Task:

In this lab activity, you'll use Wireshark to examine a sample packet capture file and filter the network traffic data.

Scenario:

In this scenario, you're a security analyst investigating traffic to a website.

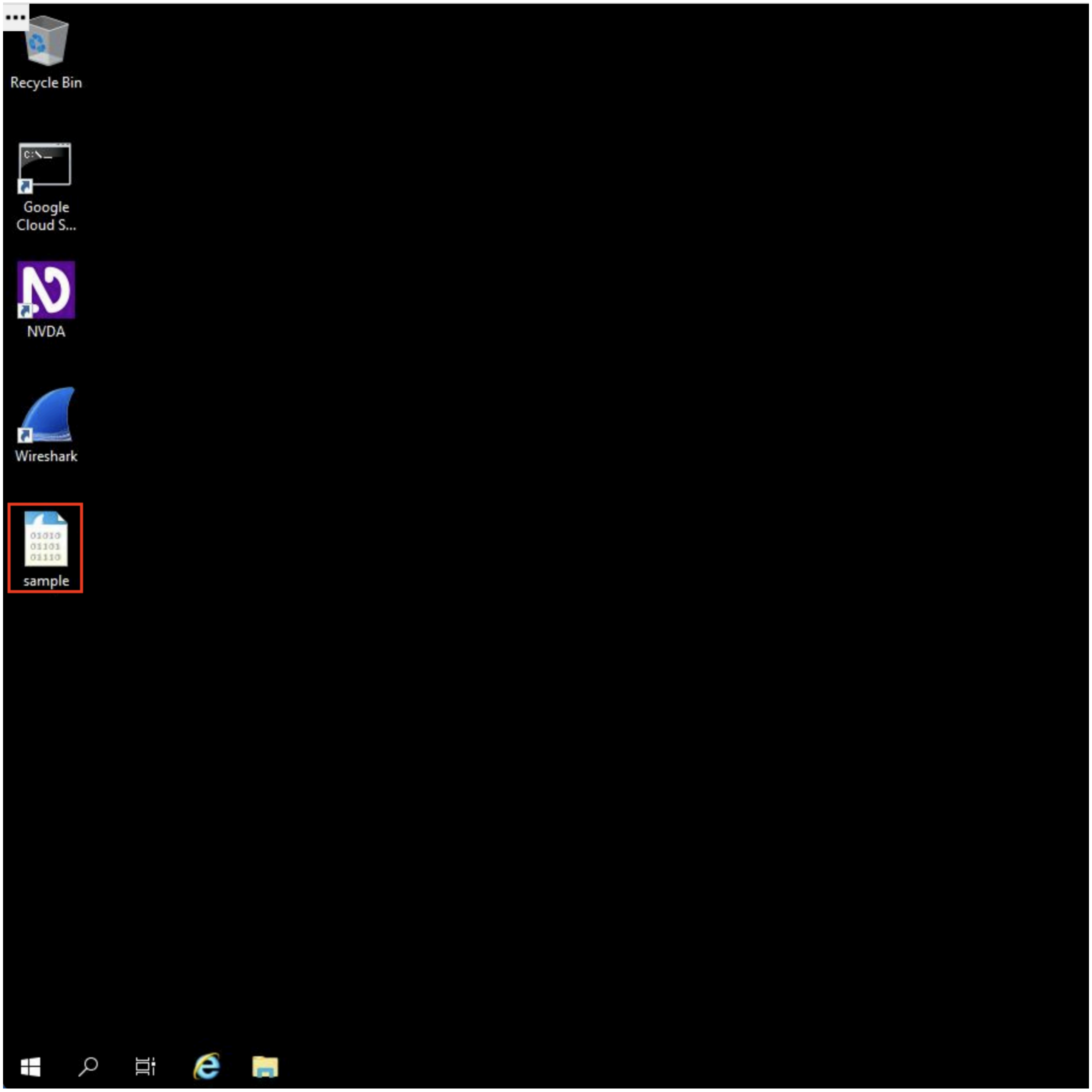
You'll analyze a network packet capture file that contains traffic data related to a user connecting to an internet site. The ability to filter network traffic using packet sniffers to gather relevant information is an essential skill as a security analyst.

You must filter the data in order to:

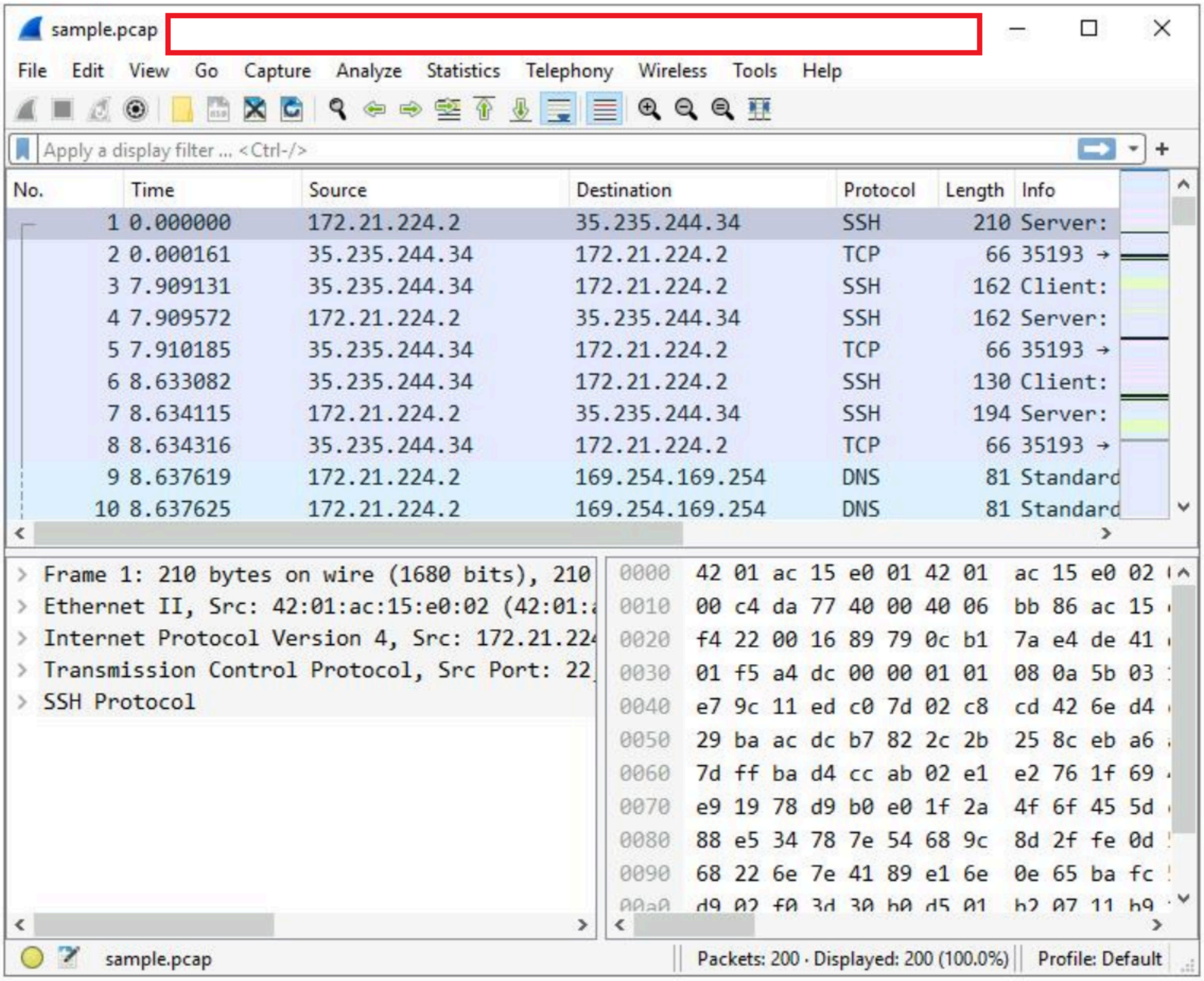
- identify the source and destination IP addresses involved in this web browsing session,
- examine the protocols that are used when the user makes the connection to the website, and
- analyze some of the data packets to identify the type of information sent and received by the systems that connect to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

Task 1. Explore data with Wireshark

1. **To open the packet capture file, double-click the sample file on the Windows desktop. This will start Wireshark.**


Recycle Bin
Google Cloud S...
NVDA
Wireshark
sample

2. **Double-click the Wireshark title bar next to the sample.pcap filename to maximize the Wireshark application window.**


sample.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
sample.pcap
Packets: 200 · Displayed: 200 (100.0%)
Profile: Default

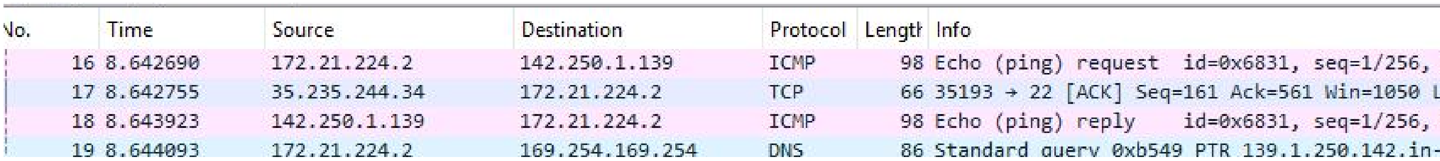
```
No.   Time       Source          Destination      Protocol  Length Info
    1 0.000000   172.21.224.2    35.235.244.34    SSH       210 Server:
    2 0.000161   35.235.244.34   172.21.224.2     TCP        66 35193 →
    3 7.909131   35.235.244.34   172.21.224.2     SSH       162 Client:
    4 7.909572   172.21.224.2    35.235.244.34    SSH       162 Server:
    5 7.910185   35.235.244.34   172.21.224.2     TCP        66 35193 →
    6 8.633082   35.235.244.34   172.21.224.2     SSH       130 Client:
    7 8.634115   172.21.224.2    35.235.244.34    SSH       194 Server:
    8 8.634316   35.235.244.34   172.21.224.2     TCP        66 35193 →
    9 8.637619   172.21.224.2    169.254.169.254  DNS        81 Standard
   10 8.637625   172.21.224.2    169.254.169.254  DNS        81 Standard
```

```
> Frame 1: 210 bytes on wire (1680 bits), 210
> Ethernet II, Src: 42:01:ac:15:e0:02 (42:01:
> Internet Protocol Version 4, Src: 172.21.22
> Transmission Control Protocol, Src Port: 22
> SSH Protocol
```

```
0000  42 01 ac 15 e0 01 42 01  ac 15 e0 02
0010  00 c4 da 77 40 00 40 06  bb 86 ac 15
0020  f4 22 00 16 89 79 0c b1  7a e4 de 41
0030  01 f5 a4 dc 00 00 01 01  08 0a 5b 03
0040  e7 9c 11 ed c0 7d 02 c8  cd 42 6e d4
0050  29 ba ac dc b7 82 2c 2b  25 8c eb a6
0060  7d ff ba d4 cc ab 02 e1  e2 76 1f 69
0070  e9 19 78 d9 b0 e0 1f 2a  4f 6f 45 5d
0080  88 e5 34 78 7e 54 68 9c  8d 2f fe 0d
0090  68 22 6e 7e 41 89 e1 6e  0e 65 ba fc
00a0  d9 02 f0 3d 30 b0 d5 01  b2 07 11 b9
```

3. **Scroll down the packet list until a packet is listed where the info column starts with the words 'Echo (ping) request'.**

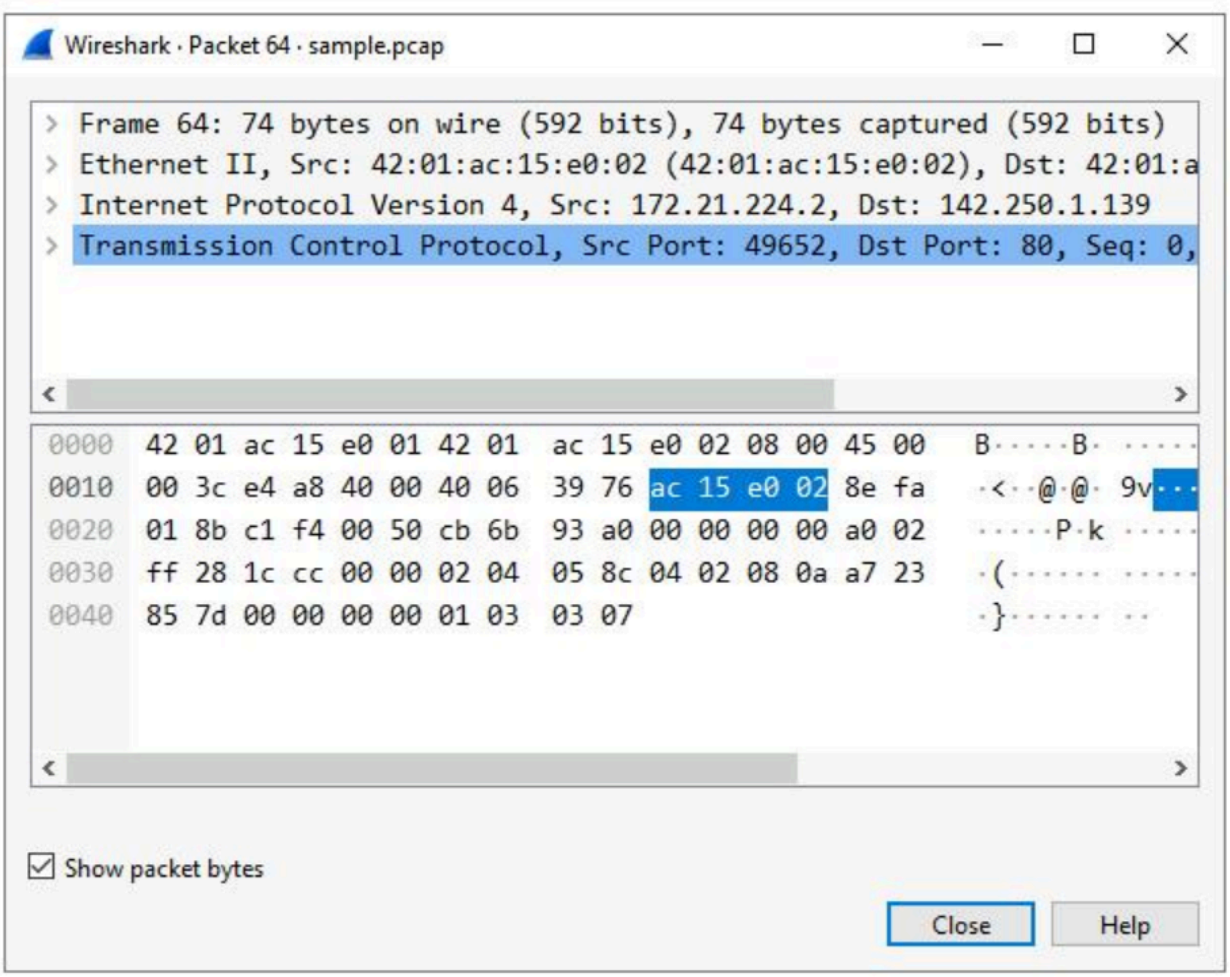
    a. What is the protocol of the first packet in the list where the info column starts with the words 'Echo (ping) request'?: **ICMP is the protocol type listed for the first (and all) packets that contain 'Echo (ping) request' in the info column.**



```
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,
   17 8.642755   35.235.244.34   172.21.224.2     TCP        66 35193 → 22 [ACK] Seq=161 Ack=561 Win=1050 L
   18 8.643923   142.250.1.139   172.21.224.2     ICMP       98 Echo (ping) reply    id=0x6831, seq=1/256,
   19 8.644093   172.21.224.2    169.254.169.254  DNS        86 Standard query 0xb549 PTR 139.1.250.142.in-
```

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

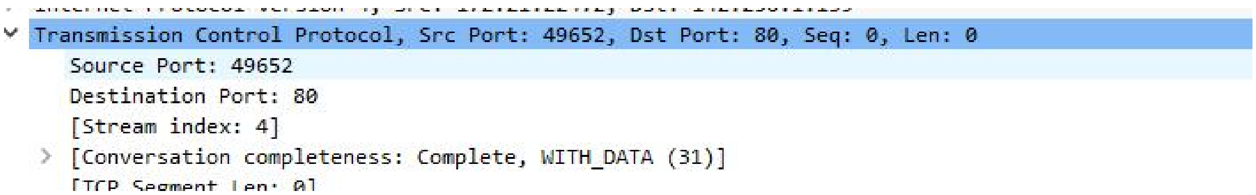
1. **Enter the following filter for traffic associated with a specific IP address. Enter this into the Apply a display filter: ip.addr == 142.250.1.139**
2. **Press ENTER or click the Apply display filter icon in the filter text box.**
3. **Double-click the first packet that lists TCP as the protocol.**



```
> 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:a
> 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,
```

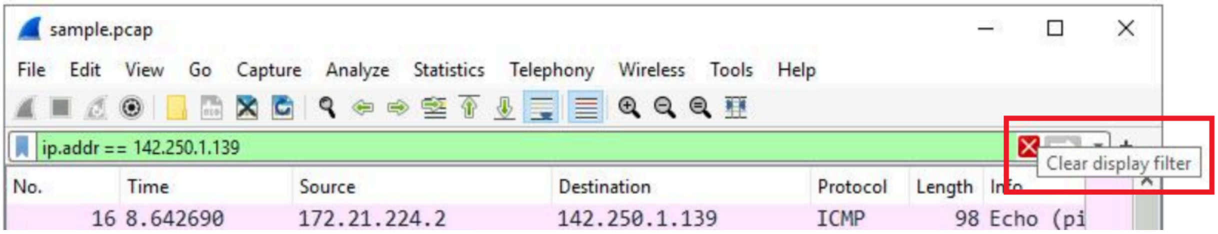
```
0000  42 01 ac 15 e0 01 42 01  ac 15 e0 02 08 00 45 00   B·····B· ·····
0010  00 3c e4 a8 40 00 40 06  39 76 ac 15 e0 02 8e fa   ·<··@·@· 9v···
0020  01 8b c1 f4 00 50 cb 6b  93 a0 00 00 00 00 a0 02   ·····P·k ·····
0030  ff 28 1c cc 00 00 02 04  05 8c 04 02 08 0a a7 23   ·(······ ·····
0040  85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

4. **Double-click the Transmission Control Protocol subtree.**
   a. What is the TCP destination port of this TCP packet?:



```
> Internet Protocol Version 4, Src: 172.21.224.2, Dst: 142.250.1.139
v Transmission Control Protocol, Src Port: 49652, Dst Port: 80, Seq: 0, Len: 0
      Source Port: 49652
      Destination Port: 80
      [Stream index: 4]
    > [Conversation completeness: Complete, WITH_DATA (31)]
      [TCP Segment Len: 0]
```

5. **Click the X Clear display filter icon in the Wireshark filter bar to clear the IP address filter.**



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 (pi