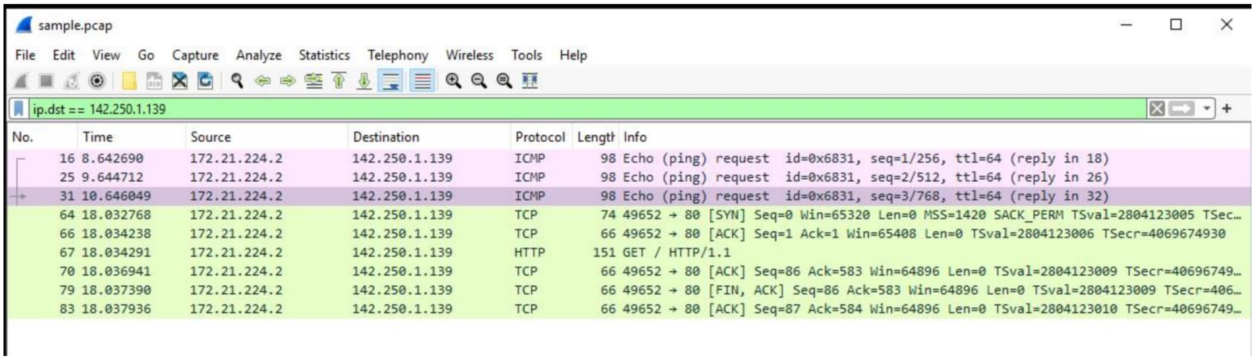
On the title bar, type ip.src == 142.250.1.139 to filter for traffic associated with a specific IP address. src means it is where the packet comes from.



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=4069…
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=280…

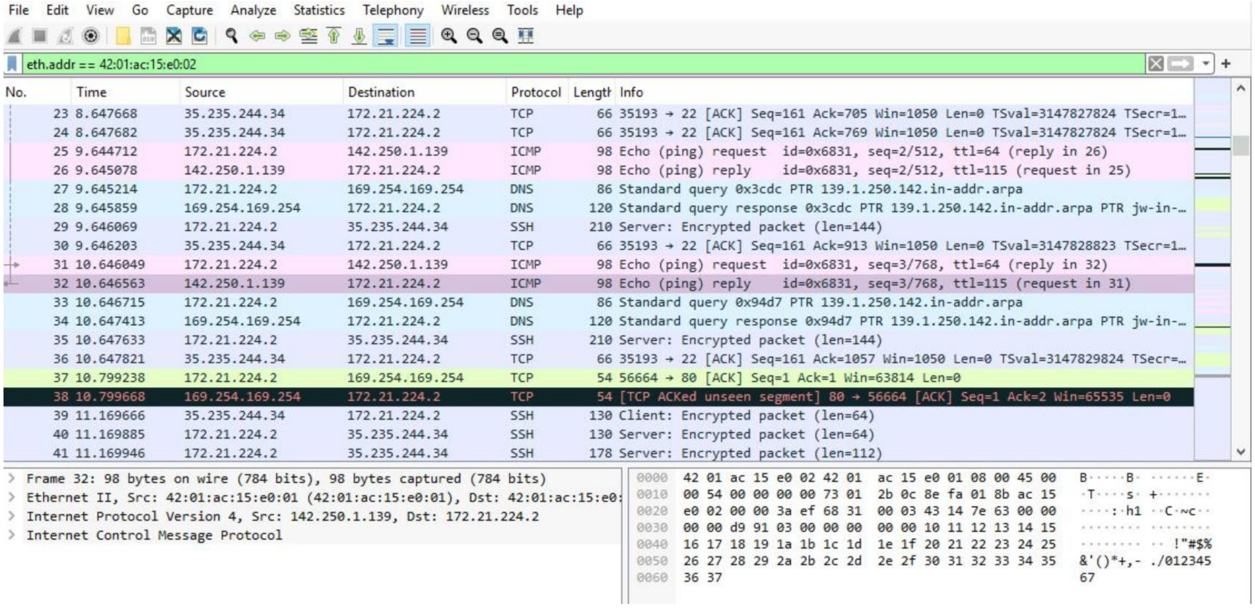
On the title bar, type ip.dst == 142.250.1.139 to filter for traffic associated with a specific IP address. dst means it is where the packet goes to.



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 TSec…
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=40696749…
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=406…
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=40696749…

On the title bar, type eth.addr == 42:01:ac:15:e0:02 to filter for traffic associated with a specific Ethernet MAC address. addr means either the source or the destination IP.



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

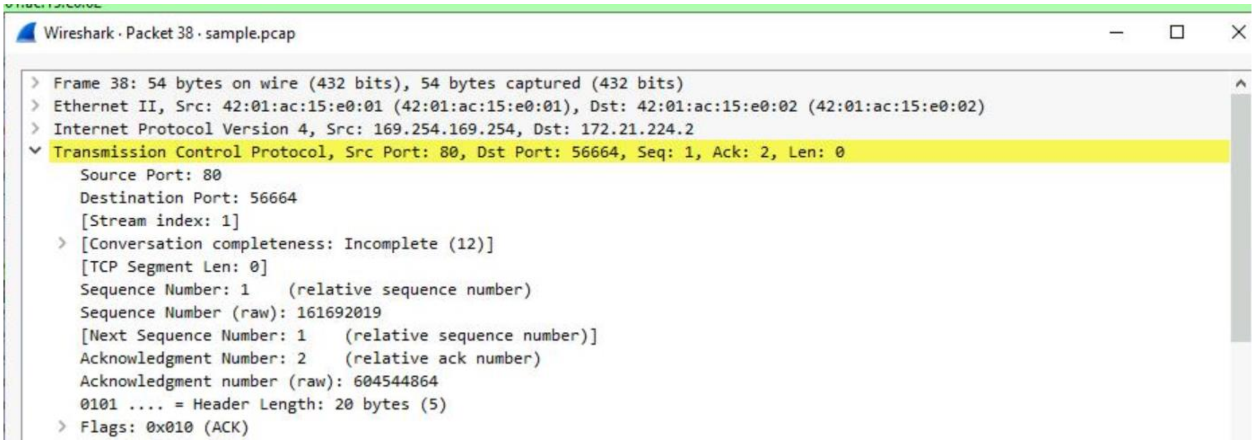
No.	Time	Source	Destination	Protocol	Length	Info
23	8.647668	35.235.244.34	172.21.224.2	TCP	66	35193 → 22 [ACK] Seq=161 Ack=705 Win=1050 Len=0 TSval=3147827824 TSecr=1…
24	8.647682	35.235.244.34	172.21.224.2	TCP	66	35193 → 22 [ACK] Seq=161 Ack=769 Win=1050 Len=0 TSval=3147827824 TSecr=1…
25	9.644712	172.21.224.2	142.250.1.139	ICMP	98	Echo (ping) request id=0x6831, seq=2/512, ttl=64 (reply in 26)
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)
27	9.645214	172.21.224.2	169.254.169.254	DNS	86	Standard query 0x3cdc PTR 139.1.250.142.in-addr.arpa
28	9.645859	169.254.169.254	172.21.224.2	DNS	120	Standard query response 0x3cdc PTR 139.1.250.142.in-addr.arpa PTR jw-in-…
29	9.646069	172.21.224.2	35.235.244.34	SSH	210	Server: Encrypted packet (len=144)
30	9.646203	35.235.244.34	172.21.224.2	TCP	66	35193 → 22 [ACK] Seq=161 Ack=913 Win=1050 Len=0 TSval=3147828823 TSecr=1…
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)
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)
33	10.646715	172.21.224.2	169.254.169.254	DNS	86	Standard query 0x94d7 PTR 139.1.250.142.in-addr.arpa
34	10.647413	169.254.169.254	172.21.224.2	DNS	120	Standard query response 0x94d7 PTR 139.1.250.142.in-addr.arpa PTR jw-in-…
35	10.647633	172.21.224.2	35.235.244.34	SSH	210	Server: Encrypted packet (len=144)
36	10.647821	35.235.244.34	172.21.224.2	TCP	66	35193 → 22 [ACK] Seq=161 Ack=1057 Win=1050 Len=0 TSval=3147829824 TSecr=…
37	10.799238	172.21.224.2	169.254.169.254	TCP	54	56664 → 80 [ACK] Seq=1 Ack=1 Win=63814 Len=0
38	10.799668	169.254.169.254	172.21.224.2	TCP	54	[TCP ACKed unseen segment] 80 → 56664 [ACK] Seq=1 Ack=2 Win=65535 Len=0
39	11.169666	35.235.244.34	172.21.224.2	SSH	130	Client: Encrypted packet (len=64)
40	11.169885	172.21.224.2	35.235.244.34	SSH	130	Server: Encrypted packet (len=64)
41	11.169946	172.21.224.2	35.235.244.34	SSH	178	Server: Encrypted packet (len=112)

```
> Frame 32: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)
> Ethernet II, Src: 42:01:ac:15:e0:01 (42:01:ac:15:e0:01), Dst: 42:01:ac:15:e0:
> Internet Protocol Version 4, Src: 142.250.1.139, Dst: 172.21.224.2
> Internet Control Message Protocol
```

```
0000  42 01 ac 15 e0 02 42 01  ac 15 e0 01 08 00 45 00   B·····B· ······E·
0010  00 54 00 00 00 00 73 01  2b 0c 8e fa 01 8b ac 15   ·T····s· +·······
0020  e0 02 00 00 3a ef 68 31  00 03 43 14 7e 63 00 00   ····:·h1 ··C·~c··
0030  00 00 d9 91 03 00 00 00  00 00 10 11 12 13 14 15   ········ ········
0040  16 17 18 19 1a 1b 1c 1d  1e 1f 20 21 22 23 24 25   ········ ·· !"#$%
0050  26 27 28 29 2a 2b 2c 2d  2e 2f 30 31 32 33 34 35   &'()*+,- ./012345
0060  36 37                                              67
```

2. Examine the protocols that are used when the user makes the connection to the website.

The TCP destination port of this TCP packet is 80 when ip.addr == 142.250.1.139 which contains the initial web request to an HTPP website that will typically be listening on TCP port 80.



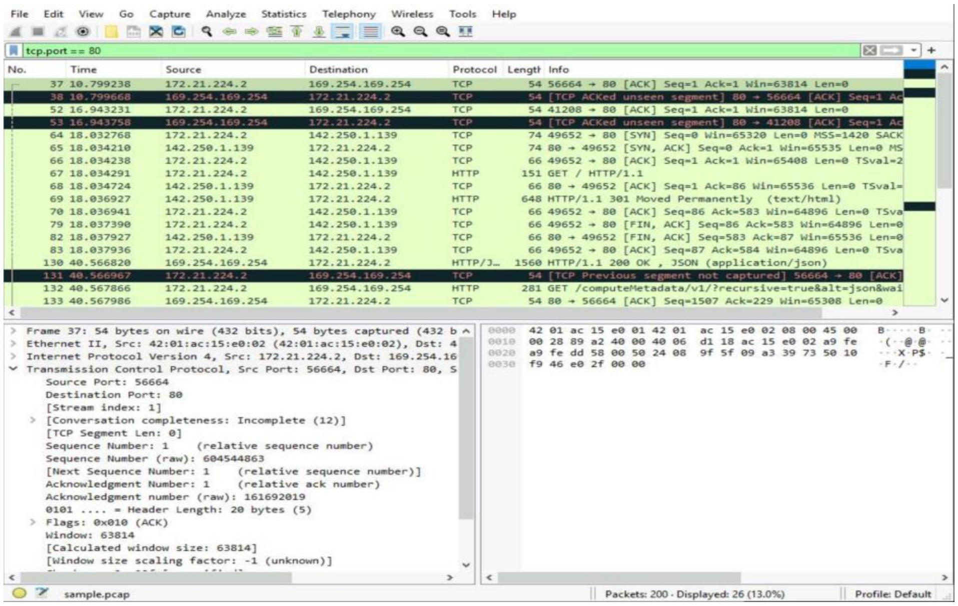
Wireshark · Packet 38 · sample.pcap

```
> Frame 38: 54 bytes on wire (432 bits), 54 bytes captured (432 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
v Transmission Control Protocol, Src Port: 80, Dst Port: 56664, Seq: 1, Ack: 2, Len: 0
    Source Port: 80
    Destination Port: 56664
    [Stream index: 1]
  > [Conversation completeness: Incomplete (12)]
    [TCP Segment Len: 0]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 161692019
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 2    (relative ack number)
    Acknowledgment number (raw): 604544864
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
```

The protocol destination port is TCP when Etherenet address was 42:01:ac:15:e0:02. Source address is 169.254.169.254 and the destination address is 172.21.224.2

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.

On the title bar, type tcp.port == 80 to filter for traffic associated with a specific port number. tcp.port == 80 means only the tcp port is 80 will be shown.



File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port == 80

No.	Time	Source	Destination	Protocol	Length	Info
37	10.799238	172.21.224.2	169.254.169.254	TCP	54	56664 → 80 [ACK] Seq=1 Ack=1 Win=63814 Len=0
38	10.799668	169.254.169.254	172.21.224.2	TCP	54	[TCP ACKed unseen segment] 80 → 56664 [ACK] Seq=1 Ac
52	16.943231	172.21.224.2	169.254.169.254	TCP	54	41208 → 80 [ACK] Seq=1 Ack=1 Win=63814 Len=0
53	16.943758	169.254.169.254	172.21.224.2	TCP	54	[TCP ACKed unseen segment] 80 → 41208 [ACK] Seq=1 Ac
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
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 MS
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=2
67	18.034291	172.21.224.2	142.250.1.139	HTTP	151	GET / HTTP/1.1
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=
69	18.036927	142.250.1.139	172.21.224.2	HTTP	648	HTTP/1.1 301 Moved Permanently (text/html)
70	18.036941	172.21.224.2	142.250.1.139	TCP	66	49652 → 80 [ACK] Seq=86 Ack=583 Win=64896 Len=0 TSva
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
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
83	18.037936	172.21.224.2	142.250.1.139	TCP	66	49652 → 80 [ACK] Seq=87 Ack=584 Win=64896 Len=0 TSva
130	40.566820	169.254.169.254	172.21.224.2	HTTP/J…	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]
132	40.567866	172.21.224.2	169.254.169.254	HTTP	281	GET /computeMetadata/v1/?recursive=true&alt=json&wai
133	40.567986	169.254.169.254	172.21.224.2	TCP	54	80 → 56664 [ACK] Seq=1507 Ack=229 Win=65308 Len=0

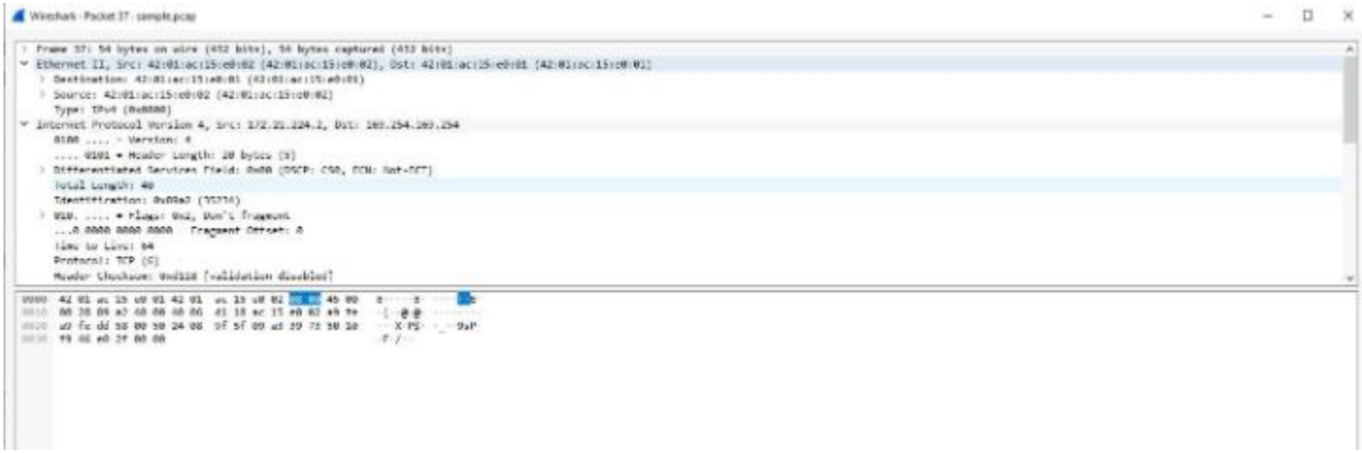
```
> Frame 37: 54 bytes on wire (432 bits), 54 bytes captured (432 b
> Ethernet II, Src: 42:01:ac:15:e0:02 (42:01:ac:15:e0:02), Dst: 4
> Internet Protocol Version 4, Src: 172.21.224.2, Dst: 169.254.16
v Transmission Control Protocol, Src Port: 56664, Dst Port: 80, S
    Source Port: 56664
    Destination Port: 80
    [Stream index: 1]
  > [Conversation completeness: Incomplete (12)]
    [TCP Segment Len: 0]
    Sequence Number: 1    (relative sequence number)
    Sequence Number (raw): 604544863
    [Next Sequence Number: 1    (relative sequence number)]
    Acknowledgment Number: 1    (relative ack number)
    Acknowledgment number (raw): 161692019
    0101 .... = Header Length: 20 bytes (5)
  > Flags: 0x010 (ACK)
    Window: 63814
    [Calculated window size: 63814]
    [Window size scaling factor: -1 (unknown)]
```

```
0000  42 01 ac 15 e0 01 42 01  ac 15 e0 02 08 00 45 00   B·····B·
0010  00 28 89 a2 40 00 40 06  d1 18 ac 15 e0 02 a9 fe   ·(··@·@·
0020  a9 fe dd 58 00 50 24 08  9f 5f 09 a3 39 73 50 10   ···X·P$·
0030  f9 46 e0 2f 00 00                                  ·F·/··
```

sample.pcap | Packets: 200 · Displayed: 26 (13.0%) | Profile: Default

When the filter tcp.port == 80 sets in play, the time to live is 64.

Time to Live: A field in the Internet Protocol (IP) header that indicates the maximum amount of time an IP packet is allowed to exist in the network before it is discarded if it has not reached its destination. TTL is used to prevent packets from circulating indefinitely in the network, which could happen in the case of routing loops. It can be used as a basic security measure to limit how far packets can propagate through the network.



Wireshark · Packet 37 · sample.pcap

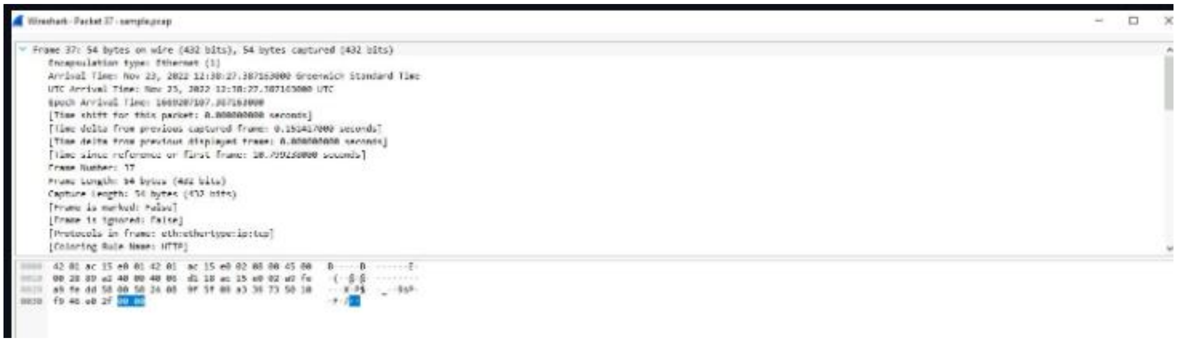
```
> Frame 37: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)
v Ethernet II, Src: 42:01:ac:15:e0:02 (42:01:ac:15:e0:02), Dst: 42:01:ac:15:e0:01 (42:01:ac:15:e0:01)
  > Destination: 42:01:ac:15:e0:01 (42:01:ac:15:e0:01)
  > Source: 42:01:ac:15:e0:02 (42:01:ac:15:e0:02)
    Type: IPv4 (0x0800)
v Internet Protocol Version 4, Src: 172.21.224.2, Dst: 169.254.169.254
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 40
    Identification: 0x89a7 (35239)
  > 010. .... = Flags: 0x2, Don't fragment
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: TCP (6)
    Header Checksum: 0xd118 [validation disabled]

0000  42 01 ac 15 e0 01 42 01  ac 15 e0 02 08 00 45 00   B·····B·······E·
0010  00 28 89 a7 40 00 40 06  d1 18 ac 15 e0 02 a9 fe   ·(··@·@·········
0020  a9 fe dd 58 00 50 24 08  9f 5f 00 a3 39 73 50 10   ···X·P$··_··9sP·
0030  f9 46 e0 2f 00 00                                  ·F·/··
```

When the filter tcp.port == 80 sets in play, the Frame Number is 37 and Frame Length is 54 bytes.

Frame Number: This is essentially the sequence number of a packet within a particular capture. It helps you identify and refer to packets more easily. In your case, a frame number of 37 means it's the 37th packet captured since the beginning of the capture session. This number is assigned sequentially as packets are captured, starting with the number 1 for the first packet.

Frame Length: This indicates the size of the packet, including all headers and payload, measured in bytes. The frame length of 54 bytes means the total size of the packet is 54 bytes. This size includes everything from the lowest layer (physical layer) up to the highest layer present in the packet that Wireshark can decode. It's useful for understanding the size of the data being transmitted and can help in various analyses, such as identifying potential issues with packet sizes that might indicate fragmentation or other problems.



Wireshark · Packet 37 · sample.pcap

```
v Frame 37: 54 bytes on wire (432 bits), 54 bytes captured (432 bits)
    Encapsulation type: Ethernet (1)
    Arrival Time: Nov 23, 2022 12:38:27.387163000 Greenwich Standard Time
    UTC Arrival Time: Nov 23, 2022 12:38:27.387163000 UTC
    Epoch Arrival Time: 1669207107.387163000
    [Time shift for this packet: 0.000000000 seconds]
    [Time delta from previous captured frame: 0.151417000 seconds]
    [Time delta from previous displayed frame: 0.000000000 seconds]
    [Time since reference or first frame: 10.759258000 seconds]
    Frame Number: 37
    Frame Length: 54 bytes (432 bits)
    Capture Length: 54 bytes (432 bits)
    [Frame is marked: False]
    [Frame is ignored: False]
    [Protocols in frame: eth:ethertype:ip:tcp]
    [Coloring Rule Name: HTTP]

0000  42 01 ac 15 e0 01 42 01  ac 15 e0 02 08 00 45 00   B·····B·······E·
0010  00 28 89 a7 40 00 40 06  d1 18 ac 15 e0 02 a9 fe   ·(··@·@·········
0020  a9 fe dd 58 00 50 24 08  9f 5f 00 a3 39 73 50 10   ···X·P$··_··9sP·
0030  f9 46 e0 2f 00 00                                  ·F·/··
```