

Analyze the network Packet

Wireshark : s an open-source network protocol analyzer. It uses a graphical user interface (GUI), which makes it easier to visualize network communications for packet analysis purposes. Wireshark has many features to explore that are beyond the scope of this course. You'll focus on how to use basic filtering to isolate network packets so that you can find what you need.

Display filters

Wireshark's display filters let you apply filters to packet capture files. This is helpful when you are inspecting packet captures with large volumes of information. Display filters will help you find specific information that's most relevant to your investigation. You can filter packets based on information such as protocols, IP addresses, ports, and virtually any other property found in a packet. Here, you'll focus on display filtering syntax and filtering for protocols, IP addresses, and ports.

Comparison operators

You can use different comparison operators to locate specific header fields and values. Comparison operators can be expressed using either abbreviations or symbols. For example, this filter using the == equal symbol in this filter `ip.src == 8.8.8.8` is identical to using the `eq` abbreviation in this filter `ip.src eq 8.8.8.8`.

This table summarizes the different types of comparison operators you can use for display filtering.

Operator type	Symbol	Abbreviation
Equal	<code>==</code>	<code>eq</code>
Not equal	<code>!=</code>	<code>ne</code>
Greater than	<code>></code>	<code>gt</code>
Less than	<code><</code>	<code>lt</code>
Greater than or equal to	<code>>=</code>	<code>ge</code>
Less than or equal to	<code><=</code>	<code>le</code>

Contains operator

The contains operator is used to filter packets that contain an exact match of a string of text. Here is an example of a filter that displays all HTTP streams that match the keyword "moved".

*** sample.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http contains "moved"

No.	Time	Source	Destination	Protocol	Length	Info
69	18.036927	142.250.1.139	172.21.224.2	HTTP	648	HTTP/1.1 301 Moved Permanently (text/html)
150	42.370267	142.250.1.102	172.21.224.2	HTTP	657	HTTP/1.1 301 Moved Permanently (text/html)

Matches operator

The matches operator is used to filter packets based on the regular expression (regex) that's specified. Regular expression is a sequence of characters that forms a pattern. You'll explore more about regular expressions later in this program.

Filter toolbar

You can apply filters to a packet capture using Wireshark's filter toolbar. In this example, dns is the applied filter, which means Wireshark will only display packets containing the DNS protocol.

*** sample.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

dns

No.	Time	Source	Destination	Protocol	Length	Info
9	8.637619	172.21.224.2	169.254.169.254	DNS	81	Standard query 0x0c26 A op
10	8.637625	172.21.224.2	169.254.169.254	DNS	81	Standard query 0xd638 AAAA
11	8.641838	169.254.169.254	172.21.224.2	DNS	193	Standard query response 0x
12	8.641978	169.254.169.254	172.21.224.2	DNS	177	Standard query response 0x
19	8.644093	172.21.224.2	169.254.169.254	DNS	86	Standard query 0xb549 PTR
20	8.647339	169.254.169.254	172.21.224.2	DNS	120	Standard query response 0x
27	9.645214	172.21.224.2	169.254.169.254	DNS	86	Standard query 0x3cdc PTR
28	9.645859	169.254.169.254	172.21.224.2	DNS	120	Standard query response 0x
33	10.646715	172.21.224.2	169.254.169.254	DNS	86	Standard query 0x94d7 PTR

Filter toolbar

Pro tip: Wireshark uses different colors to represent protocols. You can customize colors and create your own filters.

Filter for protocols

Protocol filtering is one of the simplest ways you can use display filters. You can simply enter the name of the protocol to filter. For example, to filter for DNS packets simply type dns in the filter toolbar. Here is a list of some protocols you can filter for:

- dns
- http
- ftp
- ssh
- arp
- telnet
- icmp

Filter for an IP address

You can use display filters to locate packets with a specific IP address.

For example, if you would like to filter packets that contain a specific IP address use `ip.addr`, followed by a space, the equal `==` comparison operator, and the IP address. Here is an example of a display filter that filters for the IP address 172.21.224.2:

```
ip.addr == 172.21.224.2
```

To filter for packets originating from a specific source IP address, you can use the `ip.src` filter. Here is an example that looks for the 10.10.10.10 source IP address:

```
ip.src == 10.10.10.10
```

To filter for packets delivered to a specific destination IP address, you can use the `ip.dst` filter. Here is an example that searches for the 4.4.4.4 destination IP address:

```
ip.dst == 4.4.4.4
```

Filter for a MAC address

You can also filter packets according to the Media Access Control (MAC) address. As a refresher, a MAC address is a unique alphanumeric identifier that is assigned to each physical device on a network.

Here's an example:

```
eth.addr == 00:70:f4:23:18:c4
```

Filter for ports

Port filtering is used to filter packets based on port numbers. This is helpful when you want to isolate specific types of traffic. DNS traffic uses TCP or UDP port 53 so this will list traffic related to DNS queries and responses only.

For example, if you would like to filter for a UDP port:

```
udp.port == 53
```

Likewise, you can filter for TCP ports as well:

```
tcp.port == 25
```

Now , we are going to analyze the packet using Wireshark for small.pcap file

- 1) Double-click on small.pcap to open it with wireshark as wireshark has been already installed

Columns available:

No. : The index number of the packet in this packet capture file

Time: The timestamp of the packet

Source: The source IP address

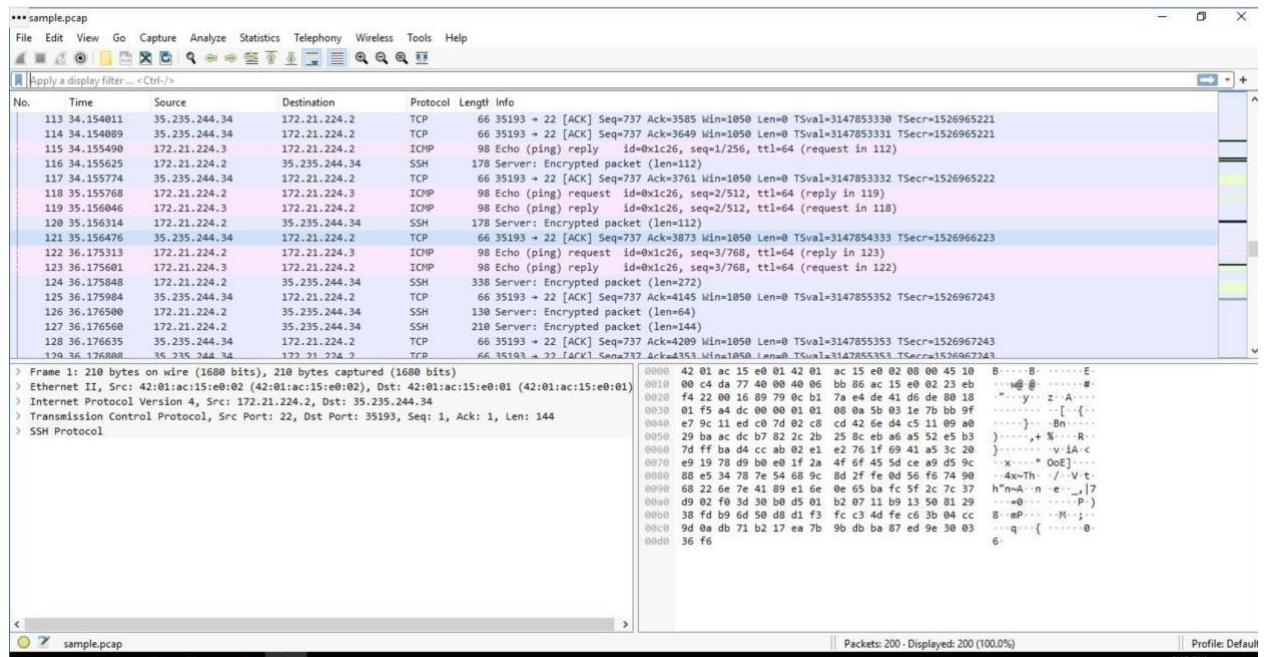
Destination: The destination IP address

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

- 1) As soon as we open the file , the Wireshark page looks in the below manner



It displays all the data based on colored format which will enable us to differentiate based on the protocols. we can also apply filters to fetch the data easily. On every packet when we check

for subsets Eg: frame , ethernet, Internet protocol, Transmission control protocol/ UDP protocol, it will provide us with all the details to detect the deviations.

Below are some filters which we use on a regular basis.

Filtering Based on IP Address:

***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=65536 Len=0 MSS=1420 SACK_PERM TSval=2804123005 TSecr=0 WS=128
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
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=2804123009 TSecr=4069674934
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=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
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

> Frame 16: 98 bytes on wire (784 bits), 98 bytes captured (784 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
> Internet Control Message Protocol

sample.pcap

Packets: 200 · Displayed: 16 (8.0%)

Profile: Default

***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=65536 Len=0 MSS=1420 SACK_PERM TSval=2804123005 TSecr=0 WS=128
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
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=2804123009 TSecr=4069674934
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=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
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

> Frame 65: 74 bytes on wire (592 bits), 74 bytes captured (592 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: 142.250.1.139, Dst: 172.21.224.2
▼ Transmission Control Protocol, Src Port: 80, Dst Port: 49652, Seq: 0, Ack: 1, Len: 0
Source Port: 80
Destination Port: 49652
[Stream index: 4]
> [Conversation completeness: Complete, WITH_DATA (31)]
> [TCP Segment Len: 0]
Sequence Number: 0 (relative sequence number)
Sequence Number (raw): 1617226787
[Next Sequence Number: 1 (relative sequence number)]
Acknowledgment Number: 1 (relative ack number)
Acknowledgment number (raw): 3412824993
1010 ... = Header Length: 40 bytes (10)
> Flags: 0x012 (SYN, ACK)
Window: 65535

sample.pcap

Packets: 200 · Displayed: 16 (8.0%)

Profile: Default

Filters to search based on MAC address:

***sample.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ethaddr == 42:01:ac:15:e0:02

No.	Time	Source	Destination	Protocol	Length	Info
8	8.634316	35.235.244.34	172.21.224.2	TCP	66	35193 → 22 [ACK] Seq=161 Ack=369 Win=1050 Len=0 TSval=3147827811 TSecr=1526939701
9	8.637619	172.21.224.2	169.254.169.254	DNS	81	Standard query 0x638 AAAA opensource.google.com
10	8.637625	172.21.224.2	169.254.169.254	DNS	81	Standard query 0x638 AAAA opensource.google.com
11	8.641838	169.254.169.254	172.21.224.2	DNS	193	Standard query response 0x638 AAAA opensource.google.com AAAA 2607:f8b0:4001:c24::8a AAAA 2607:f8b0:4001:c24::64 AAAA
12	8.641978	169.254.169.254	172.21.224.2	DNS	177	Standard query response 0x638 A opensource.google.com A 142.250.1.139 A 142.250.1.138 A 142.250.1.102 A 142.250.1.111
13	8.642416	172.21.224.2	35.235.244.34	SSH	194	Server: Encrypted packet (len=128)
14	8.642560	172.21.224.2	35.235.244.34	SSH	130	Server: Encrypted packet (len=64)
15	8.642598	35.235.244.34	172.21.224.2	TCP	66	35193 → 22 [ACK] Seq=161 Ack=497 Win=1050 Len=0 TSval=3147827819 TSecr=1526939709
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)
17	8.642755	35.235.244.34	172.21.224.2	TCP	66	35193 → 22 [ACK] Seq=161 Ack=561 Win=1050 Len=0 TSval=3147827819 TSecr=1526939709
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)
19	8.644093	172.21.224.2	169.254.169.254	DNS	86	Standard query 0xb549 PTR 139.1.250.142.in-addr.arpa
20	8.647339	169.254.169.254	172.21.224.2	DNS	120	Standard query response 0xb549 PTR 139.1.250.142.in-addr.arpa PTR ju-in-f139.1e100.net
21	8.647514	172.21.224.2	35.235.244.34	SSH	210	Server: Encrypted packet (len=144)
22	8.647587	172.21.224.2	35.235.244.34	SSH	130	Server: Encrypted packet (len=64)
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=1526939714
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=1526939714

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

Internet Control Message Protocol

Type: 8 (Echo (ping) request)

Code: 0

Checksum: 0x42fe [correct]

[Checksum Status: Good]

Identifier (BE): 26673 (0x6831)

Identifier (LE): 12648 (0x3168)

Sequence Number (BE): 1 (0x0001)

Sequence Number (LE): 256 (0x0100)

[Response frame: 18]

Timestamp from icmp data: Nov 23, 2022 12:38:25.230603000 Greenwich Standard Time

[Timestamp from icmp data (relative): 0.000012000 seconds]

Data (40 bytes)

0000 42 01 ac 15 e0 01 42 01 ac 15 e0 02 06 00 45 00 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:hl...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

Ethernet (eth), 14 bytes

Packets: 200 · Displayed: 200 (100.0%)

Profile: Default

Filters to search for DNS protocol

***sample.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

udp.port == 53

No.	Time	Source	Destination	Protocol	Length	Info
9	8.637619	172.21.224.2	169.254.169.254	DNS	81	Standard query 0xc26 A opensource.google.com
10	8.637625	172.21.224.2	169.254.169.254	DNS	81	Standard query 0xd638 AAAA opensource.google.com
11	8.641838	169.254.169.254	172.21.224.2	DNS	193	Standard query response 0xd638 AAAA opensource.google.com AAAA 2607:f8b0:4001:c24::8a AAAA 2607:f8b0:4001:c24::64 AAAA
12	8.641978	169.254.169.254	172.21.224.2	DNS	177	Standard query response 0xc26 A opensource.google.com A 142.250.1.139 A 142.250.1.138 A 142.250.1.102 A 142.250.1.111
19	8.644093	172.21.224.2	169.254.169.254	DNS	86	Standard query 0xb549 PTR 139.1.250.142.in-addr.arpa
20	8.647339	169.254.169.254	172.21.224.2	DNS	120	Standard query response 0xb549 PTR 139.1.250.142.in-addr.arpa PTR ju-in-f139.1e100.net
27	9.645214	172.21.224.2	169.254.169.254	DNS	86	Standard query 0x3cdc PTR 139.1.250.142.in-addr.arpa
29	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 ju-in-f139.1e100.net
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 ju-in-f139.1e100.net
60	18.031311	172.21.224.2	169.254.169.254	DNS	81	Standard query 0xae2b A opensource.google.com
61	18.031318	172.21.224.2	169.254.169.254	DNS	81	Standard query 0x0728 AAAA opensource.google.com
62	18.032377	169.254.169.254	172.21.224.2	DNS	193	Standard query response 0x0728 AAAA opensource.google.com AAAA 2607:f8b0:4001:c24::8a AAAA 2607:f8b0:4001:c24::64 AAAA
63	18.032379	169.254.169.254	172.21.224.2	DNS	177	Standard query response 0xae2b A opensource.google.com A 142.250.1.139 A 142.250.1.138 A 142.250.1.102 A 142.250.1.111
141	42.362339	172.21.224.2	169.254.169.254	DNS	81	Standard query 0xeb67 A opensource.google.com
142	42.362347	172.21.224.2	169.254.169.254	DNS	81	Standard query 0xd6d2 AAAA opensource.google.com
143	42.367155	169.254.169.254	172.21.224.2	DNS	177	Standard query response 0xeb67 A opensource.google.com A 142.250.1.102 A 142.250.1.100 A 142.250.1.101 A 142.250.1.113

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)

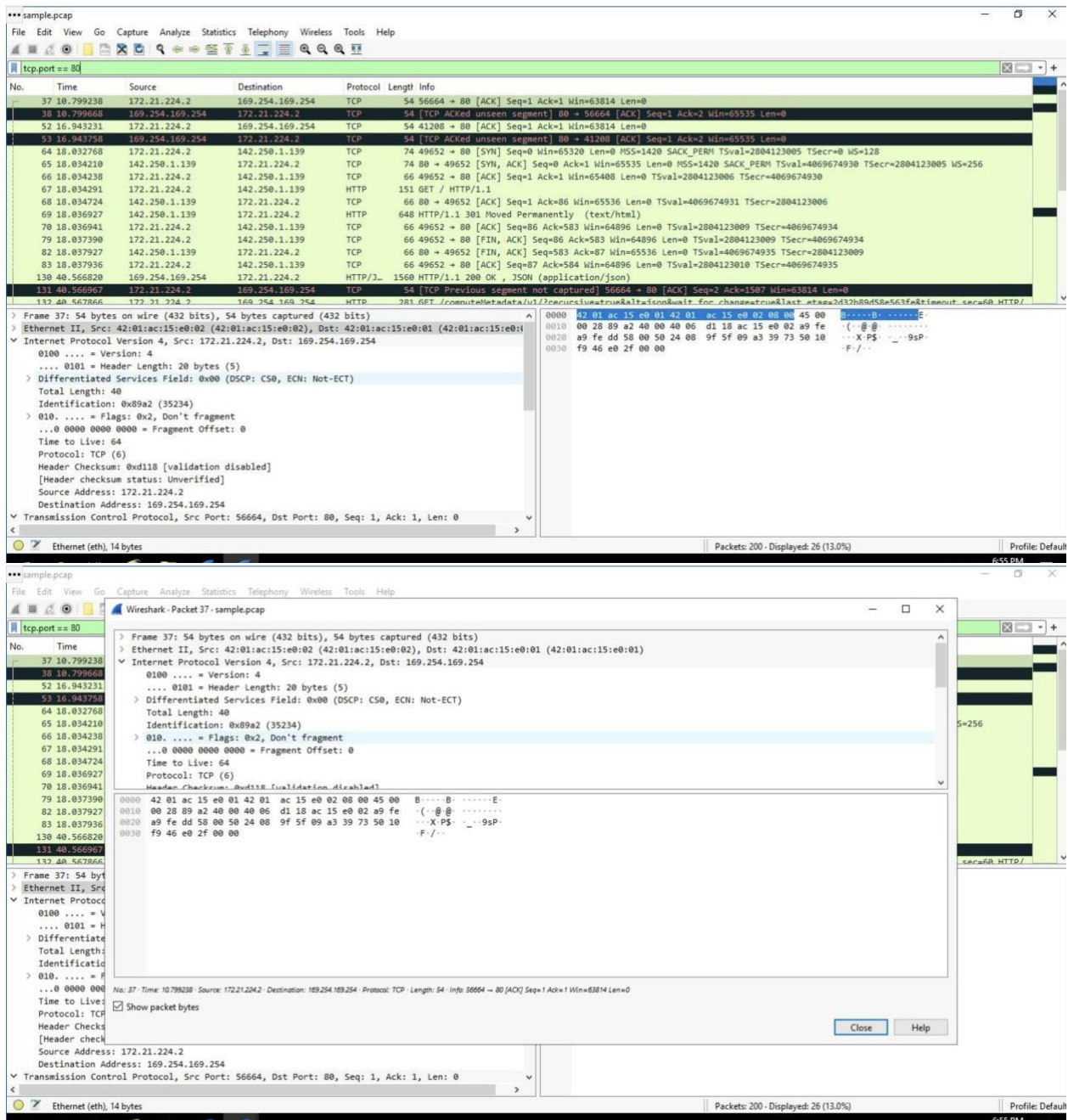
0000 42 01 ac 15 e0 02 42 01 ac 15 e0 01 00 00 45 00 B:.....E-
0010 00 a3 00 00 00 00 40 11 9a 35 a9 fe a9 fe ac 15@:..S.....
0020 e0 02 00 35 e0 06 00 8f b3 b3 0c 26 81 00 00 01 ...5.....&.....
0030 00 06 00 00 00 00 00 00 6f 70 65 6e 73 6f 72 63o pensourc
0040 65 06 67 6f 67 67 6c 65 03 63 6f 6d 00 00 01 00 e google .com...
0050 01 c0 0c 00 01 00 01 00 00 01 2c 00 04 8e fa 01
0060 8b c0 0c 00 01 00 01 00 00 01 2c 00 04 8e fa 01
0070 8a c0 0c 00 01 00 01 00 00 01 2c 00 04 8e fa 01
0080 66 c0 0c 00 01 00 01 00 00 01 2c 00 04 8e fa 01
0090 71 c0 0c 00 01 00 01 00 00 01 2c 00 04 8e fa 01
00a0 64 c0 0c 00 01 00 01 00 00 01 2c 00 04 8e fa 01 d.....
00b0 65 e

Ethernet (eth), 14 bytes

Packets: 200 · Displayed: 22 (11.0%)

Profile: Default

Filter to search based on TCP protocol for HTTP requests:



Wireshark - Packet 67 - sample.pcap

File Edit View Go Capture Analyze Statistics Telephony Window Help

tcp contains "curl"

No.	Time	Source	Destination
67	18.034291	172.21.224.2	142.250.1.139
148	42.369093	172.21.224.2	142.250.1.102

> Frame 67: 151 bytes on wire (1208 bits), 151 bytes captured on interface 0, captured on 172.21.224.2, destination 142.250.1.139

> Ethernet II, Src: 42:01:ac:15:e0:02 (42:01:ac:15:e0:02), Dst: 08:00:00:00:00:00 (08:00:00:00:00:00)

> 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: 137

> Identification: 0xe4aa (58538)

> 010. = Flags: 0x2, Don't fragment

> ...0 0000 0000 0000 = Fragment Offset: 0

> Time to Live: 64

> Protocol: TCP (6)

> Header Checksum: 0x3927 [validation disabled]

> [Header checksum status: Unverified]

> Source Address: 172.21.224.2

> Destination Address: 142.250.1.139

> Transmission Control Protocol, Src Port: 49652, Dst Port: 80, Seq: 1, Ack: 1, Len: 85

> Flags: 0x018 (PSH, ACK)

> Window: 511

> [Calculated window size: 65408]

> [Window size scaling factor: 128]

> Checksum: 0x1d19 [unverified]

> [Checksum Status: Unverified]

> Urgent Pointer: 0

> Options: (12 bytes), No-Operation (NOP), No-Operation (NOP), Timestamps

> [Timestamps]

> [SEQ/ACK analysis]

> TCP payload (85 bytes)

> Hypertext Transfer Protocol

0000 42 01 ac 15 e0 01 42 01 ac 15 e0 02 00 00 45 00 8:---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 6b 93 a1 60 64 ec 24 80 18 ---P:k---"d\$--
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 0 GET / HTTP/1.1
0050 0d 0a 40 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 */*-----

No: 67 - Time: 18.034291 - Source: 172.21.224.2 - Destination: 142.250.1.139 - Protocol: HTTP - Length: 151 - Info: GET / HTTP/1.1

☒ Show packet bytes

Ethernet (eth), 14 bytes

Packets: 200 - Displayed: 2 (1.0%)

Profile: Default

6:57 PM