

# Simple Protocols Traffic Analysis

Kevin Scrivnor

COMP 429

Spring 2023

# Recall: A network protocol defines how these devices talk to each other

- The design goals behind a protocol are
  - **Reliability**
    - Recover from errors, failures
  - **Resource allocation**
    - Sharing access to common, limited resources
  - **Evolvability**
    - Allow of change over time
  - **Security**
    - Defending the network against various attacks

# What does a protocol define?

- Types of messages exchanged
  - (what kind of messages)
  - **Consider what information needs to be transferred.**
- Syntax of the various types of messages
  - (format of messages)
  - **Text based or Binary? Line endings?**
- Semantics of the fields in the messages
  - (meaning of messages)
  - **Define and document the meaning of messages**
- and rules for when messages are exchanged
  - (sending and responding)
  - **Describe actions to be taken, define error codes/messages, think about everything that can go wrong.**

# Consider a file transferring networked program

- Types of messages exchanged
  - **What are we transferring?**
- Syntax of the various types of messages
  - **Binary based or text based?**
- Semantics of the fields in the messages
  - **What is the meaning behind our messages?**
- and rules for when messages are exchanged
  - **What is the sender supposed to be doing?**
  - **What is the receiver supposed to be doing?**

# Where are standard protocols defined?

- There is a long process to becoming an Internet Standard
- Ultimately ends up in an RFC (Request For Comments)
- For instance, the Trivial File Transfer Protocol (Revision 2) is defined in RFC 1350
  - <https://datatracker.ietf.org/doc/html/rfc1350>
- Sometimes, features get added later. Such as the blksize option:
  - <https://datatracker.ietf.org/doc/html/rfc2348>
- Reading an RFC is like reading a very detailed and long lab assignment. Every little sentence can make a huge difference in the implementation.

# Wireshark

# Solving Networking Problems is Hard

- Do I have an IP?
- Do I have connectivity on the network?
- Is my local firewall blocking traffic?
- Is the LAN firewall blocking traffic?
- Is DNS not resolving domain names?
- Am I routing packets correctly?
- Is the router routing packets correctly?
- Is the problem on my end or your end?
- WHAT THE HELL IS HAPPENING??
- *You can only solve a problem if you can see it*

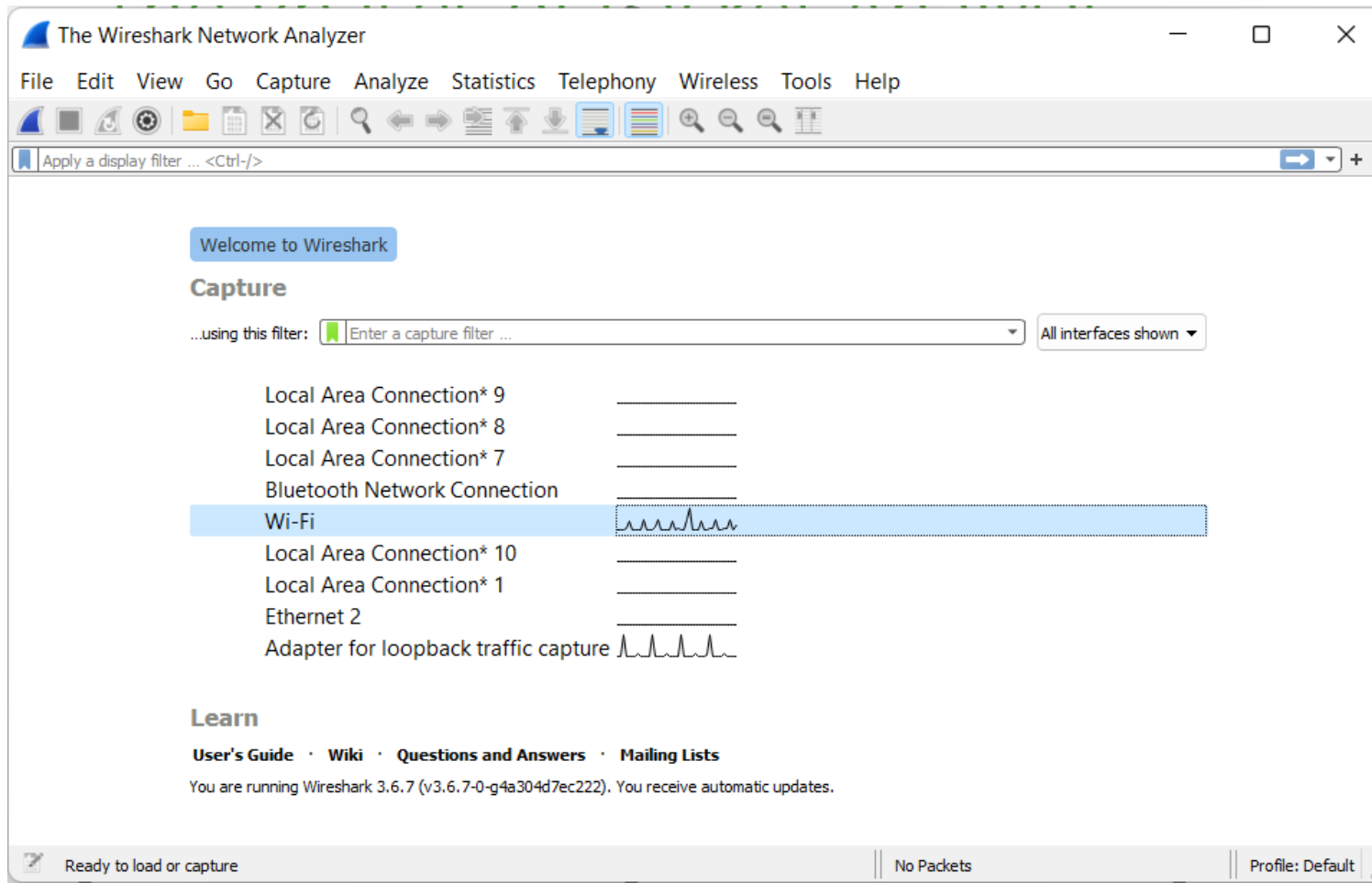
# Packet Analysis – who looks at the packets?

- Packet Capture files, or PCAP files contain data captured from some network interface
- **Network admins** – gain information about current conditions
- **Security analysts** – anything suspicious, forensics
- **Students** – learning tool to see how networking works
- **Hackers** – sniffing network traffic for enumeration
- Usually referred to as “**sniffing**”
  - As early packet tools **sniffed** the network for packets



# The Phases of Packet Analysis: Gather

- Gather/capture packets
- Select the network interface you want to capture on

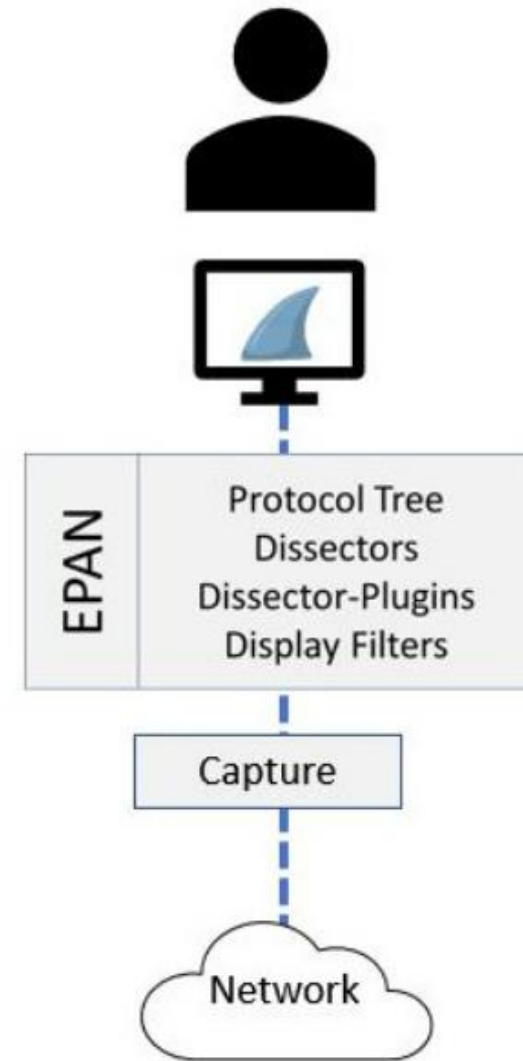


Analyze

Display

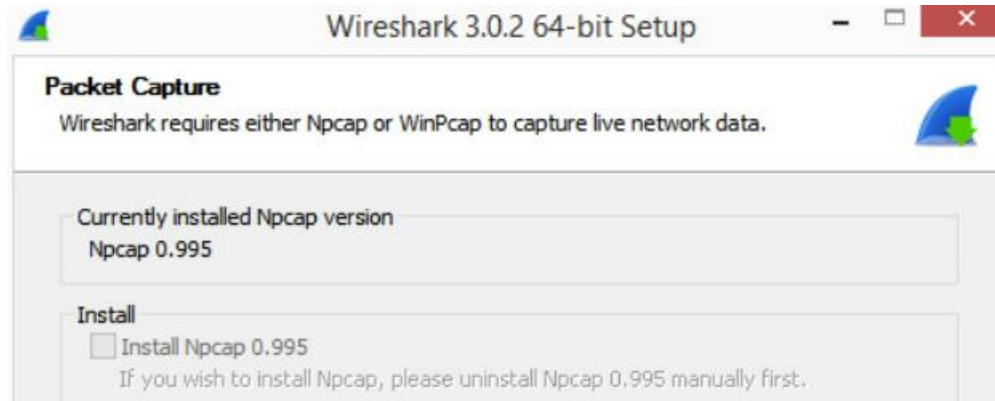
Decode

Gather



# The Phases of Packet Analysis: Capture

- Wireshark uses a capture engine: libnpcap
- This was installed with Wireshark

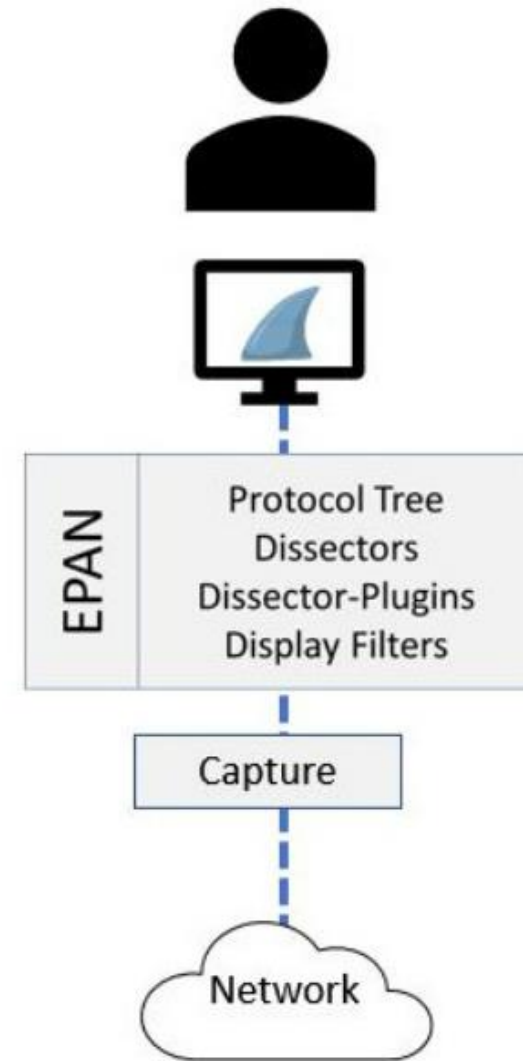


Analyze

Display

Decode

Gather



# The Phases of Packet Analysis: Decode

- The capture engine captures the raw bits of traffic going through the selected interface
- Wireshark then decodes the bits into human readable information

```
00101010 01001001 11011000 10111001
10000101 10000100 00000000 01010000
10101101 11010110 00011000 01111100
```



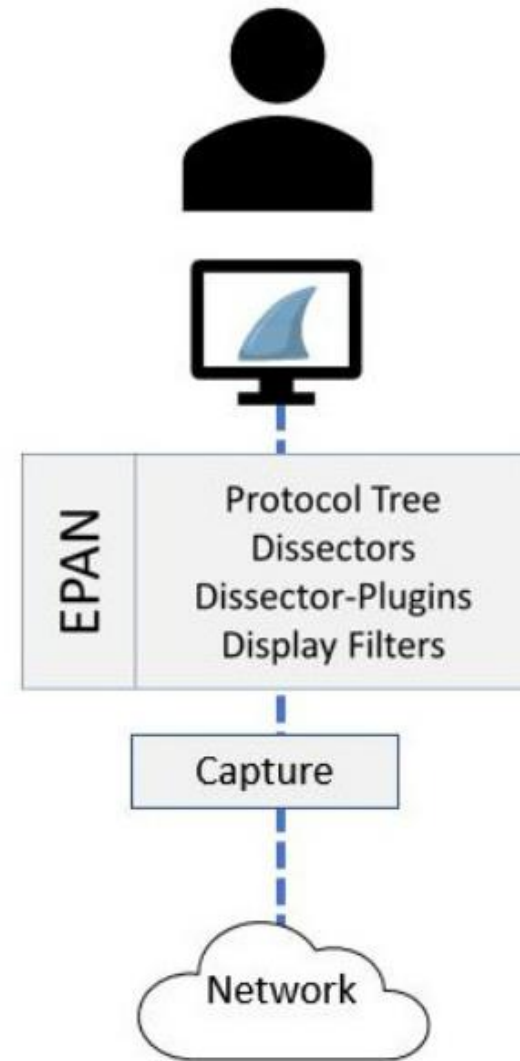
Source IP: 216.185.152.112  
Source Port: 80  
Destination IP: 172.16.133.132  
Destination Port: 54627  
Protocol: HTTP

Analyze

Display

Decode

Gather



# The Phases of Packet Analysis: Display

- Packet List, top
- Packet Details, left
- Packet Bytes, right

The screenshot shows the Wireshark network protocol analyzer interface. The top pane displays a list of captured packets. The middle pane shows the details of the selected packet (Transmission Control Protocol). The bottom pane shows the raw packet bytes in hexadecimal and ASCII.

No.	Time	Source	Destination	Protocol	Length	Info
13	2.085279	3.227.250.203	192.168.254.90	TCP	1494	443
14	2.085344	192.168.254.90	3.227.250.203	TCP	54	57060
15	2.085655	3.227.250.203	192.168.254.90	TCP	1494	443
16	2.085655	3.227.250.203	192.168.254.90	TLSv1.2	906	Cert
17	2.085655	3.227.250.203	192.168.254.90	TLSv1.2	392	Serv
18	2.085655	3.227.250.203	192.168.254.90	TLSv1.2	63	Serv
19	2.085752	192.168.254.90	3.227.250.203	TCP	54	57060

Transmission Control Protocol, Src Port 443  
Source Port: 443  
Destination Port: 57060  
[Stream index: 2]  
[Conversation completeness: Complete]  
[TCP Segment Len: 1440]  
Sequence Number: 1541 (relative sequence number)  
Sequence Number (raw): 535152916

0020 fe 5a 01 bb de e4 1f e5 c9 14 81  
0030 01 f9 96 63 00 00 8b bf 29 d7 68  
0040 a8 75 86 be 97 ea fc 77 21 7b c0  
0050 e0 30 0d 06 09 2a 86 48 86 f7 0d  
0060 03 82 01 01 00 76 06 61 51 33 b0  
0070 f6 63 bc 29 1e 7c af a5 4e fd 73  
0080 a6 ae 20 81 2e 96 ed 90 8f 7b 31  
0090 5a 58 30 6f 66 25 9e ae d6 af 90  
00a0 35 c0 d7 c1 7a 53 c7 35 cf ce 2c

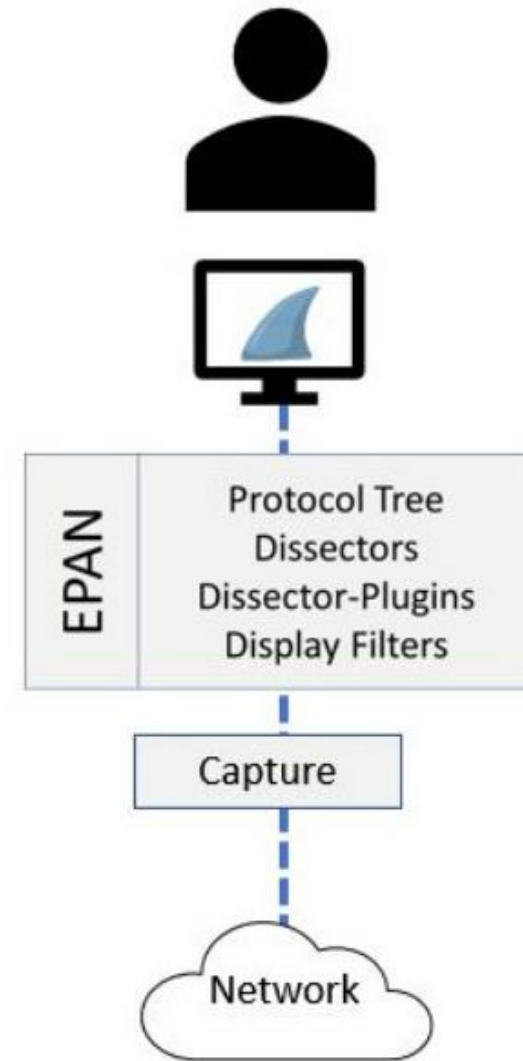
Destination Port (tcp.dstport), 2 bytes  
Packets: 44 · Displayed: 44 (100.0%) · Dropped: 0 (0.0%) Profile: Default

Analyze

Display

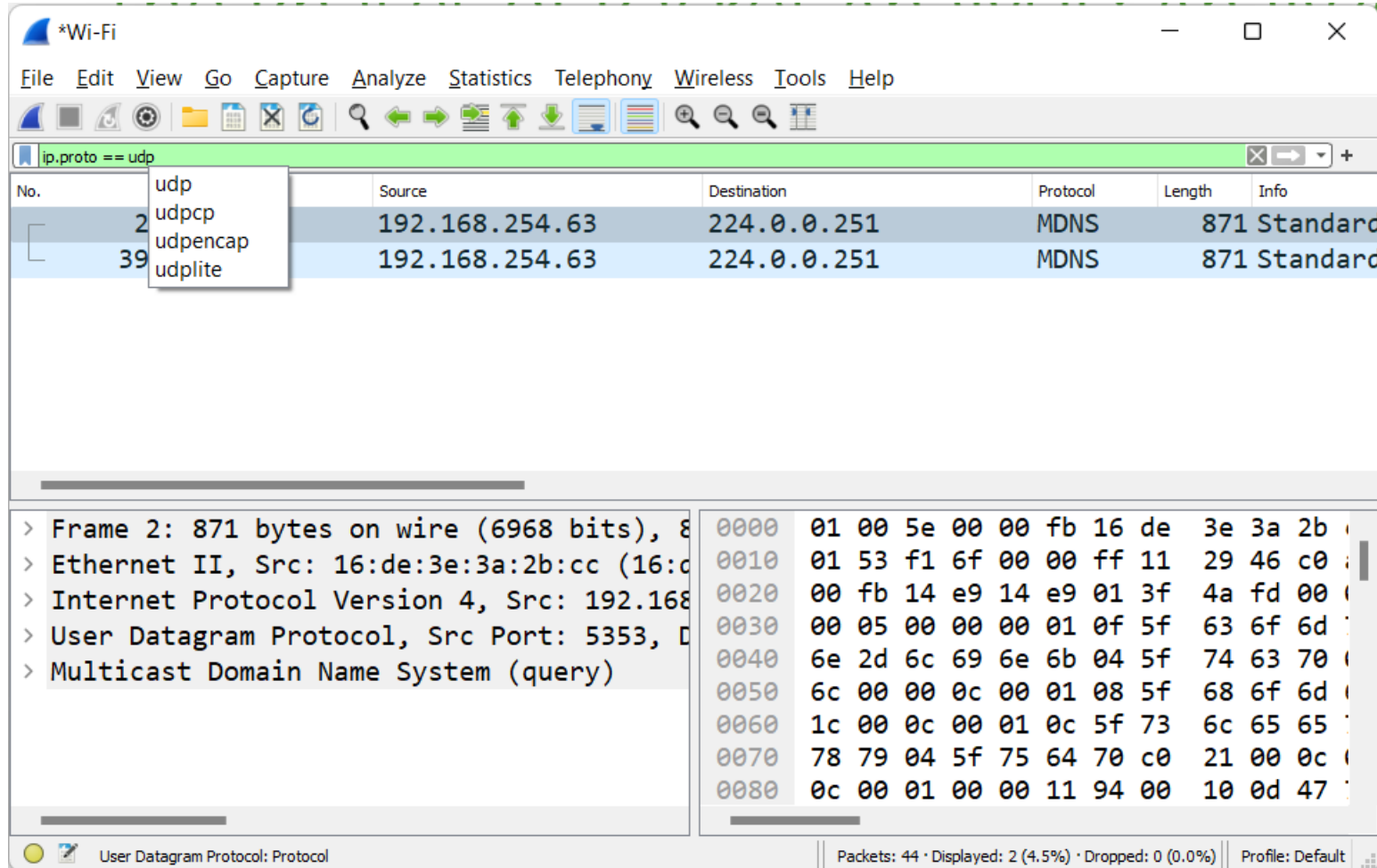
Decode

Gather



# The Phases of Packet Analysis: Analyze

- Apply a display filter to look for specific packets



The screenshot shows the Wireshark interface with the display filter `ip.proto == udp` applied. The packet list shows two packets of type MDNS. The packet details pane shows the structure of the selected packet (Frame 2).

No.	Source	Destination	Protocol	Length	Info
2	192.168.254.63	224.0.0.251	MDNS	871	Standard
39	192.168.254.63	224.0.0.251	MDNS	871	Standard

Packet details for Frame 2:

- > Frame 2: 871 bytes on wire (6968 bits), 8
- > Ethernet II, Src: 16:de:3e:3a:2b:cc (16:c
- > Internet Protocol Version 4, Src: 192.168
- > User Datagram Protocol, Src Port: 5353, D
- > Multicast Domain Name System (query)

Hex dump of the packet data:

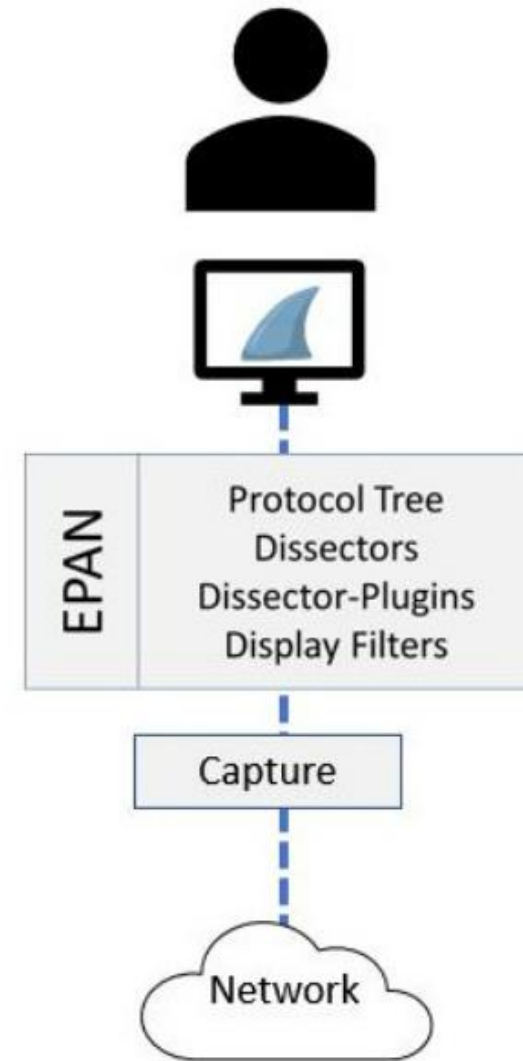
```
0000 01 00 5e 00 00 fb 16 de 3e 3a 2b 01 00 00 00
0010 01 53 f1 6f 00 00 ff 11 29 46 c0 00 00 00 00
0020 00 fb 14 e9 14 e9 01 3f 4a fd 00 00 00 00 00 00
0030 00 05 00 00 00 01 0f 5f 63 6f 6d 00 00 00 00
0040 6e 2d 6c 69 6e 6b 04 5f 74 63 70 00 00 00 00
0050 6c 00 00 0c 00 01 08 5f 68 6f 6d 00 00 00 00
0060 1c 00 0c 00 01 0c 5f 73 6c 65 65 00 00 00 00
0070 78 79 04 5f 75 64 70 c0 21 00 0c 00 00 00 00
0080 0c 00 01 00 00 11 94 00 10 0d 47 00 00 00 00
```

Analyze

Display

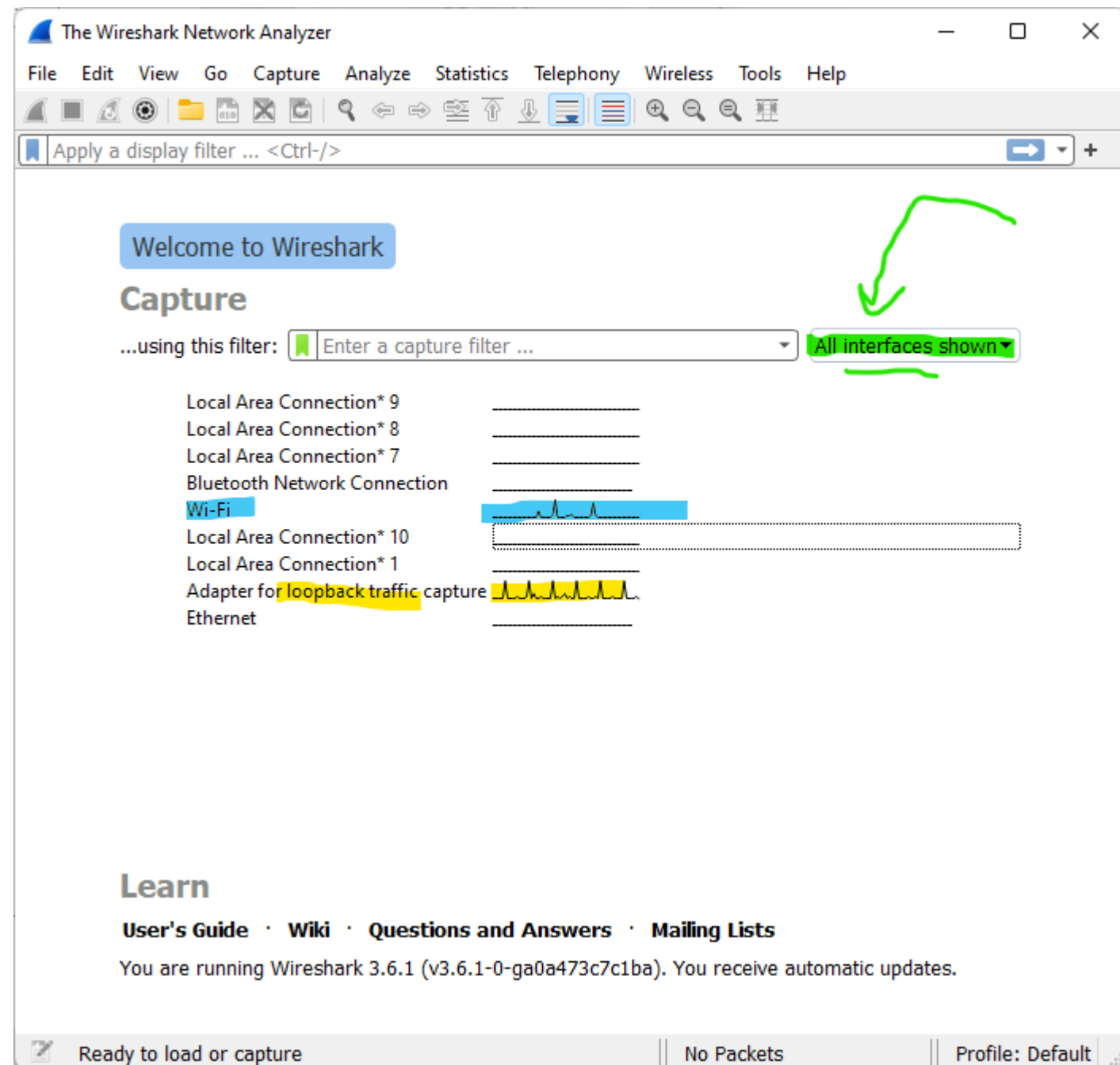
Decode

Gather



# First View of Wireshark

- You can limit the interfaces with choices of wired/wireless/Bluetooth as you see fit.
- My computer is on wifi, so I'll be capturing network traffic from this interface.
- The loopback is traffic from my computer to my computer. I'm not sure what Windows 11 is doing with all this local traffic...





Capturing from Wi-Fi

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
13890	56.780584	74.125.103.12	192.168.254.90	QUIC	1292	Protected Payload (KP0)
13891	56.781452	74.125.103.12	192.168.254.90	QUIC	1292	Protected Payload (KP0)
13892	56.781452	74.125.103.12	192.168.254.90	QUIC	1292	Protected Payload (KP0)
13893	56.781452	74.125.103.12	192.168.254.90	QUIC	1292	Protected Payload (KP0)
13894	56.781452	74.125.103.12	192.168.254.90	QUIC	1292	Protected Payload (KP0)
13895	56.781452	74.125.103.12	192.168.254.90	QUIC	1292	Protected Payload (KP0)
13896	56.781452	74.125.103.12	192.168.254.90	QUIC	1292	Protected Payload (KP0)
13897	56.781452	74.125.103.12	192.168.254.90	QUIC	1292	Protected Payload (KP0)

> Frame 12506: 79 bytes on wire (632 bits), 79 bytes captured (632 bits) on interface \Device\NPF\_{274205E5-AEF1-4D}

> Ethernet II, Src: IntelCor\_54:6b:27 (dc:21:48:54:6b:27), Dst: ARRISGro\_d0:30:40 (a8:97:cd:d0:30:40)

> Internet Protocol Version 4, Src: 192.168.254.90, Dst: 142.250.68.78

> User Datagram Protocol, Src Port: 54692, Dst Port: 443

> QUIC IETF

0000 a8 97 cd d0 30 40 dc 21 48 54 6b 27 08 00 45 00 ...0@.!.HTK'..E-

0010 00 41 b5 9a 40 00 80 11 00 00 c0 a8 fe 5a 8e fa -A..@... ..Z..

0020 44 4e d5 a4 01 bb 00 2d 92 8a 45 53 4f 22 34 10 DN.....- ..ESO"4-

0030 b0 bb e9 7b 3d 76 ca 27 5f 6d f2 08 73 f8 d0 18 ...{=v.' \_m..s...

0040 1e 3f c4 af d3 4e 92 c8 39 96 64 3c bc 39 ef -?...N.. 9.d<.9-

Destination Hardware Address (eth.dst), 6 bytes

Packets: 14316 · Displayed: 14316 (100.0%)

Profile: Default

# Reading Hex

0000	a8 97 cd d0 30 40 dc 21 48 54 6b 27 08 00 45 b8	....0@. ! HTk'..E.
0010	00 cb a5 19 40 00 80 06 00 00 c0 a8 fe 5a 8a c5	....@... ..Z..
0020	c0 4e c0 13 00 50 89 1c 12 63 08 cd b1 40 50 18	.N...P... .c...@P.
0030	04 02 74 2f 00 00 47 45 54 20 2f 20 48 54 54 50	..t/...GE T / HTTP
0040	2f 31 2e 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74	/1.1...Us er-Agent
0050	3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 57	: Mozill a/5.0 (W
0060	69 6e 64 6f 77 73 20 4e 54 3b 20 57 69 6e 64 6f	indows N T; Windo
0070	77 73 20 4e 54 20 31 30 2e 30 3b 20 65 6e 2d 55	ws NT 10 .0; en-U
0080	53 29 20 57 69 6e 64 6f 77 73 50 6f 77 65 72 53	S) Windo wsPowerS
0090	68 65 6c 6c 2f 35 2e 31 2e 32 32 30 30 30 2e 32	hell/5.1 .22000.2
00a0	38 32 0d 0a 48 6f 73 74 3a 20 73 63 72 69 76 6e	82..Host : scrivn
00b0	6f 72 2e 63 69 6b 65 79 73 2e 63 6f 6d 0d 0a 43	or.cikey s.com..C
00c0	6f 6e 6e 65 63 74 69 6f 6e 3a 20 4b 65 65 70 2d	onnectio n: Keep-
00d0	41 6c 69 76 65 0d 0a 0d 0a	Alive... .



# The 5 Layer Model in Action

The image shows a Wireshark packet capture window titled "Capturing from Wi-Fi". The main packet list displays several QUIC packets (No. 13890-13897) with the same source and destination IP addresses (74.125.103.12 to 192.168.254.90) and a length of 1292 bytes. The selected packet (No. 13897) is expanded to show its details:

- > Frame 12506: 79 bytes on wire (632 bits), 79 bytes captured (632 bits) on interface \Device\NPF\_{274205E5-AEF1-4D...}
- > Ethernet II, Src: IntelCor\_54:6b:27 (dc:21:48:54:6b:27), Dst: ARRISGro\_d0:30:40 (a8:97:cd:d0:30:40)
- > Internet Protocol Version 4, Src: 192.168.254.90, Dst: 142.250.68.78
- > User Datagram Protocol, Src Port: 54692, Dst Port: 443
- > QUIC IETF

The packet bytes panel shows the raw data for the selected packet, with the destination hardware address (a8 97 cd d0 30 40) highlighted in blue. The corresponding ASCII representation is shown to the right of the hex values.

No.	Time	Source	Destination	Protocol	Length	Info
13890	56.780584	74.125.103.12	192.168.254.90	QUIC	1292	Protected Payload (KP0)
13891	56.781452	74.125.103.12	192.168.254.90	QUIC	1292	Protected Payload (KP0)
13892	56.781452	74.125.103.12	192.168.254.90	QUIC	1292	Protected Payload (KP0)
13893	56.781452	74.125.103.12	192.168.254.90	QUIC	1292	Protected Payload (KP0)
13894	56.781452	74.125.103.12	192.168.254.90	QUIC	1292	Protected Payload (KP0)
13895	56.781452	74.125.103.12	192.168.254.90	QUIC	1292	Protected Payload (KP0)
13896	56.781452	74.125.103.12	192.168.254.90	QUIC	1292	Protected Payload (KP0)
13897	56.781452	74.125.103.12	192.168.254.90	QUIC	1292	Protected Payload (KP0)

Offset	Hex	ASCII
0000	a8 97 cd d0 30 40	...0@.!. HTk'..E.
0010	dc 21 48 54 6b 27 08 00 45 00	..A..@... ..Z..
0020	00 41 b5 9a 40 00 80 11 00 00 c0 a8 fe 5a 8e fa	DN..... ..ESO"4.
0030	44 4e d5 a4 01 bb 00 2d 92 8a 45 53 4f 22 34 10	...{=v.' _m..s...
0040	b0 bb e9 7b 3d 76 ca 27 5f 6d f2 08 73 f8 d0 18	..?...N.. 9.d<.9.
0050	1e 3f c4 af d3 4e 92 c8 39 96 64 3c bc 39 ef	

Destination Hardware Address (eth.dst), 6 bytes

Packets: 14316 · Displayed: 14316 (100.0%) | Profile: Default

# Physical Layer – actual bits on the wire

The image shows a Wireshark packet capture window titled '\*Wi-Fi'. The interface includes a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help) and a toolbar. A packet list table is displayed with the following columns: No., Time, Source, Destination, Protocol, Length, and Info. The first entry is highlighted:

No.	Time	Source	Destination	Protocol	Length	Info
49	9.066881	192.168.254.90	138.197.192.78	HTTP	217	GET / HTTP/1.1

Below the packet list, the packet details pane shows the following information for the selected packet:

- > Frame 49: 217 bytes on wire (1736 bits), 217 bytes captured (1736 bits) on interface \Device\NPF\_{274205E5-AEF1-4D43-...}
- > Ethernet II, Src: IntelCor\_54:6b:27 (dc:21:48:54:6b:27), Dst: ARRISGro\_d0:30:40 (a8:97:cd:d0:30:40)
- > Internet Protocol Version 4, Src: 192.168.254.90, Dst: 138.197.192.78
- > Transmission Control Protocol, Src Port: 49171, Dst Port: 80, Seq: 1, Ack: 1, Len: 163
- > Hypertext Transfer Protocol

The packet bytes pane at the bottom displays the raw data in hexadecimal and ASCII. The first 163 bytes of the packet are shown, corresponding to the HTTP GET request.

```
0000 a8 97 cd d0 30 40 dc 21 48 54 6b 27 08 00 45 b8 ....0@.! HTk'..E.
0010 00 cb a5 19 40 00 80 06 00 00 c0 a8 fe 5a 8a c5 ....@... ..Z..
0020 c0 4e c0 13 00 50 89 1c 12 63 08 cd b1 40 50 18 .N...P.. .c...@P.
0030 04 02 74 2f 00 00 47 45 54 20 2f 20 48 54 54 50 ..t/..GE T / HTTP
0040 2f 31 2e 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74 /1.1..Us er-Agent
0050 3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 57 : Mozill a/5.0 (W
0060 69 6e 64 6f 77 73 20 4e 54 3b 20 57 69 6e 64 6f indows N T; Windo
0070 77 73 20 4e 54 20 31 30 2e 30 3b 20 65 6e 2d 55 ws NT 10 .0; en-U
0080 53 29 20 57 69 6e 64 6f 77 73 50 6f 77 65 72 53 S) Windo wsPowerS
0090 68 65 6c 6c 2f 35 2e 31 2e 32 32 30 30 30 2e 32 hell/5.1 .22000.2
00a0 38 32 0d 0a 48 6f 73 74 3a 20 73 63 72 69 76 6e 82..Host : scrivn
00b0 6f 72 2e 63 69 6b 65 79 73 2e 63 6f 6d 0d 0a 43 or.cikey s.com..C
00c0 6f 6e 6e 65 63 74 69 6f 6e 3a 20 4b 65 65 70 2d onnectio n: Keep-
00d0 41 6c 69 76 65 0d 0a 0d 0a Alive... .
```

The status bar at the bottom indicates: Frame (frame), 217 bytes | Packets: 71 · Displayed: 2 (2.8%) · Dropped: 0 (0.0%) | Profile: Default

# Data Link Layer – wifi in this case

The image shows a Wireshark packet capture window titled '\*Wi-Fi'. The interface includes a menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help) and a toolbar. A display filter bar shows 'Apply a display filter ... <Ctrl-/>'. The packet list pane shows a single packet, No. 49, at time 9.066881, from source 192.168.254.90 to destination 138.197.192.78, protocol HTTP, length 217, with info 'GET / HTTP/1.1'. The packet details pane shows the following structure:

- > Frame 49: 217 bytes on wire (1736 bits), 217 bytes captured (1736 bits) on interface \Device\NPF\_{274205E5-AEF1-4D43-...}
- > Ethernet II, Src: IntelCor\_54:6b:27 (dc:21:48:54:6b:27), Dst: ARRISGro\_d0:30:40 (a8:97:cd:d0:30:40)
  - > Destination: ARRISGro\_d0:30:40 (a8:97:cd:d0:30:40)
  - > Source: IntelCor\_54:6b:27 (dc:21:48:54:6b:27)
  - Type: IPv4 (0x0800)
- > Internet Protocol Version 4, Src: 192.168.254.90, Dst: 138.197.192.78
- > Transmission Control Protocol, Src Port: 49171, Dst Port: 80, Seq: 1, Ack: 1, Len: 163
- > Hypertext Transfer Protocol

The packet bytes pane shows the raw data in hexadecimal and ASCII. The first 14 bytes of the Ethernet frame are highlighted in blue:

Offset	Hex	ASCII
0000	a8 97 cd d0 30 40 dc 21 48 54 6b 27 08 00 45 b8	....0@.!.HTk'..E-
0010	00 cb a5 19 40 00 80 06 00 00 c0 a8 fe 5a 8a c5	....@... ..Z..
0020	c0 4e c0 13 00 50 89 1c 12 63 08 cd b1 40 50 18	..N...P...-c...@P..
0030	04 02 74 2f 00 00 47 45 54 20 2f 20 48 54 54 50	..t/..GE T / HTTP
0040	2f 31 2e 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74	/1.1..Us er-Agent
0050	3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 57	: Mozill a/5.0 (W
0060	69 6e 64 6f 77 73 20 4e 54 3b 20 57 69 6e 64 6f	indows N T; Windo
0070	77 73 20 4e 54 20 31 30 2e 30 3b 20 65 6e 2d 55	ws NT 10 .0; en-U
0080	53 29 20 57 69 6e 64 6f 77 73 50 6f 77 65 72 53	S) Windo wsPowerS
0090	68 65 6c 6c 2f 35 2e 31 2e 32 32 30 30 30 2e 32	hell/5.1 .22000.2
00a0	38 32 0d 0a 48 6f 73 74 3a 20 73 63 72 69 76 6e	82..Host : scrivn
00b0	6f 72 2e 63 69 6b 65 79 73 2e 63 6f 6d 0d 0a 43	or.cikey s.com..C
00c0	6f 6e 6e 65 63 74 69 6f 6e 3a 20 4b 65 65 70 2d	onnectio n: Keep-
00d0	41 6c 69 76 65 0d 0a 0d 0a	Alive... -

The status bar at the bottom shows 'Ethernet (eth), 14 bytes' and 'Packets: 71 · Displayed: 71 (100.0%) · Dropped: 0 (0.0%) | Profile: Default'.

# Network Layer – IPv4

The image shows a Wireshark packet capture window titled '\*Wi-Fi'. The packet list at the top shows a single packet (No. 49) at time 9.066881, from source 192.168.254.90 to destination 138.197.192.78, protocol HTTP, length 217 bytes, with info 'GET / HTTP/1.1'. The packet details pane shows the following structure:

- > Frame 49: 217 bytes on wire (1736 bits), 217 bytes captured (1736 bits) on interface \Device\NPF\_{274205E5-AEF1-4D4}
- > Ethernet II, Src: IntelCor\_54:6b:27 (dc:21:48:54:6b:27), Dst: ARRISGro\_d0:30:40 (a8:97:cd:d0:30:40)
- ✓ Internet Protocol Version 4, Src: 192.168.254.90, Dst: 138.197.192.78
  - 0100 .... = Version: 4
  - .... 0101 = Header Length: 20 bytes (5)
  - > Differentiated Services Field: 0xb8 (DSCP: EF PHB, ECN: Not-ECT)
  - Total Length: 203
  - Identification: 0xa519 (42265)
  - > Flags: 0x40, Don't fragment
  - ...0 0000 0000 0000 = Fragment Offset: 0
  - Time to Live: 128
  - Protocol: TCP (6)
  - Header Checksum: 0x0000 [validation disabled]
  - [Header checksum status: Unverified]

The packet bytes pane shows the raw data in hexadecimal and ASCII. The first 20 bytes (0000 to 00d0) are highlighted in blue, corresponding to the IPv4 header. The ASCII column shows the beginning of the HTTP request: '....@.!. HTk'..E..'. The status bar at the bottom indicates 'Internet Protocol Version 4 (ip), 20 bytes' and 'Packets: 71 · Displayed: 71 (100.0%) · Dropped: 0 (0.0%) | Profile: Default'.

No.	Time	Source	Destination	Protocol	Length	Info
49	9.066881	192.168.254.90	138.197.192.78	HTTP	217	GET / HTTP/1.1

```
> Frame 49: 217 bytes on wire (1736 bits), 217 bytes captured (1736 bits) on interface \Device\NPF_{274205E5-AEF1-4D4}
> Ethernet II, Src: IntelCor_54:6b:27 (dc:21:48:54:6b:27), Dst: ARRISGro_d0:30:40 (a8:97:cd:d0:30:40)
✓ Internet Protocol Version 4, Src: 192.168.254.90, Dst: 138.197.192.78
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 bytes (5)
  > Differentiated Services Field: 0xb8 (DSCP: EF PHB, ECN: Not-ECT)
  Total Length: 203
  Identification: 0xa519 (42265)
  > Flags: 0x40, Don't fragment
  ...0 0000 0000 0000 = Fragment Offset: 0
  Time to Live: 128
  Protocol: TCP (6)
  Header Checksum: 0x0000 [validation disabled]
  [Header checksum status: Unverified]
```

Offset	Hex	ASCII
0000	a8 97 cd d0 30 40 dc 21 48 54 6b 27 08 00 45 b8	....@.!. HTk'..E..
0010	00 cb a5 19 40 00 80 06 00 00 c0 a8 fe 5a 8a c5	....@... ..Z..
0020	c0 4e c0 13 00 50 89 1c 12 63 08 cd b1 40 50 18	.N...P...c...@P..
0030	04 02 74 2f 00 00 47 45 54 20 2f 20 48 54 54 50	..t/..GE T / HTTP
0040	2f 31 2e 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74	/1.1..Us er-Agent
0050	3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 57	: Mozill a/5.0 (W
0060	69 6e 64 6f 77 73 20 4e 54 3b 20 57 69 6e 64 6f	indows N T; Windo
0070	77 73 20 4e 54 20 31 30 2e 30 3b 20 65 6e 2d 55	ws NT 10 .0; en-U
0080	53 29 20 57 69 6e 64 6f 77 73 50 6f 77 65 72 53	S) Windo wsPowerS
0090	68 65 6c 6c 2f 35 2e 31 2e 32 32 30 30 30 2e 32	hell/5.1 .22000.2
00a0	38 32 0d 0a 48 6f 73 74 3a 20 73 63 72 69 76 6e	82..Host : scrivn
00b0	6f 72 2e 63 69 6b 65 79 73 2e 63 6f 6d 0d 0a 43	or.cikey s.com..C
00c0	6f 6e 6e 65 63 74 69 6f 6e 3a 20 4b 65 65 70 2d	onnectio n: Keep-
00d0	41 6c 69 76 65 0d 0a 0d 0a	Alive... .

Internet Protocol Version 4 (ip), 20 bytes | Packets: 71 · Displayed: 71 (100.0%) · Dropped: 0 (0.0%) | Profile: Default

# Transport Layer - TCP

The image shows a Wireshark packet capture window titled '\*Wi-Fi'. The packet list at the top shows a single packet (No. 49) at time 9.066881, from source 192.168.254.90 to destination 138.197.192.78, protocol HTTP, length 217 bytes, with info 'GET / HTTP/1.1'. The packet details pane shows the following structure:

- Frame 49: 217 bytes on wire (1736 bits), 217 bytes captured (1736 bits) on interface \Device\NPF\_{274205E5-AEF1-4D4}
- Ethernet II, Src: IntelCor\_54:6b:27 (dc:21:48:54:6b:27), Dst: ARRISGro\_d0:30:40 (a8:97:cd:d0:30:40)
- Internet Protocol Version 4, Src: 192.168.254.90, Dst: 138.197.192.78
- Transmission Control Protocol, Src Port: 49171, Dst Port: 80, Seq: 1, Ack: 1, Len: 163
  - Source Port: 49171
  - Destination Port: 80
  - [Stream index: 10]
  - [Conversation completeness: Incomplete, DATA (15)]
  - [TCP Segment Len: 163]
  - Sequence Number: 1 (relative sequence number)
  - Sequence Number (raw): 2300318307
  - [Next Sequence Number: 164 (relative sequence number)]
  - Acknowledgment Number: 1 (relative ack number)
  - Acknowledgment number (raw): 147697984

The packet bytes pane shows the raw data in hexadecimal and ASCII. The ASCII column highlights the HTTP request line: 'GET / HTTP/1.1'.

No.	Time	Source	Destination	Protocol	Length	Info
49	9.066881	192.168.254.90	138.197.192.78	HTTP	217	GET / HTTP/1.1

```
> Frame 49: 217 bytes on wire (1736 bits), 217 bytes captured (1736 bits) on interface \Device\NPF_{274205E5-AEF1-4D4}
> Ethernet II, Src: IntelCor_54:6b:27 (dc:21:48:54:6b:27), Dst: ARRISGro_d0:30:40 (a8:97:cd:d0:30:40)
> Internet Protocol Version 4, Src: 192.168.254.90, Dst: 138.197.192.78
> Transmission Control Protocol, Src Port: 49171, Dst Port: 80, Seq: 1, Ack: 1, Len: 163
  Source Port: 49171
  Destination Port: 80
  [Stream index: 10]
  [Conversation completeness: Incomplete, DATA (15)]
  [TCP Segment Len: 163]
  Sequence Number: 1 (relative sequence number)
  Sequence Number (raw): 2300318307
  [Next Sequence Number: 164 (relative sequence number)]
  Acknowledgment Number: 1 (relative ack number)
  Acknowledgment number (raw): 147697984
```

Offset	Hex	ASCII
0000	a8 97 cd d0 30 40 dc 21 48 54 6b 27 08 00 45 b8	....@.!. HTk'..E.
0010	00 cb a5 19 40 00 80 06 00 00 c0 a8 fe 5a 8a c5	....@... ..Z..
0020	c0 4e c0 13 00 50 89 1c 12 63 08 cd b1 40 50 18	.N...P...c...@P.
0030	04 02 74 2f 00 00 47 45 54 20 2f 20 48 54 54 50	..t/..GE T / HTTP
0040	2f 31 2e 31 0d 0a 55 73 65 72 2d 41 67 65 6e 74	/1.1..Us er-Agent
0050	3a 20 4d 6f 7a 69 6c 6c 61 2f 35 2e 30 20 28 57	: Mozill a/5.0 (W
0060	69 6e 64 6f 77 73 20 4e 54 3b 20 57 69 6e 64 6f	indows N T; Windo
0070	77 73 20 4e 54 20 31 30 2e 30 3b 20 65 6e 2d 55	ws NT 10 .0; en-U
0080	53 29 20 57 69 6e 64 6f 77 73 50 6f 77 65 72 53	S) Windo wsPowerS
0090	68 65 6c 6c 2f 35 2e 31 2e 32 32 30 30 30 2e 32	hell/5.1 .22000.2
00a0	38 32 0d 0a 48 6f 73 74 3a 20 73 63 72 69 76 6e	82..Host : scrivn
00b0	6f 72 2e 63 69 6b 65 79 73 2e 63 6f 6d 0d 0a 43	or.cikey s.com..C
00c0	6f 6e 6e 65 63 74 69 6f 6e 3a 20 4b 65 65 70 2d	onnectio n: Keep-
00d0	41 6c 69 76 65 0d 0a 0d 0a	Alive... .

Transmission Control Protocol (tcp), 20 bytes | Packets: 71 · Displayed: 71 (100.0%) · Dropped: 0 (0.0%) | Profile: Default

# Application Layer – HTTP (client request for file)

The image shows a Wireshark packet capture window titled '\*Wi-Fi'. The packet list pane at the top shows a single packet, No. 49, at time 9.066881, from source 192.168.254.90 to destination 138.197.192.78, protocol HTTP, length 217, and info 'GET / HTTP/1.1'. The packet details pane shows the following structure:

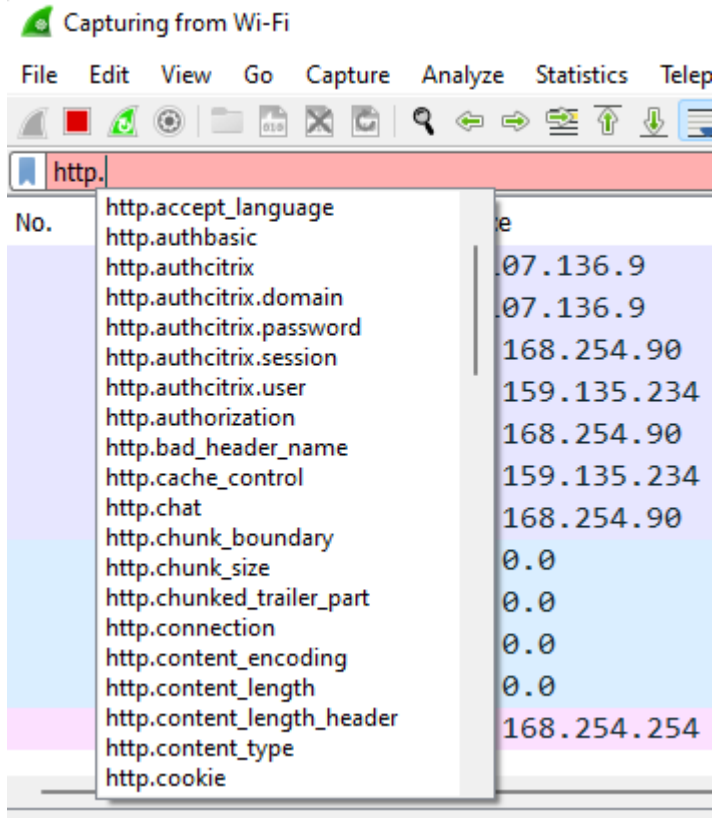
- > Frame 49: 217 bytes on wire (1736 bits), 217 bytes captured (1736 bits) on interface \Device\NPF\_{274205E5-AEF1-4D43-...}
- > Ethernet II, Src: IntelCor\_54:6b:27 (dc:21:48:54:6b:27), Dst: ARRISGro\_d0:30:40 (a8:97:cd:d0:30:40)
- > Internet Protocol Version 4, Src: 192.168.254.90, Dst: 138.197.192.78
- > Transmission Control Protocol, Src Port: 49171, Dst Port: 80, Seq: 1, Ack: 1, Len: 163
- ▼ Hypertext Transfer Protocol
  - > GET / HTTP/1.1\r\n
    - User-Agent: Mozilla/5.0 (Windows NT; Windows NT 10.0; en-US) WindowsPowerShell/5.1.22000.282\r\n
    - Host: scrivnor.cikeys.com\r\n
    - Connection: Keep-Alive\r\n
    - \r\n
    - [Full request URI: <http://scrivnor.cikeys.com/>]
    - [HTTP request 1/1]
    - [Response in frame: 52]

The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII. The ASCII column shows the request line and headers, with some characters highlighted in blue:

```
....@.!. HTk'..E-
....@... ..Z..
-N...P... -c...@P-
..t/..GE T / HTTP
/1.1..Us er-Agent
: Mozill a/5.0 (W
indows N T; Windo
ws NT 10 .0; en-U
S) Windo wsPowerS
hell/5.1 .22000.2
82..Host : scrivn
or.cikey s.com..C
onnectio n: Keep-
Alive... .
```

The status bar at the bottom indicates: Hypertext Transfer Protocol (http), 163 bytes | Packets: 71 · Displayed: 71 (100.0%) · Dropped: 0 (0.0%) | Profile: Default

# Filtering



- Filtering is a very powerful and useful feature in Wireshark.
- You filter by protocol, say http
  - Valid filter: http
  - Result: shows you all the packets related to http
- You filter more specifically
  - Valid filter: http.request.method == "GET"
  - Result: shows you all the GET requests made.
- You have to know what the protocol headers are, so that you can filter down.
  - IDEA: type "http." into the filter bar and look at the results. You can drill down until you find what you need.

Practice



# Download and Analyze TFTP Wireshark Capture

- Wireshark Capture: <http://429.scrivnor.cikeys.com/captures/week03-tftp.pcapng>
1. How many unique message types do we see in this transaction?
  2. What is the default block size for TFTP?
  3. What is the block size used in this transaction?
  4. Which frame numbers contain read requests, if any?
  5. Which frame numbers contain write requests, if any?
  6. Is a file being transferred from the server or being sent to the server?  
How do you know?
  7. Given your answer to number 1, which IP address represents the server?
  8. How does the client/server know the file transfer is complete?