

Wireshark 101

®

Essential Skills for Network Analysis

Course Contents

- [Enter the topics you wish to cover.]

WARNING

Before you capture your first packet, **ensure you have permission to listen to the network traffic**. If you are an IT staff member, obtain written permission to listen in to network traffic for troubleshooting, optimization, security, and application analysis.

Consult a legal specialist to understand your local and national laws regarding packet capture on wired or wireless networks.

Fundamentals – Why Wireshark?

- **Wireshark Capabilities**

- General traffic analysis
- Troubleshooting
- Security
- Application analysis

- **Supported OSes**

- Windows
- *NIX
- MAC

- Determine **who** is talking in the trace file
- Determine **which applications** are in use
- Filter on the **conversation** of interest
- Graph the **IO rate** to look for drops in throughput
- Open the **Expert** to look for problems
- Determine the **round trip time** to identify path latency

Key Graphical Interface Elements

The image shows the Wireshark network protocol analyzer interface. Eight numbered callouts highlight specific features:

- 1**: Title bar (tr-winsize.pcapng)
- 2**: Menu bar (File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, Help)
- 3**: Toolbar (icons for various functions)
- 4**: Filter bar (Apply a display filter ... <Ctrl-/>)
- 5**: Packet list pane (table of captured packets)
- 6**: Packet details pane (hierarchical view of packet structure)
- 7**: Packet bytes pane (hexadecimal and ASCII representation of packet data)
- 8**: Status bar (Source Port (tcp.srcport), 2 bytes; Packets: 9354 · Displayed: 9354 (100.0%) · Load time: 0:0.183; Profile: Default)

No.	Time	Source	Destination	Protocol	Length	Info
6591	16.943030	10.0.52.164	204.152.184.134	TCP	54	[TCP Window Update] 2646 → 80 [AC
6592	16.943244	204.152.184.134	10.0.52.164	TCP	1514	80 → 2646 [ACK] Seq=4636960 Ack=4
6593	16.943528	10.0.52.164	204.152.184.134	TCP	54	2646 → 80 [ACK] Seq=444 Ack=46384
6594	16.944303	204.152.184.134	10.0.52.164	TCP	1514	80 → 2646 [ACK] Seq=4638420 Ack=4
6595	16.944436	10.0.52.164	204.152.184.134	TCP	54	2646 → 80 [ACK] Seq=444 Ack=46398
6596	16.945186	204.152.184.134	10.0.52.164	TCP	1230	80 → 2646 [PSH, ACK] Seq=4639880

Frame 6592: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interfac...

Ethernet II, Src: 3ComCorp_c9:51:b6 (00:04:75:c9:51:b6), Dst: SonyCorp_f4:3a:09 (08:00:46...

Internet Protocol Version 4, Src: 204.152.184.134, Dst: 10.0.52.164

Transmission Control Protocol, Src Port: 80 (80), Dst Port: 2646 (2646), Seq: 4636960, Ac...

Source Port: 80

Destination Port: 2646

Offset	Hex	ASCII
0020	34 a4 00 50 0a 56 48 e7 b8 75 b8 08 dc ba 50 10	4..P.VH. .u....P.
0030	ff ff cc 1c 00 00 e5 48 57 93 63 36 04 7a d6 78	H W.c6.z.x
0040	68 d0 d5 ac a6 e2 a0 be 35 1d 79 48 50 ad 00 ed	h..... 5.yHP...
0050	15 f3 b0 c0 0f c4 5b ec 32 ba 1d ef 29 4d dd c2	[. 2...)M..

Source Port (tcp.srcport), 2 bytes

Packets: 9354 · Displayed: 9354 (100.0%) · Load time: 0:0.183

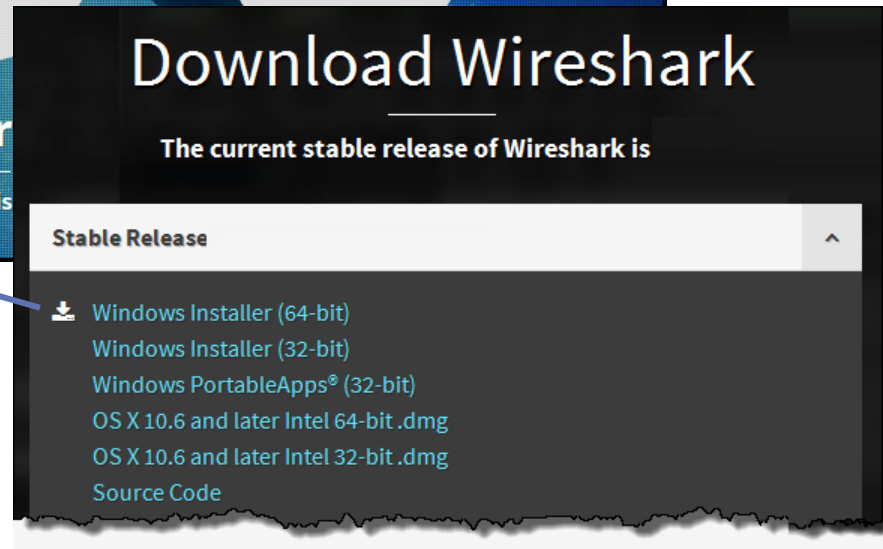
Profile: Default

wireshark.org/download.html

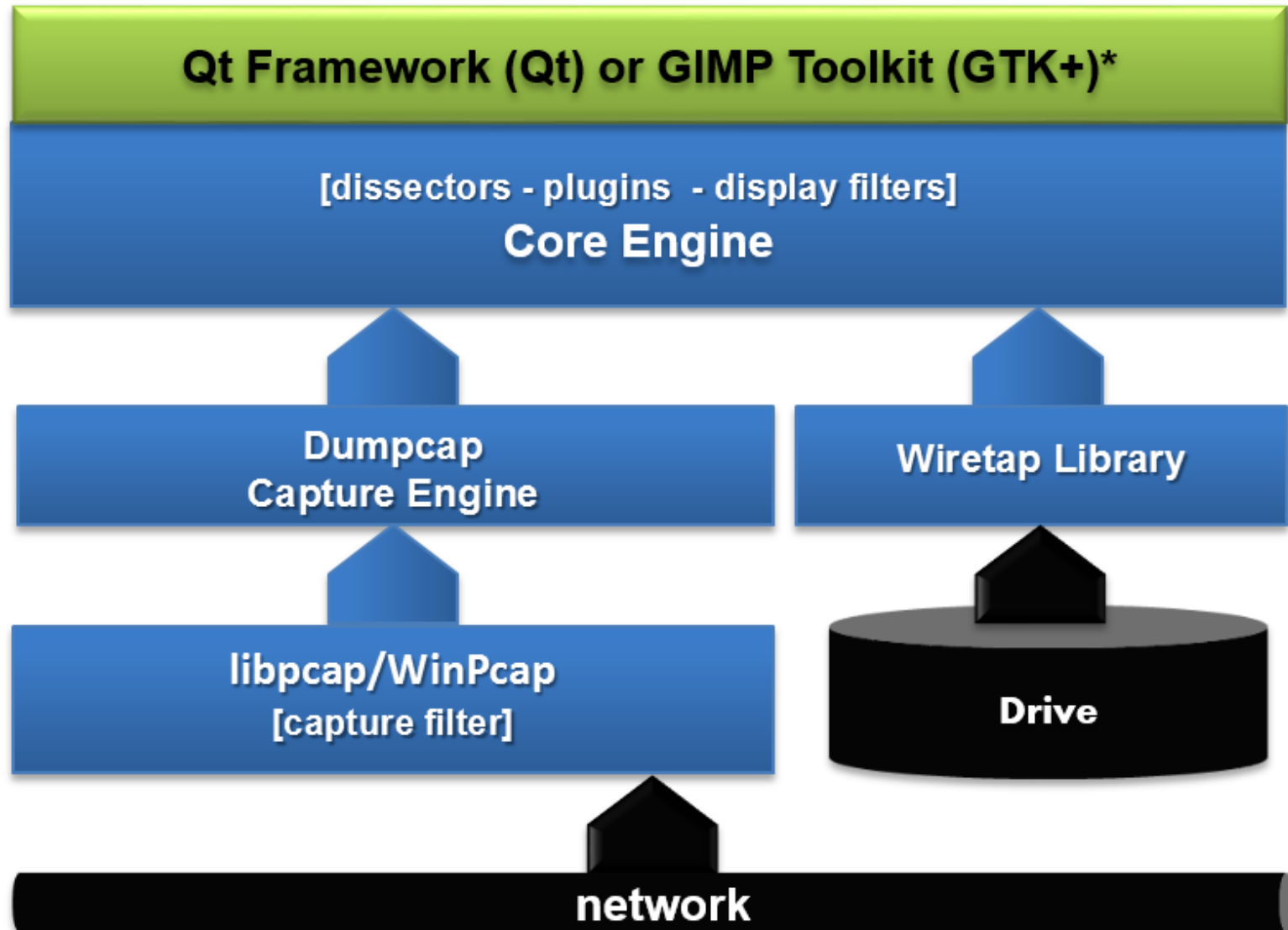


NOTE

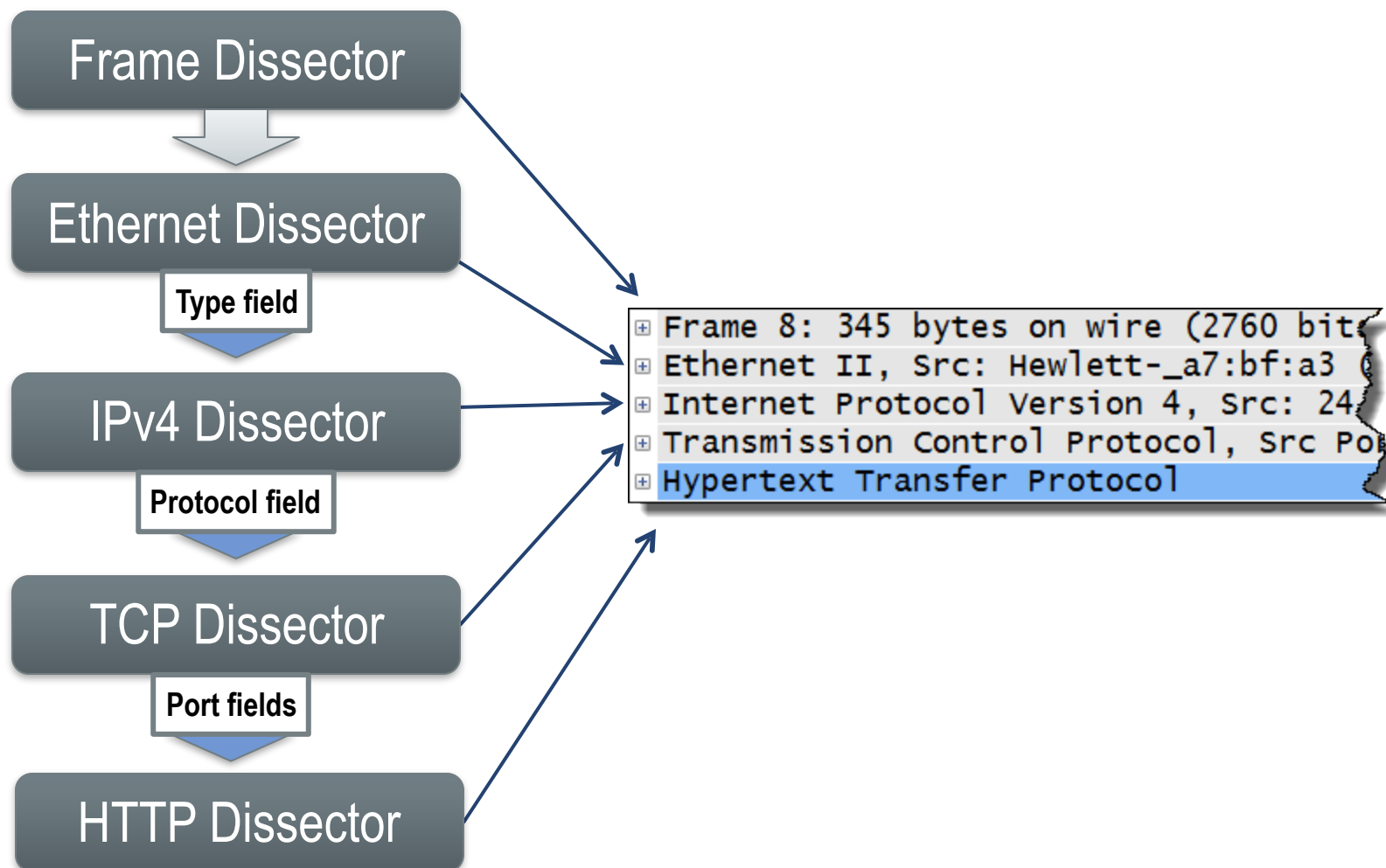
The download.html page suggests the version that matches your incoming HTTP GET request



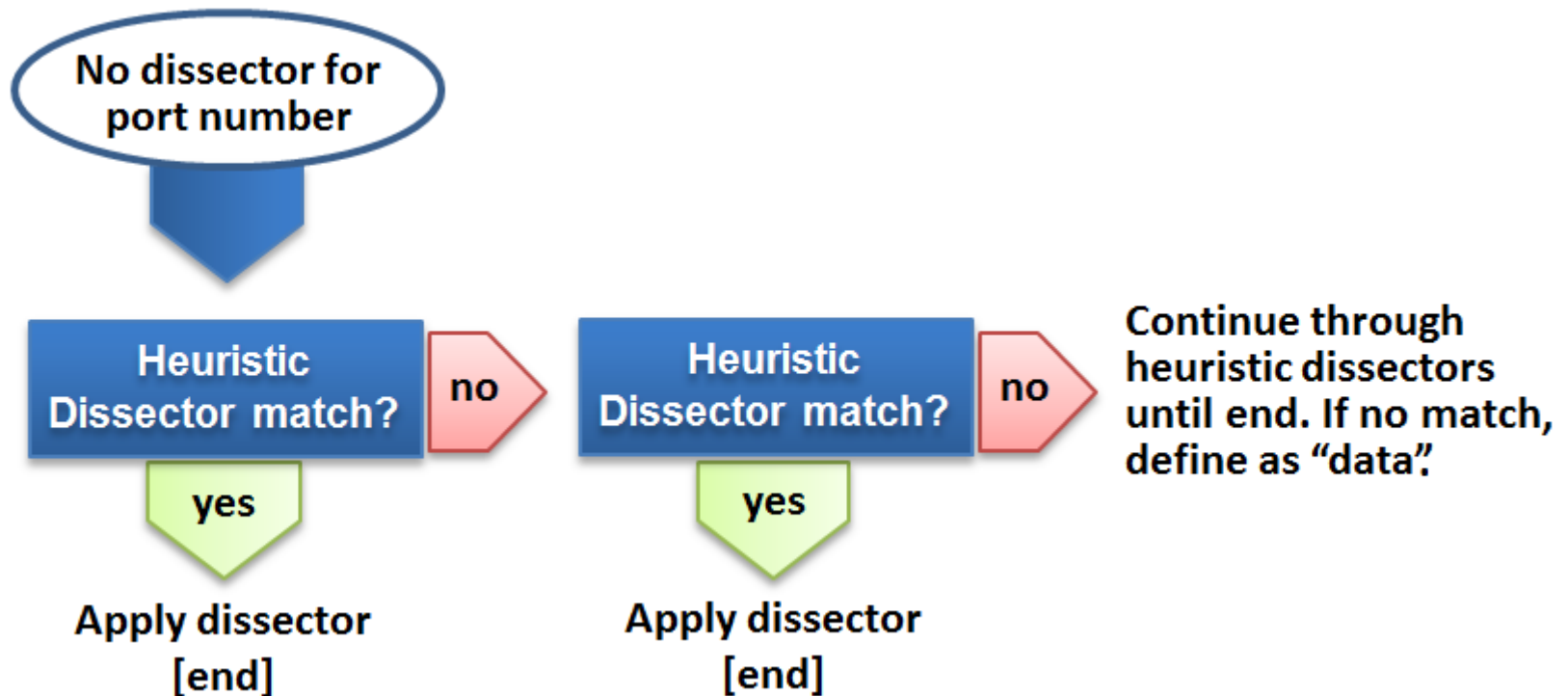
Wireshark Capture Elements



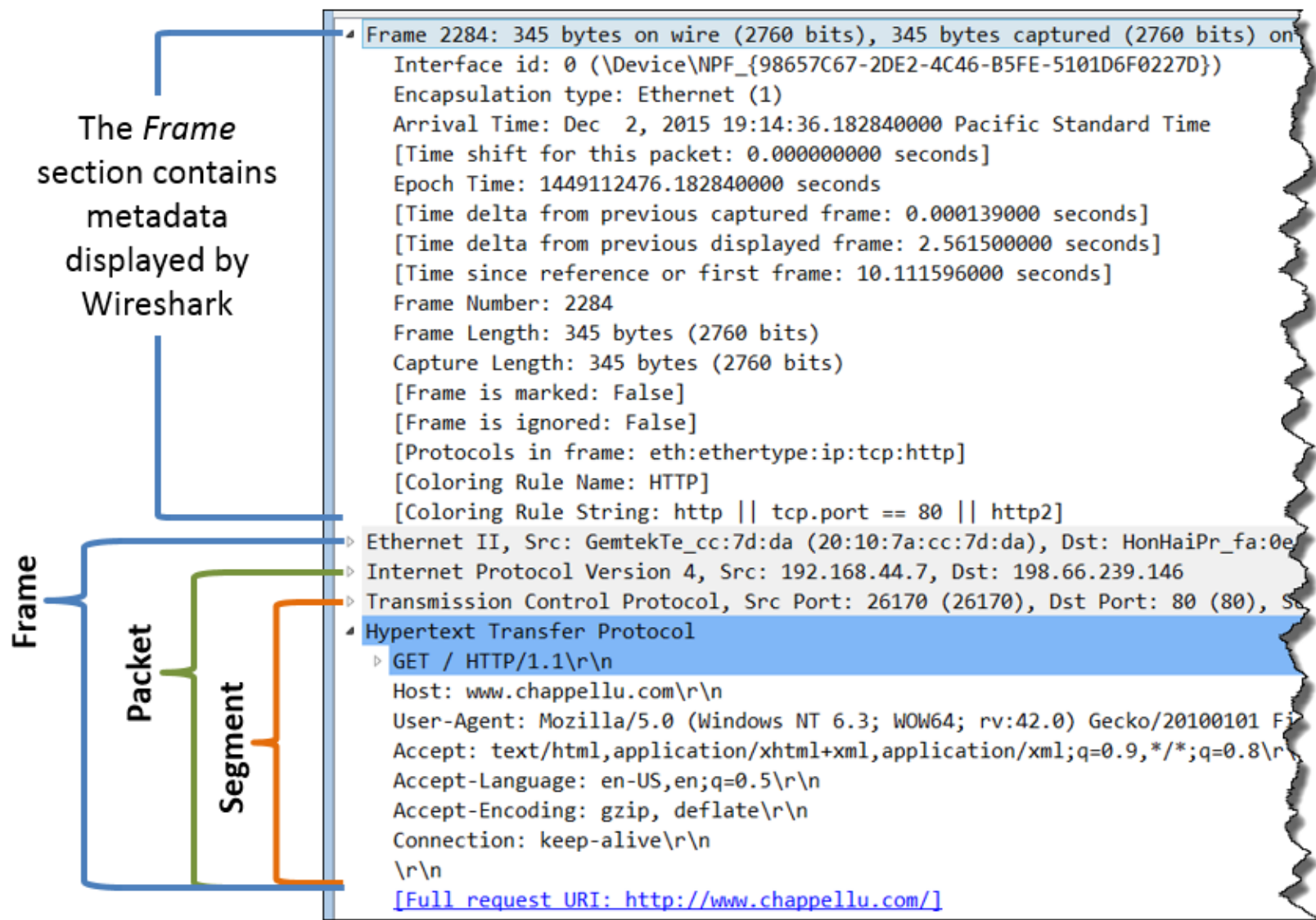
Dissect the Wireshark Dissectors



How Heuristic Dissectors Work



Frames vs. Packets vs. Segments



Wireshark Resource: Q & A Forum



[login](#) [about](#) [faq](#)

1 Questions

2 Tags

3 Users

4 Badges

5 Unanswered

6 Ask a Question

Search

☒ Questions ☐ Tags ☐ Users

8 Questions

9 Answers

10

0 votes

1 answer

63 views

Decrypting a Wireshark capture without the private key

ssl ssl_decrypt

12 active

newest

most voted

13 27 mins ago Kurt Knochner ♦ 24.0k

0 votes

1 answer

16 views

HTTP response not being decoded

http

11

1 hour ago Christian_R 1.0k

0 votes

2 answers

37 views

Losing connection with weird behavior (disconnecting everyone)

ack disconnect

1 hour ago mmguy 6

0 votes

1 answer

27 views

Exporting Decrypted 802.11 WPA/PSK Packets in PCAP Format

wpa-psk export pcap wifi 802.11

4 hours ago Amato_C 742

0 votes

1 answer

37 views

Wireshark /.config permission problems under OS X 10.11.1 (El Capitan)

config permissions

5 hours ago Christian_R 1.0k

0 votes

0 answers

21 views

Wireshark 2.0 for linux nonroot users

linux

Welcome to Wireshark Q&A

Ask and answer questions about Wireshark, protocols, and Wireshark development

[about](#)
[faq](#)

10184 Questions

11346 answers

questions

You have a trillion packets.

You need to see four of them.

[Riverbed Technology](#) lets you seamlessly move between packets and flows for comprehensive monitoring, analysis and

The Default Three-Pane View

The image displays the Wireshark network protocol analyzer interface, showing the default three-pane view for a packet capture file named `http-google101.pcapng`.

Packet List Pane: This pane shows a list of captured packets. The selected packet is packet 1, which is a DNS Standard query from 24.6.173.220 to 75.75.75.75. Other packets in the list include a DNS Standard query response, a TCP SYN packet, and a TCP ACK packet.

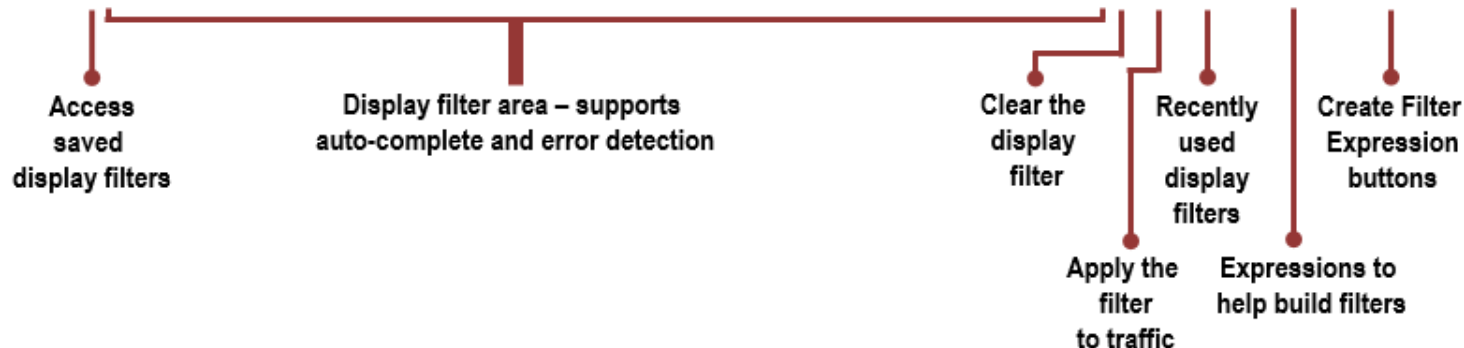
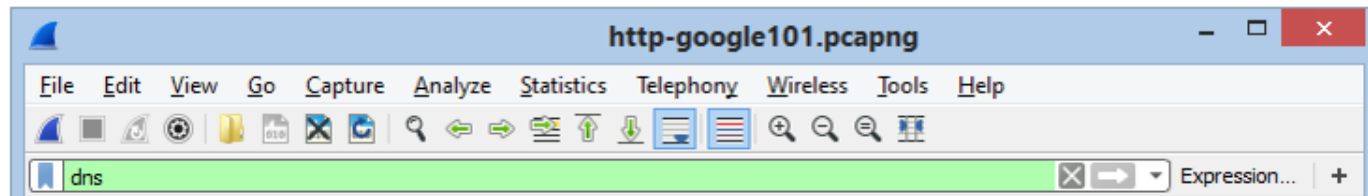
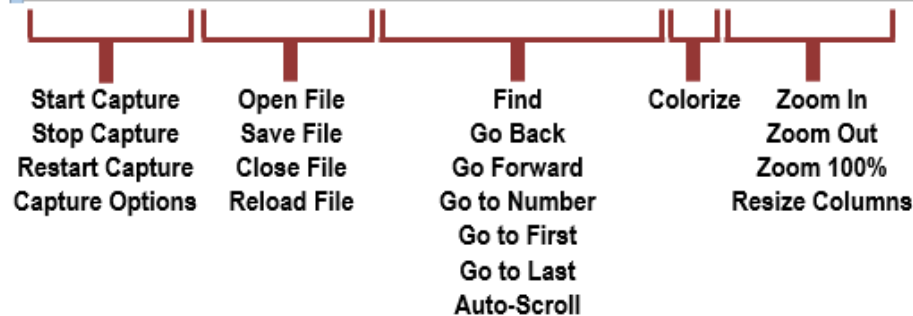
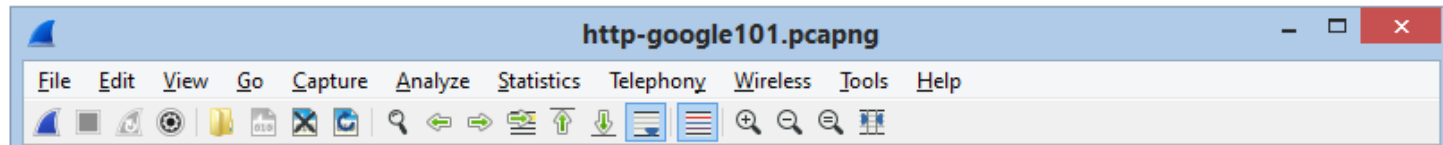
Packet Details Pane: This pane shows the hierarchical structure of the selected packet (Frame 1). The selected item is the Domain Name System (query) section, which includes a transaction ID of 0xae0b, flags of 0x0100, and one question. The question is for the domain `www.google...`.

Packet Bytes Pane: This pane shows the raw packet bytes in hexadecimal and ASCII. The selected packet (Frame 1) is 74 bytes long. The bytes are displayed in hexadecimal (00 10 00 3c 08 3d 00 00 80 11 00 00 18 06 ad dc 4b 4b) and ASCII (.<.=....KK).

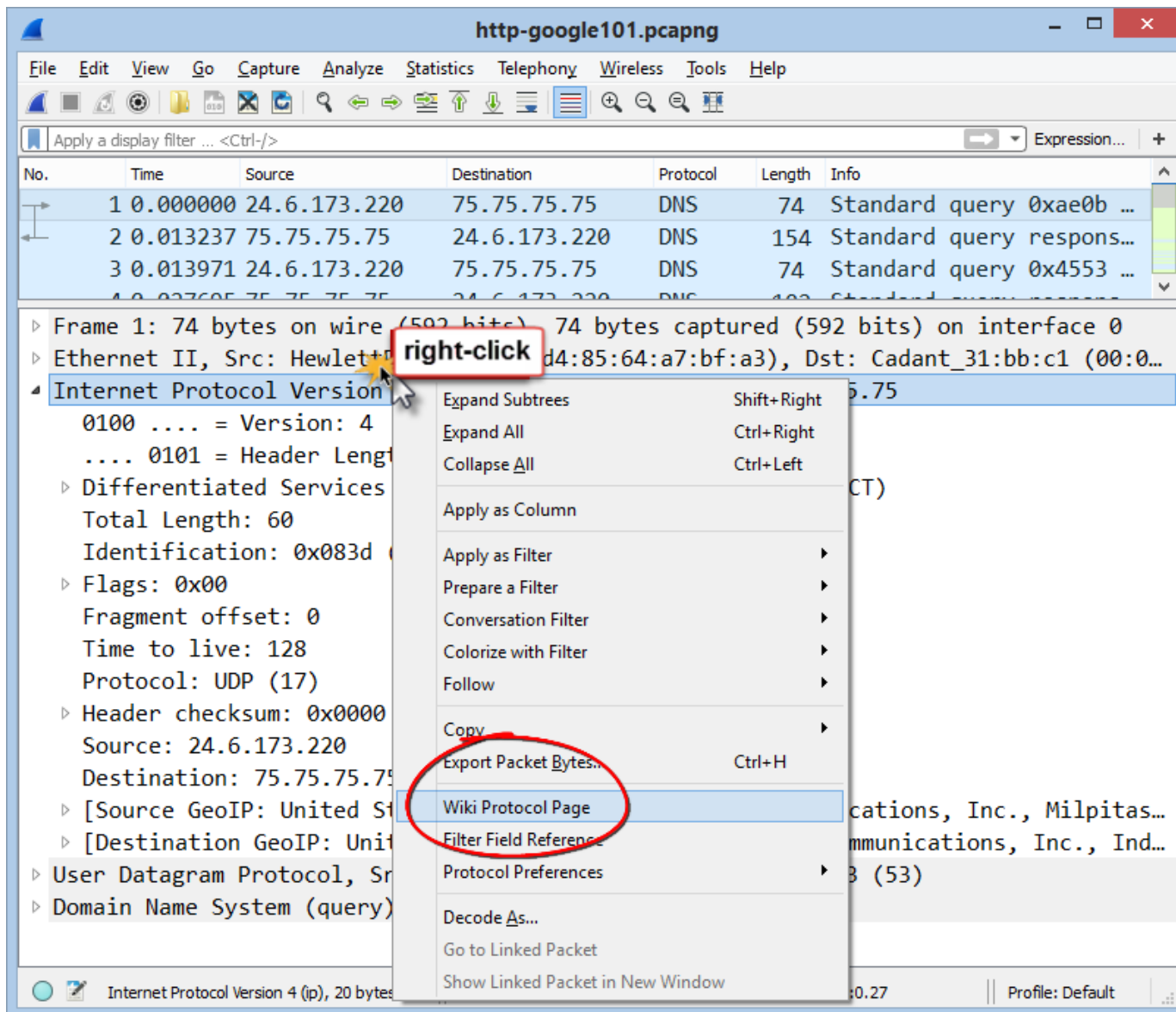
The bottom status bar shows the number of queries in the packet (2), the total number of packets (374), and the display filter (dns.count.queries, 2 bytes).

Use the Main Wireshark View

Note: The Start page appears when no trace file is open. Become accustomed to using the menus and toolbars in Wireshark.



Wireshark Resource: Wiki Pages



Customize Wireshark for Different Tasks

Columns

Alignment

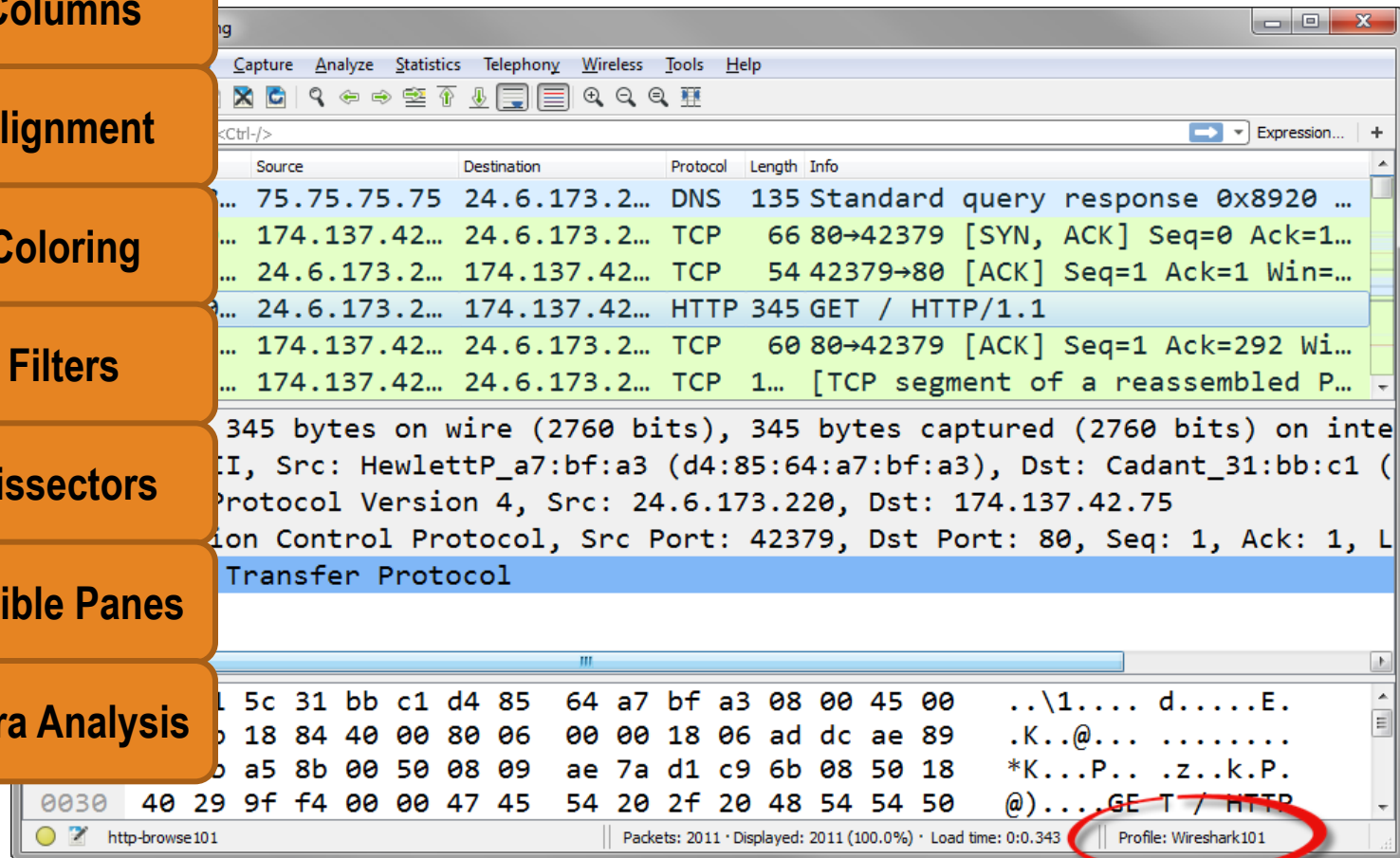
Coloring

Filters

Dissectors

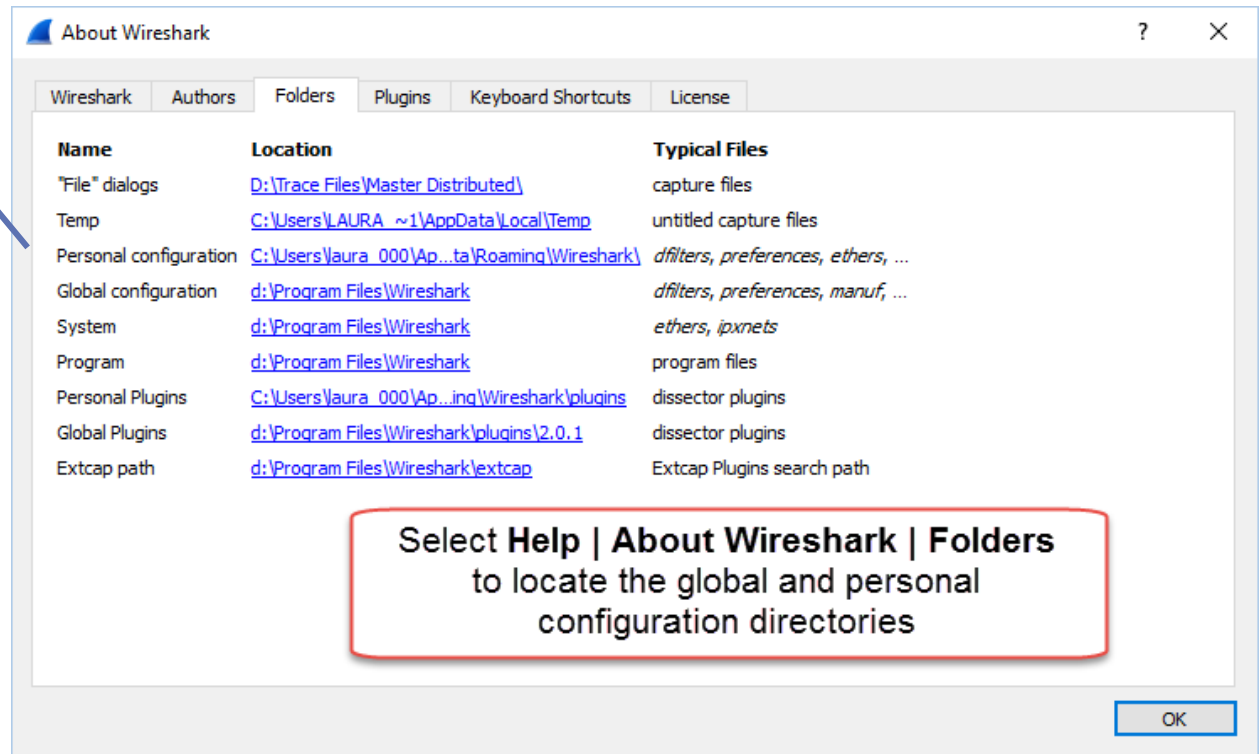
Visible Panes

Extra Analysis



Locate Key Wireshark Configuration Files

Your custom profiles are located in a profiles directory under the Personal configuration folder



Related Packets Indicator

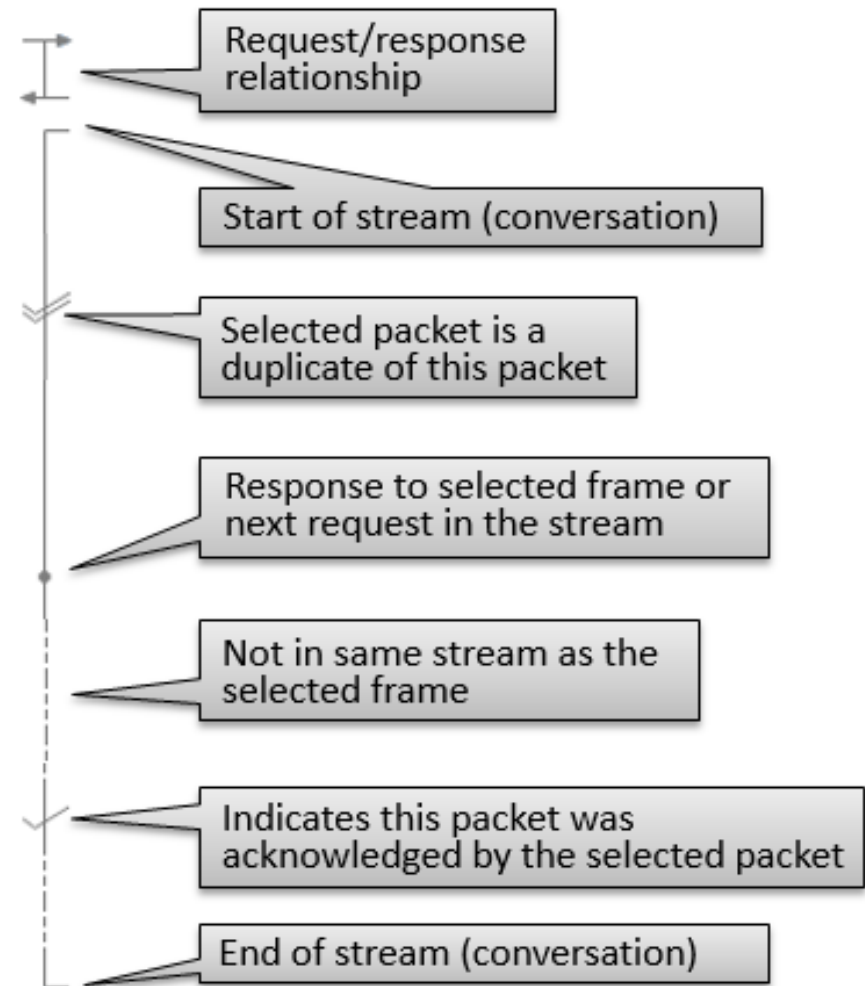
http-google101.pcapng

File Edit View Go Ca Wireless

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

No.	Time	Source	Destination
1	0.000000	24.6.173.220	75.75.75
2	0.013237	75.75.75.75	24.6.173.
3	0.013971	24.6.173.220	75.75.75
4	0.027695	75.75.75.75	24.6.173.
5	0.028699	24.6.173.220	74.125.224
6	0.046071	74.125.224.80	24.6.173
7	0.046258	24.6.173.220	74.125.224
8	0.046998	24.6.173.220	74.125.224
9	0.065701	74.125.224.80	24.6.173

The Related Packets Indicator is actually part of the No. column



Work with Columns in the Packet List Pane

The screenshot shows the Wireshark interface with the file `http-esp101.pcapng` open. The Packet List pane displays a table of network packets. A right-click context menu is open over the 'Time to live' column of the first packet. The menu options are:

- Expand Subtrees (Shift+Right)
- Expand All (Ctrl+Right)
- Collapse All (Ctrl+Left)
- Apply as Column** (highlighted with a red circle)
- Apply as Filter
- Prepare a Filter
- Conversation Filter
- Colorize with Filter
- Follow
- Copy
- Export Packet Bytes... (Ctrl+H)
- Wiki Protocol Page
- Filter Field Reference
- Protocol Preferences
- Decode As...
- Go to Linked Packet
- Open Packet in New Window

The 'Time to live' column in the packet list is highlighted in yellow. The first packet's details show:

```

No. 1      Time 0.000000 Source 24.6.173.220 Destination 75.75.75.75 Protocol DNS Length 72 Time to live 128 Info Standard quer...
No. 2      Time 0.014749 Source 75.75.75.75 Destination 24.6.173.220 Protocol DNS Length 88 Time to live 59 Info Standard quer...
No. 3      Time 0.015870 Source 24.6.173.220 Destination 75.75.75.75 Protocol DNS Length 72 Time to live 128 Info Standard quer...
No. 4      Time 0.029205 Source 75.75.75.75 Destination 24.6.173.220 Protocol DNS Length 122 Time to live 59 Info Standard quer...
  
```

The details pane for the selected packet shows the following information:

```

Internet Protocol Version 4, Src: 24.6.173.220, Dst: 75.75.75.75
  0100 .... = Version: 4
  .... 0101 = Header Length: 20 (Minimum)
  Differentiated Services Code Point 000000 = Diff-Serv Code Point: 0
  .... ..00 = Explicit Congestion Notification: Not Set
  Total Length: 58
  Identification: 0x00000000
  Flags: 0x00
    0... .... = Reserved: Not Set
    .0... .... = Don't Fragment: Not Set
    ..0. .... = More Fragments: Not Set
  Fragment offset: 0
  Time to live: 128
  
```

A red box labeled 'right-click' points to the context menu.

Sort and Reorder Columns

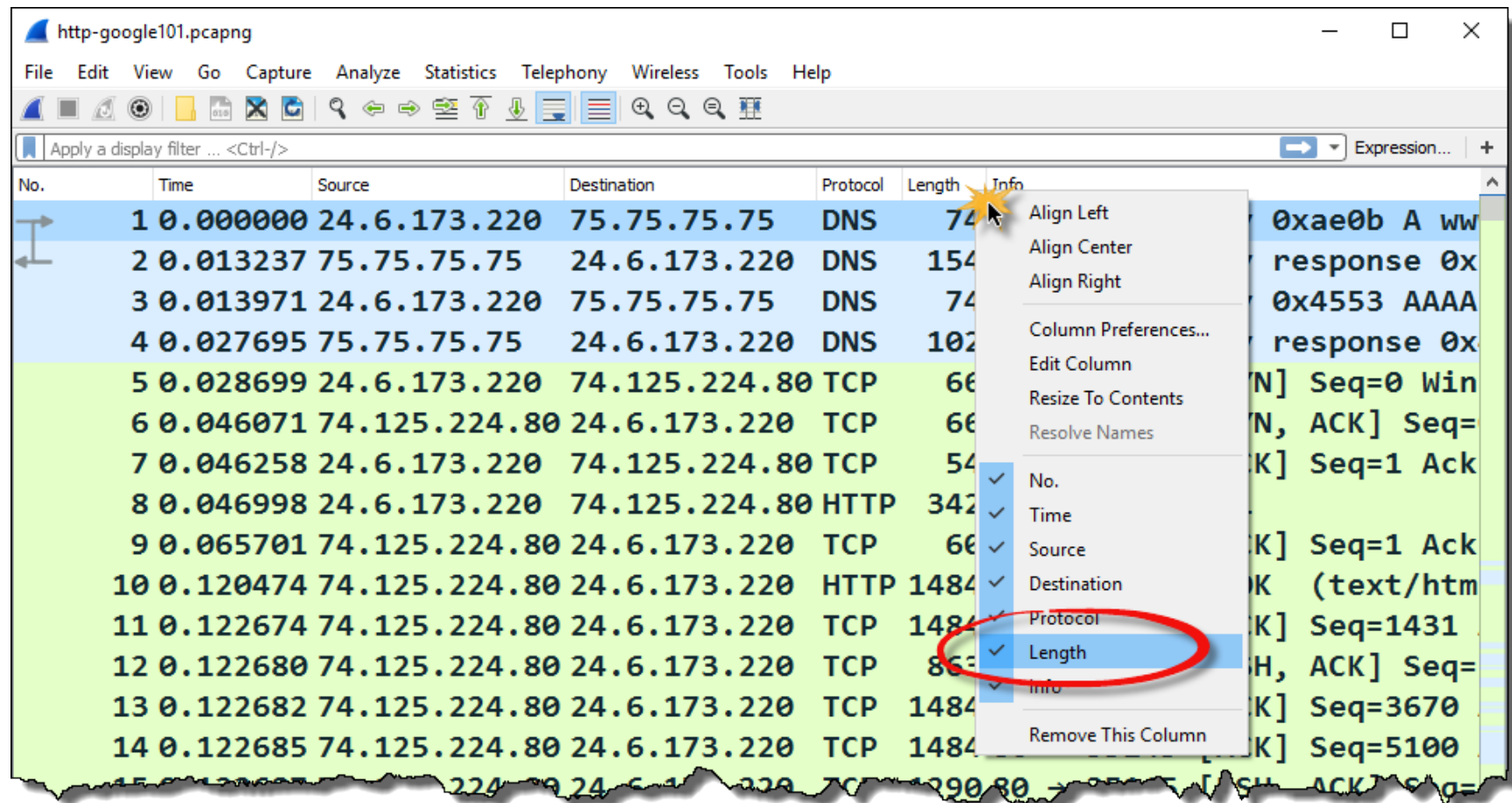
The top screenshot shows the Wireshark interface with the packet list table. A red circle highlights the 'Protocol' column header, and a red arrow points to it with the text 'Click Here'. The table shows several DNS queries and responses.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	24.6.173.220	75.75.75.75	DNS	74	Standard query 0xae0b A www
2	0.013237	75.75.75.75	24.6.173.220	DNS	154	Standard query response 0xae0b A www
3	0.013971	24.6.173.220	75.75.75.75	DNS	74	Standard query 0x4553 AAAA
4	0.027695	75.75.75.75	24.6.173.220	DNS	102	Standard query response 0x4553 AAAA
231	0.558709	24.6.173.220	75.75.75.75	DNS	75	Standard query 0x6fbd A plus
232	0.558727	24.6.173.220	75.75.75.75	DNS	75	Standard query 0xa730 A map
233	0.558808	24.6.173.220	75.75.75.75	DNS	75	Standard query 0xe258 A play
237	0.563201	24.6.173.220	75.75.75.75	DNS	75	Standard query 0x75ab A ssl
238	0.570053	75.75.75.75	24.6.173.220	DNS	251	Standard query response 0x75ab A ssl

The bottom screenshot shows the same Wireshark interface, but the columns have been reordered. A red circle highlights the 'Source' column header, and a red arrow points to it with the text 'Click and Drag'. The table shows the same DNS queries and responses, but now includes TCP packets at the bottom.

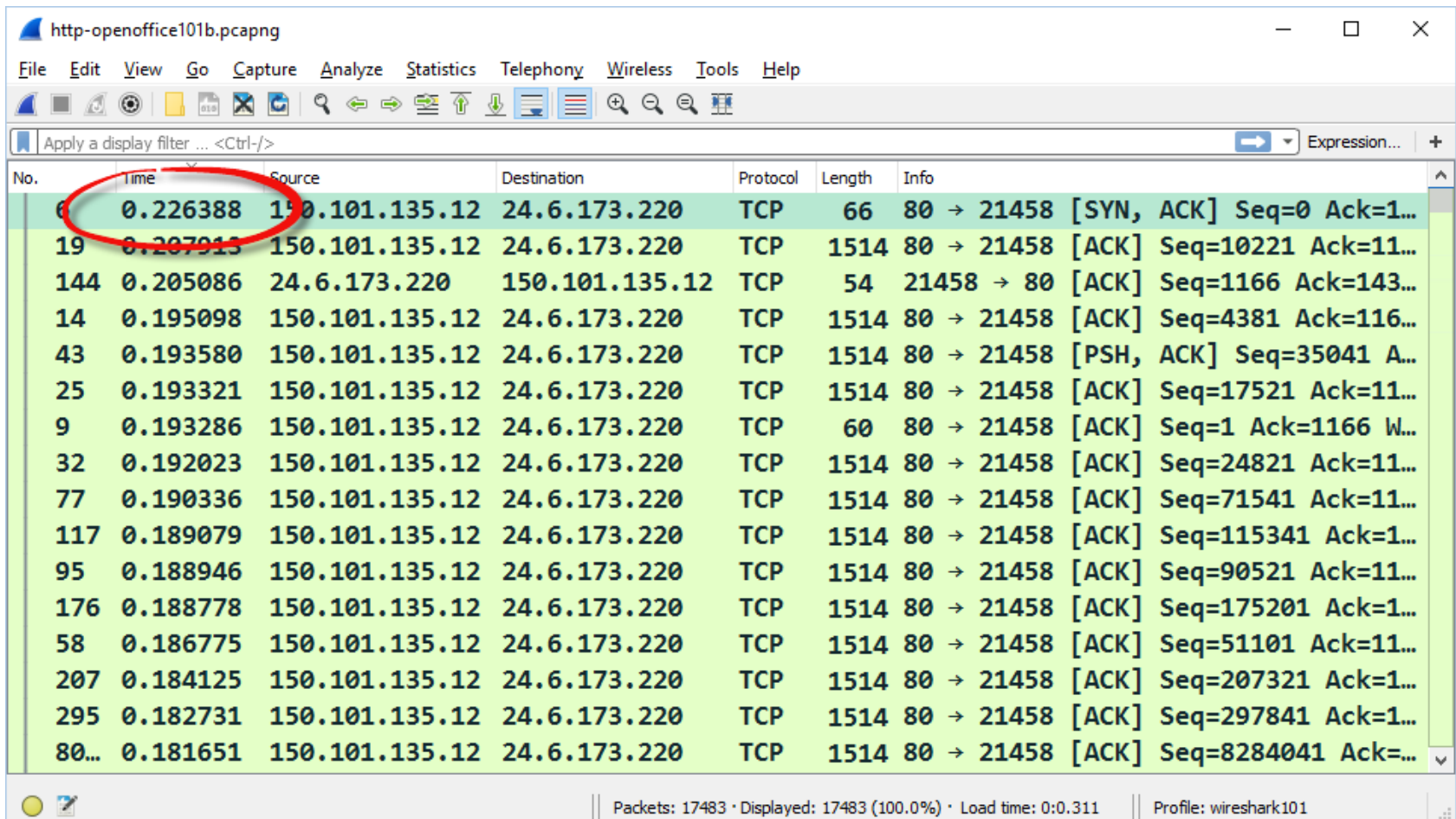
No.	Source	Destination	Protocol	Time	Length	Info
1	24.6.173.220	75.75.75.75	DNS	0.000000	74	Standard query 0xae0b A www
2	75.75.75.75	24.6.173.220	DNS	0.013237	154	Standard query response 0xae0b A www
3	24.6.173.220	75.75.75.75	DNS	0.013971	74	Standard query 0x4553 AAAA
4	75.75.75.75	24.6.173.220	DNS	0.027695	102	Standard query response 0x4553 AAAA
5	24.6.173.220	74.125.224.80	TCP	0.028699	66	35145 → 80 [SYN] Seq=0 Win=0
6	74.125.224.80	24.6.173.220	TCP	0.046071	66	80 → 35145 [SYN, ACK] Seq=0

Hide, Display, Rename, and Remove Columns



Change the Time Column Setting

View | Time Display Format | Seconds Since Previous Displayed Packet



The screenshot shows the Wireshark interface with the packet list pane. The 'Time' column is circled in red. The status bar at the bottom indicates 17483 packets displayed.

No.	Time	Source	Destination	Protocol	Length	Info
6	0.226388	150.101.135.12	24.6.173.220	TCP	66	80 → 21458 [SYN, ACK] Seq=0 Ack=1...
19	0.207915	150.101.135.12	24.6.173.220	TCP	1514	80 → 21458 [ACK] Seq=10221 Ack=11...
144	0.205086	24.6.173.220	150.101.135.12	TCP	54	21458 → 80 [ACK] Seq=1166 Ack=143...
14	0.195098	150.101.135.12	24.6.173.220	TCP	1514	80 → 21458 [ACK] Seq=4381 Ack=116...
43	0.193580	150.101.135.12	24.6.173.220	TCP	1514	80 → 21458 [PSH, ACK] Seq=35041 A...
25	0.193321	150.101.135.12	24.6.173.220	TCP	1514	80 → 21458 [ACK] Seq=17521 Ack=11...
9	0.193286	150.101.135.12	24.6.173.220	TCP	60	80 → 21458 [ACK] Seq=1 Ack=1166 W...
32	0.192023	150.101.135.12	24.6.173.220	TCP	1514	80 → 21458 [ACK] Seq=24821 Ack=11...
77	0.190336	150.101.135.12	24.6.173.220	TCP	1514	80 → 21458 [ACK] Seq=71541 Ack=11...
117	0.189079	150.101.135.12	24.6.173.220	TCP	1514	80 → 21458 [ACK] Seq=115341 Ack=1...
95	0.188946	150.101.135.12	24.6.173.220	TCP	1514	80 → 21458 [ACK] Seq=90521 Ack=11...
176	0.188778	150.101.135.12	24.6.173.220	TCP	1514	80 → 21458 [ACK] Seq=175201 Ack=1...
58	0.186775	150.101.135.12	24.6.173.220	TCP	1514	80 → 21458 [ACK] Seq=51101 Ack=11...
207	0.184125	150.101.135.12	24.6.173.220	TCP	1514	80 → 21458 [ACK] Seq=207321 Ack=1...
295	0.182731	150.101.135.12	24.6.173.220	TCP	1514	80 → 21458 [ACK] Seq=297841 Ack=1...
80...	0.181651	150.101.135.12	24.6.173.220	TCP	1514	80 → 21458 [ACK] Seq=8284041 Ack=...

Packets: 17483 · Displayed: 17483 (100.0%) · Load time: 0:0.311 | Profile: wireshark101

Right-Click in the Packet List Pane

The image shows the Wireshark network protocol analyzer interface. The title bar indicates the file is 'http-google101.pcapng'. The menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. The toolbar contains various icons for file operations, capture, and analysis. A display filter bar at the top shows 'Apply a display filter ... <Ctrl-/>'. The main packet list pane displays a table of captured packets. Packet 3 is selected, and a right-click context menu is open over it. The menu options include: Mark/Unmark Packet (Ctrl+M), Ignore/Unignore Packet (Ctrl+D), Set/Unset Time Reference (Ctrl+T), Time Shift... (Ctrl+Shift+T), Packet Comment..., Edit Resolved Name, Apply as Filter, Prepare a Filter, Conversation Filter, Colorize Conversation, SCTP, Follow, Copy, Protocol Preferences, Decode As..., and Show Packet in New Window. The packet details pane below the list shows the structure of packet 3: Frame 3: 74 bytes on wire (592 bits), 7 Ethernet II, Src: HewlettP_a7:bf:a3 (d4), Internet Protocol Version 4, Src: 24.6., User Datagram Protocol, Src Port: 63342. The packet bytes pane at the bottom shows the raw data in hexadecimal and ASCII.

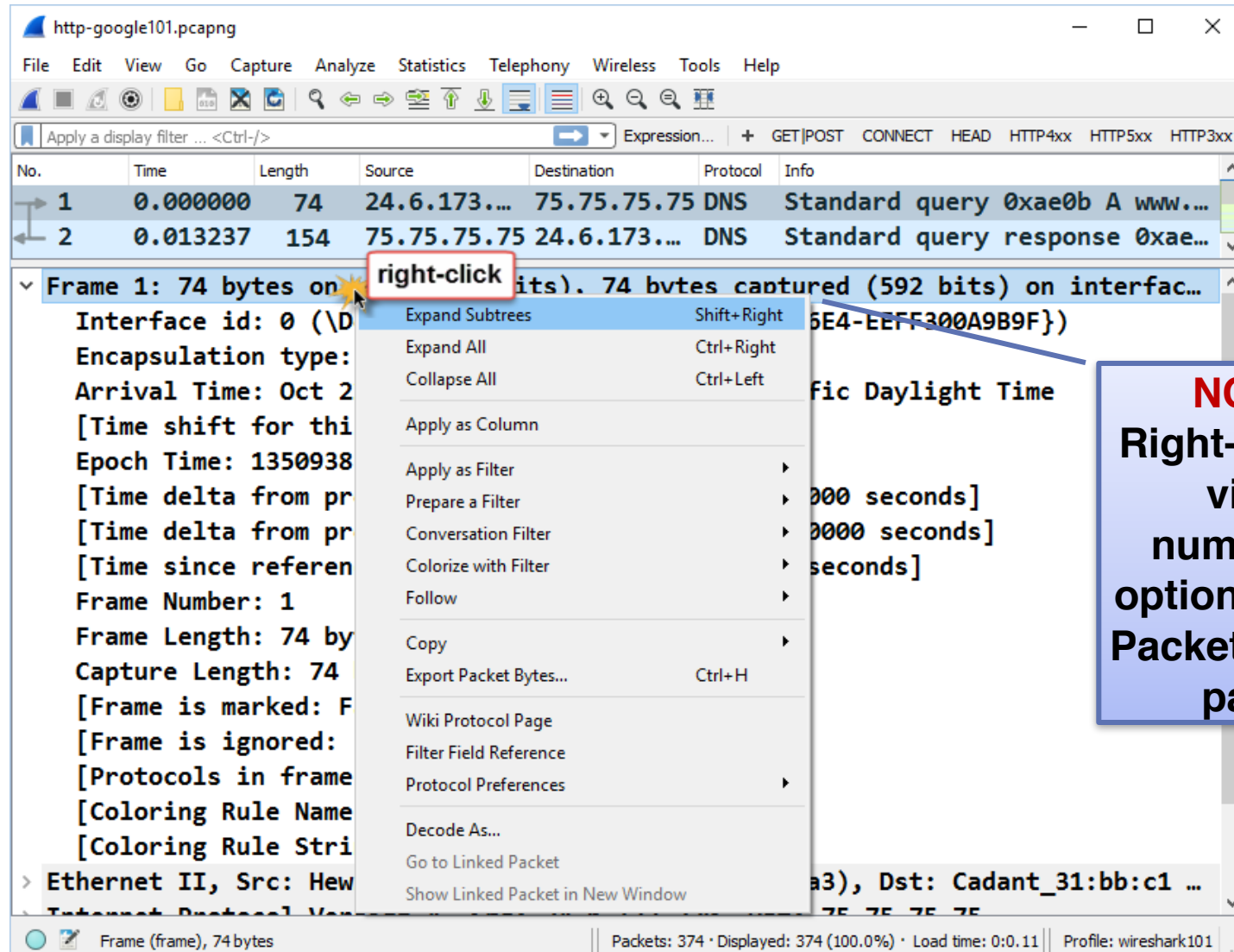
No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	24.6.173.220	75.75.75.75	DNS	74	Standard query 0xae0b A www
2	0.013237	75.75.75.75	24.6.173.220	DNS	154	Standard query response 0xa
3	0.013971	24.6.173.220	75.75.75.75	DNS	74	Standard query 0x4553 AAAA
4	0.027695	75.75.75.75	24.6.173.220	TCP	60	4553 → 443 [ACK] Seq=0 Win=
5	0.028699	24.6.173.220	74.125.224.8	TCP	60	443 → 4553 [ACK] Seq=0
6	0.046071	74.125.224.80	24.6.173.220	TCP	60	4553 → 443 [ACK] Seq=0
7	0.046258	24.6.173.220	74.125.224.8	TCP	60	443 → 4553 [ACK] Seq=1 Ack=
8	0.046998	24.6.173.220	74.125.224.8	TCP	60	4553 → 443 [ACK] Seq=1 Ack=
9	0.065701	74.125.224.80	24.6.173.220	TCP	60	443 → 4553 [ACK] Seq=1 Ack=
10	0.120474	74.125.224.80	24.6.173.220	TCP	60	4553 → 443 [ACK] Seq=1 Ack=

Frame 3: 74 bytes on wire (592 bits), 7 Ethernet II, Src: HewlettP_a7:bf:a3 (d4), Internet Protocol Version 4, Src: 24.6., User Datagram Protocol, Src Port: 63342

0000 00 01 5c 31 bb c1 d4 85 64 a7 bf
 0010 00 3c 08 3e 00 00 80 11 00 00 18
 0020 4b 4b f7 6e 00 35 00 28 5c b2 45
 0030 00 00 00 00 00 00 03 77 77 77 06 67 6f 6f 67 6cw ww.googl

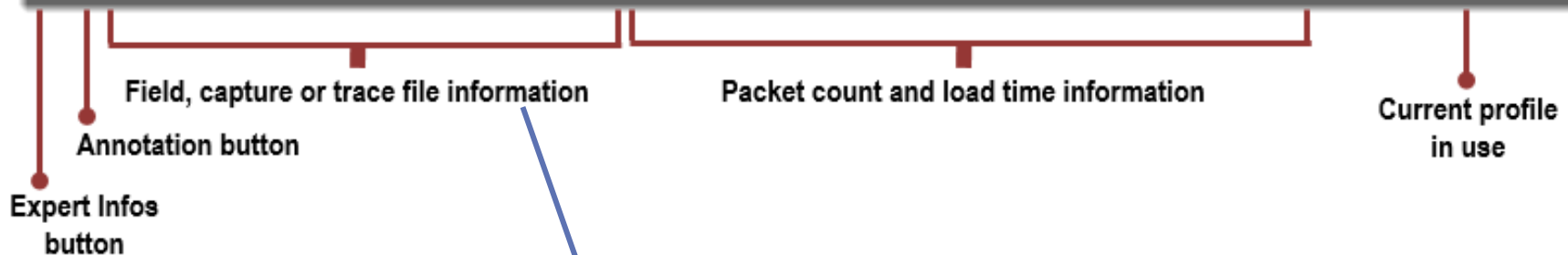
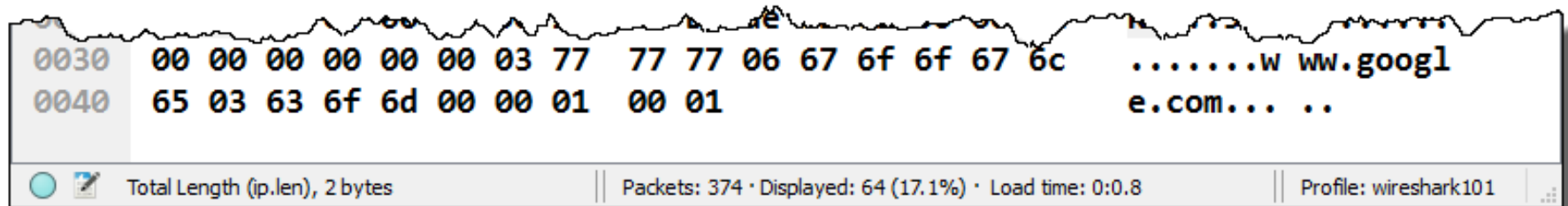
Packets: 374 · Displayed: 374 (100.0%) · Load time: 0:0.7 | Profile: Default

Right-Click in the Packet Details Pane



NOTE
Right-click to view numerous options in the Packet Details pane

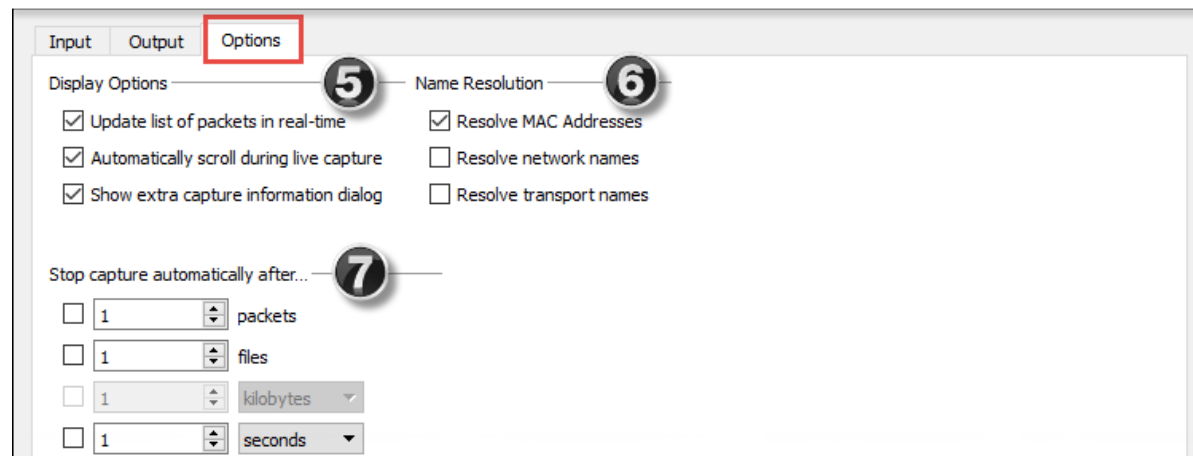
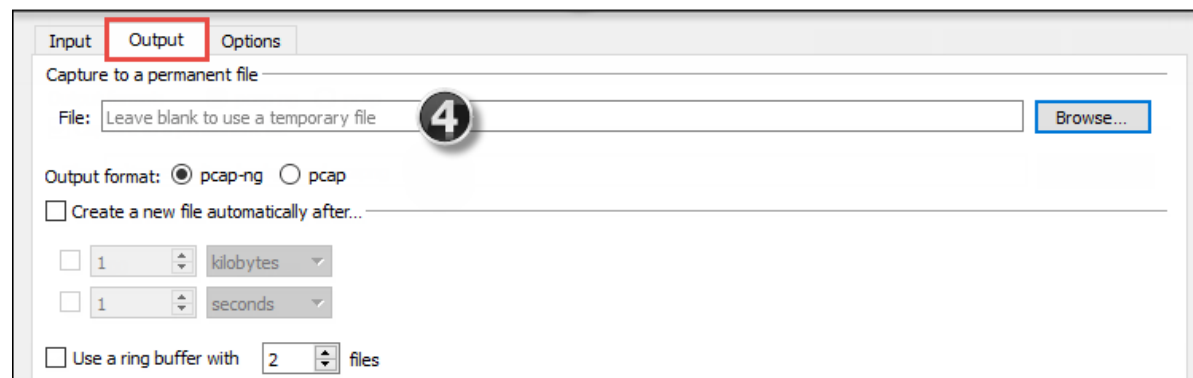
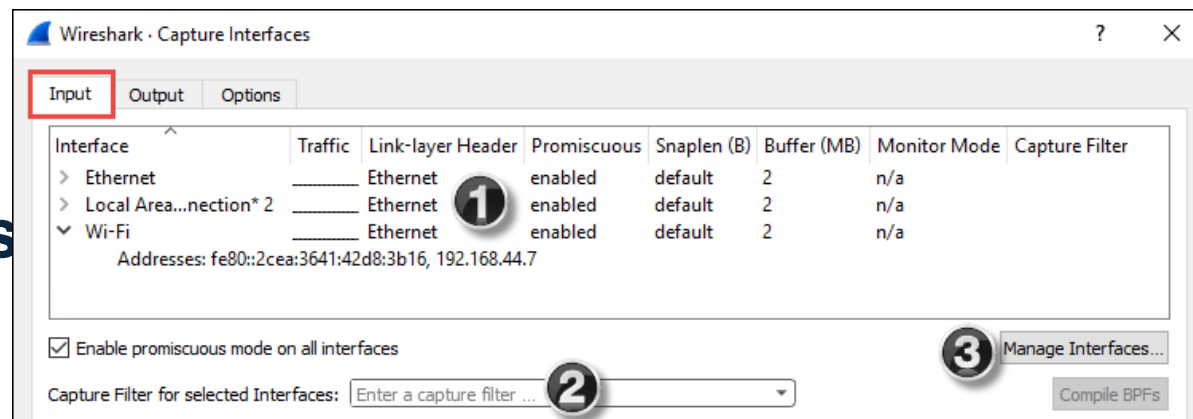
Pay Attention to the Status Bar



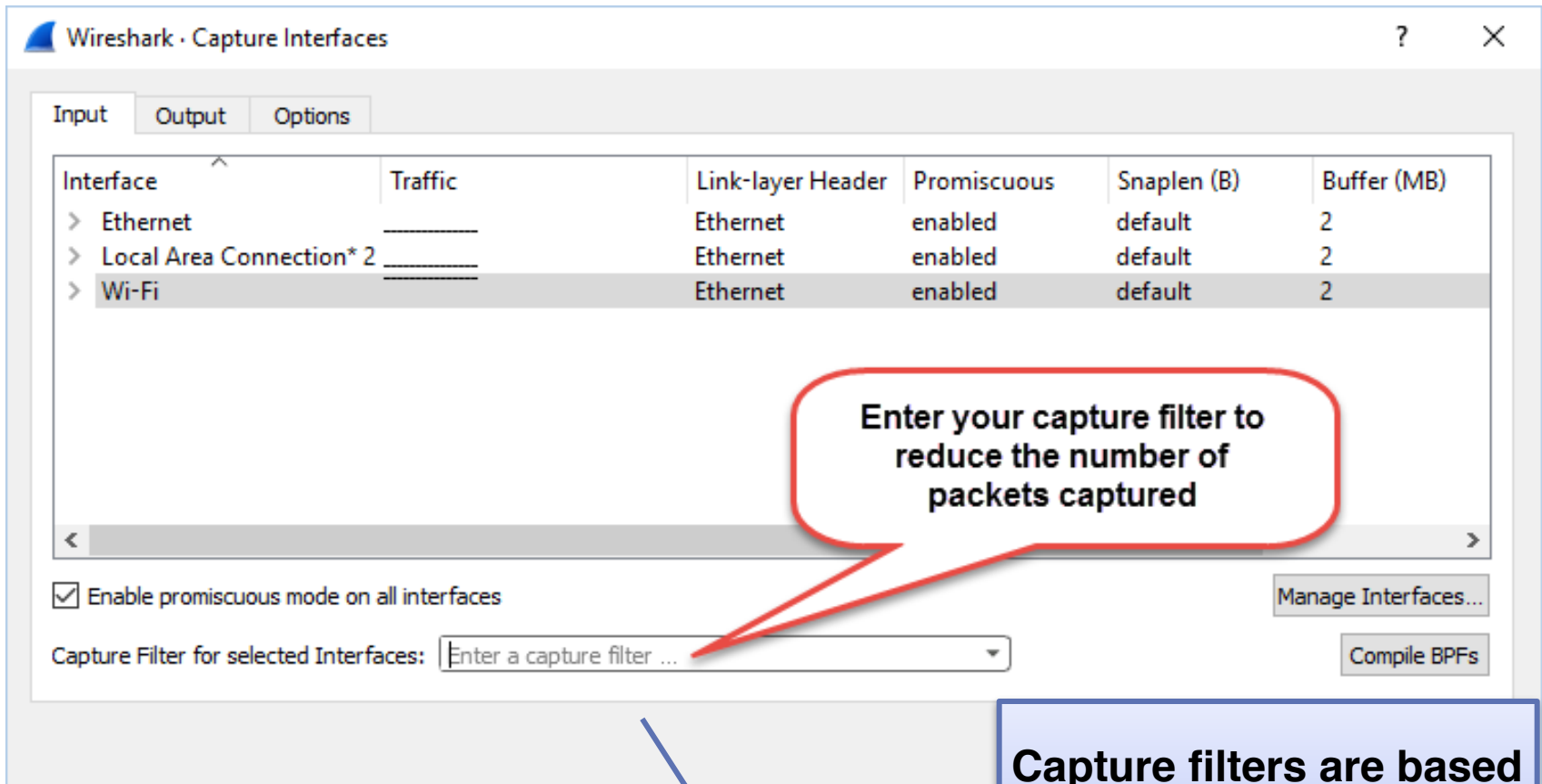
NOTE

Contents of this column changes depending on what you've highlighted in the three panes

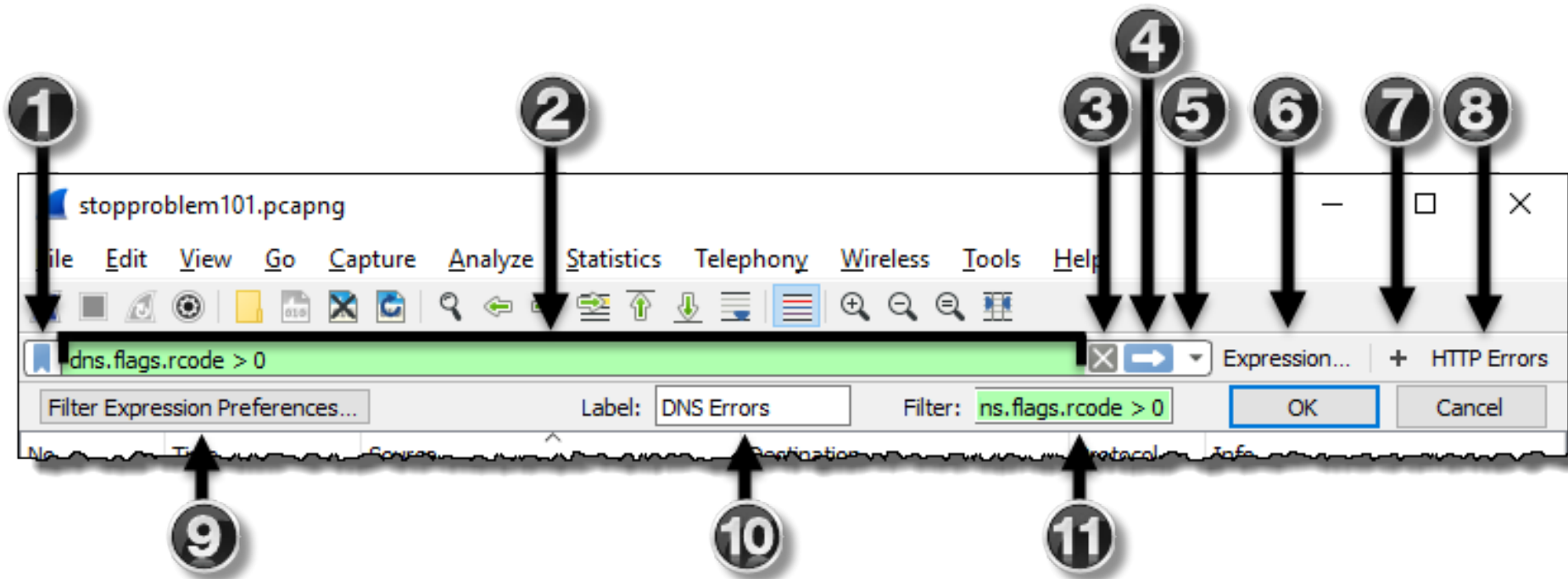
Capture Options



Apply Capture Filters



Display Filter Area



Use Proper Display Filter Syntax (Wireshark-Specific Syntax)

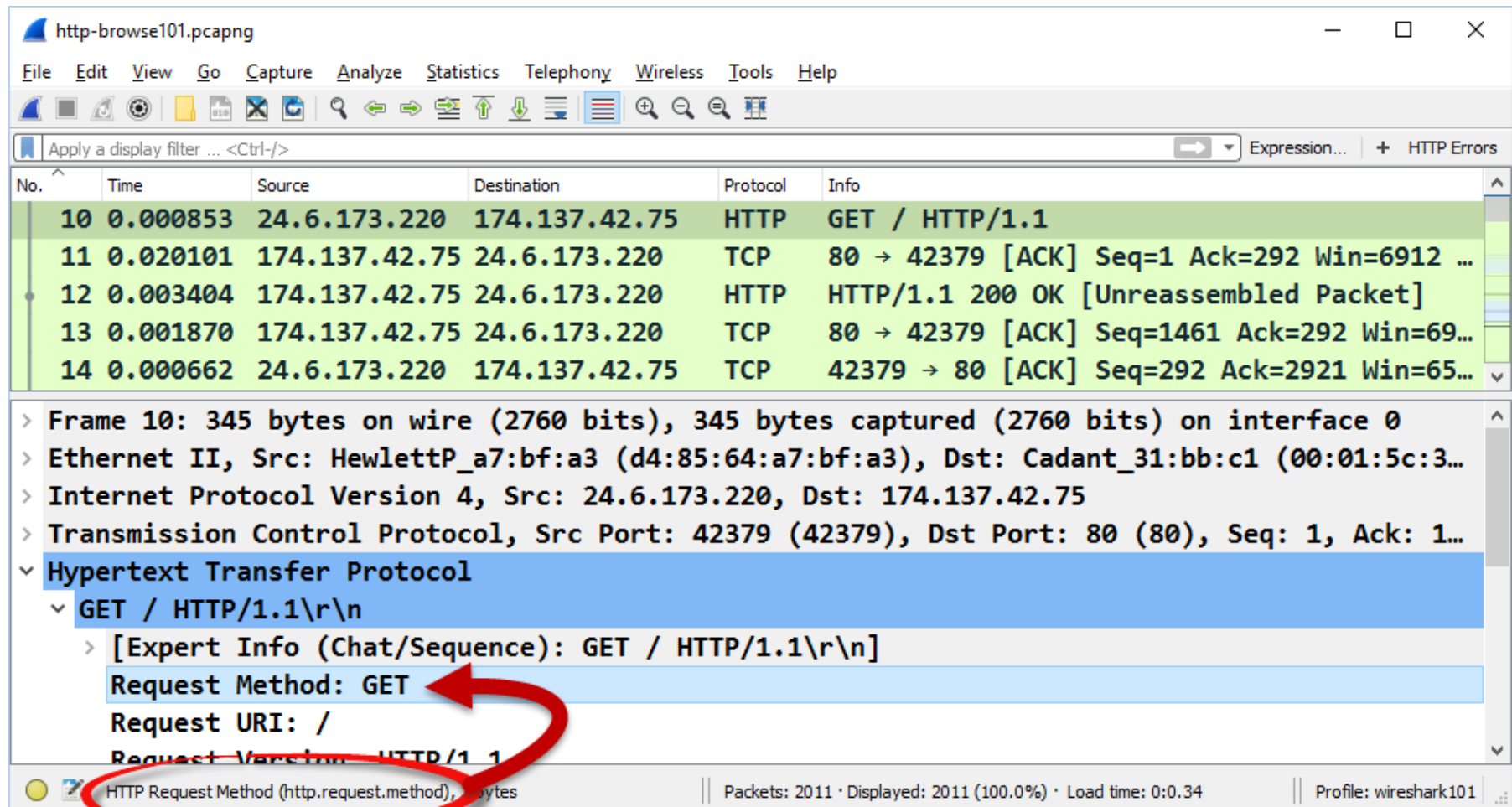
Filter Type	Filter Example
Protocol	<code>arp</code>
Application	<code>dns</code>
Field Existence	<code>http.host</code>
Characteristic Existence	<code>tcp.analysis.zero_window</code>
Field Value	<code>http.host=="www.wireshark.org"</code>
Regex* Search Term	<code>http.host matches "\.(?i)(exe zip)"</code>

*Wireshark uses the Pearl-Compatible Regular Expression (PCRE) engine.

Display Filter Techniques

- **Type in** if you know the field names/syntax (error detection mechanism)
- **Auto-complete** to walk you through building a display filter
- **Expressions** to walk you through building a display filter with/without comparison operators
- **Recall** saved or previously-used filter
- **Right-click** in the Packet List pane for conversation filters or on a Table row
- **Create buttons** out of your favorite display filters.

Learn the Field Names



http-browse101.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression... + HTTP Errors

No.	Time	Source	Destination	Protocol	Info
10	0.000853	24.6.173.220	174.137.42.75	HTTP	GET / HTTP/1.1
11	0.020101	174.137.42.75	24.6.173.220	TCP	80 → 42379 [ACK] Seq=1 Ack=292 Win=6912 ...
12	0.003404	174.137.42.75	24.6.173.220	HTTP	HTTP/1.1 200 OK [Unreassembled Packet]
13	0.001870	174.137.42.75	24.6.173.220	TCP	80 → 42379 [ACK] Seq=1461 Ack=292 Win=69...
14	0.000662	24.6.173.220	174.137.42.75	TCP	42379 → 80 [ACK] Seq=292 Ack=2921 Win=65...

> Frame 10: 345 bytes on wire (2760 bits), 345 bytes captured (2760 bits) on interface 0

> Ethernet II, Src: HewlettP_a7:bf:a3 (d4:85:64:a7:bf:a3), Dst: Cadant_31:bb:c1 (00:01:5c:3...

> Internet Protocol Version 4, Src: 24.6.173.220, Dst: 174.137.42.75

> Transmission Control Protocol, Src Port: 42379 (42379), Dst Port: 80 (80), Seq: 1, Ack: 1...

▼ Hypertext Transfer Protocol

▼ GET / HTTP/1.1\r\n

> [Expert Info (Chat/Sequence): GET / HTTP/1.1\r\n]

Request Method: GET

Request URI: /

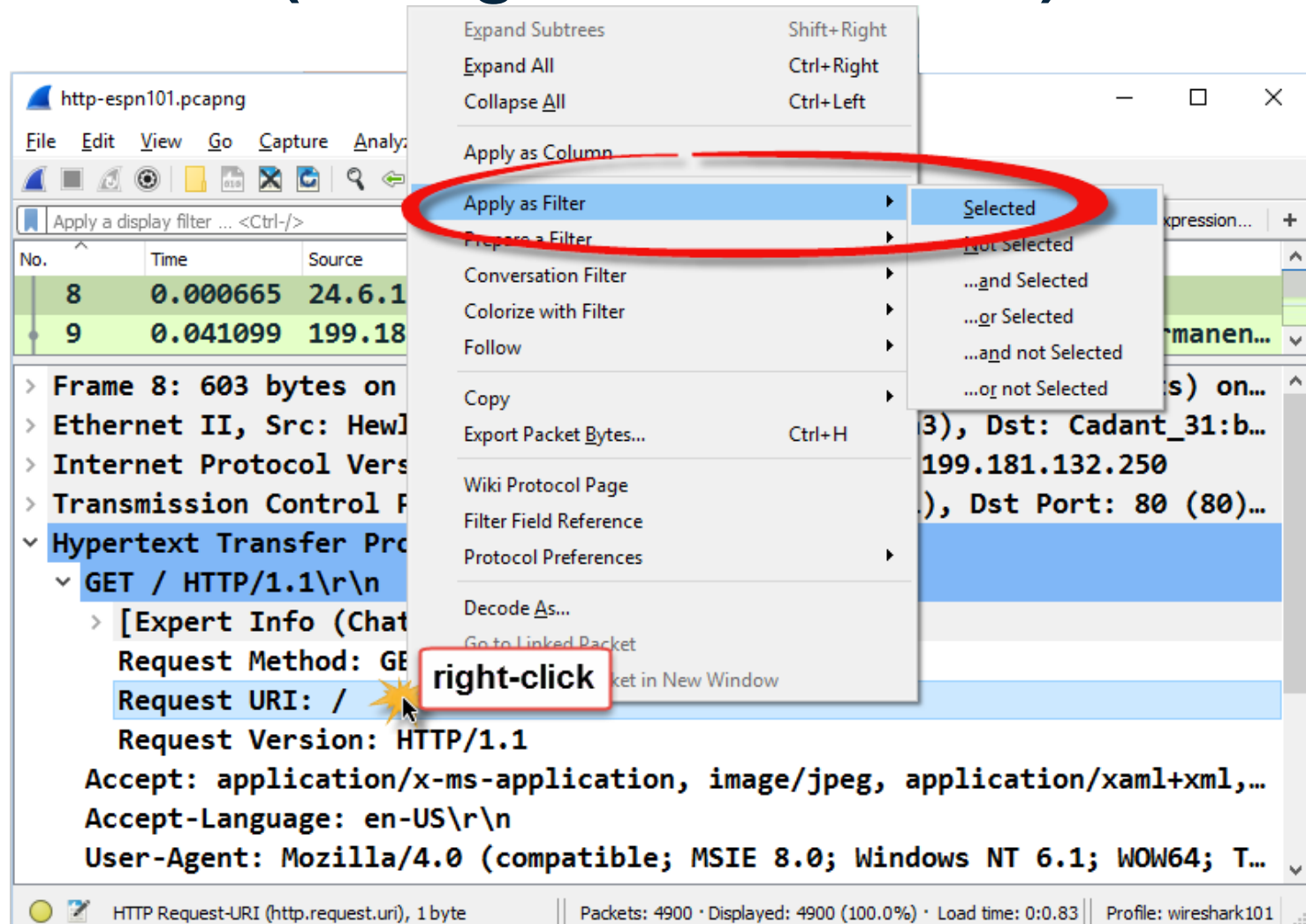
Request Version: HTTP/1.1

HTTP Request Method (http.request.method), bytes

Packets: 2011 · Displayed: 2011 (100.0%) · Load time: 0:0.34

Profile: wireshark101

Quickly Filter on a Field in a Packet (the right-click method)



Use Auto-Complete to Build Display Filters

Top Screenshot: http.request.

Auto-complete lists all available display filters that begin with http.request.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	173.220.75.75	75.75.75.75	DNS	Standard query 0x9ba8 A www.wireshark...
2	0.000000	75.75.75.75	173.220.75.75	DNS	Standard query response 0x9ba8 A www.w...
3	0.000783	24.6.173.220	75.75.75.75	DNS	Standard query 0x8920 AAAA www.wiresha...
4	0.030017	24.6.173.220	75.75.75.75	DNS	Standard query 0x8920 AAAA www.wiresha...
5	0.003284	75.75.75.75	24.6.173.220	DNS	Standard query response 0x8920 AAAA ww...

Bottom Screenshot: tcp.analysis.

TCP packets analyzed by Wireshark's Expert System can be viewed using one of the filters beginning with tcp.analysis.

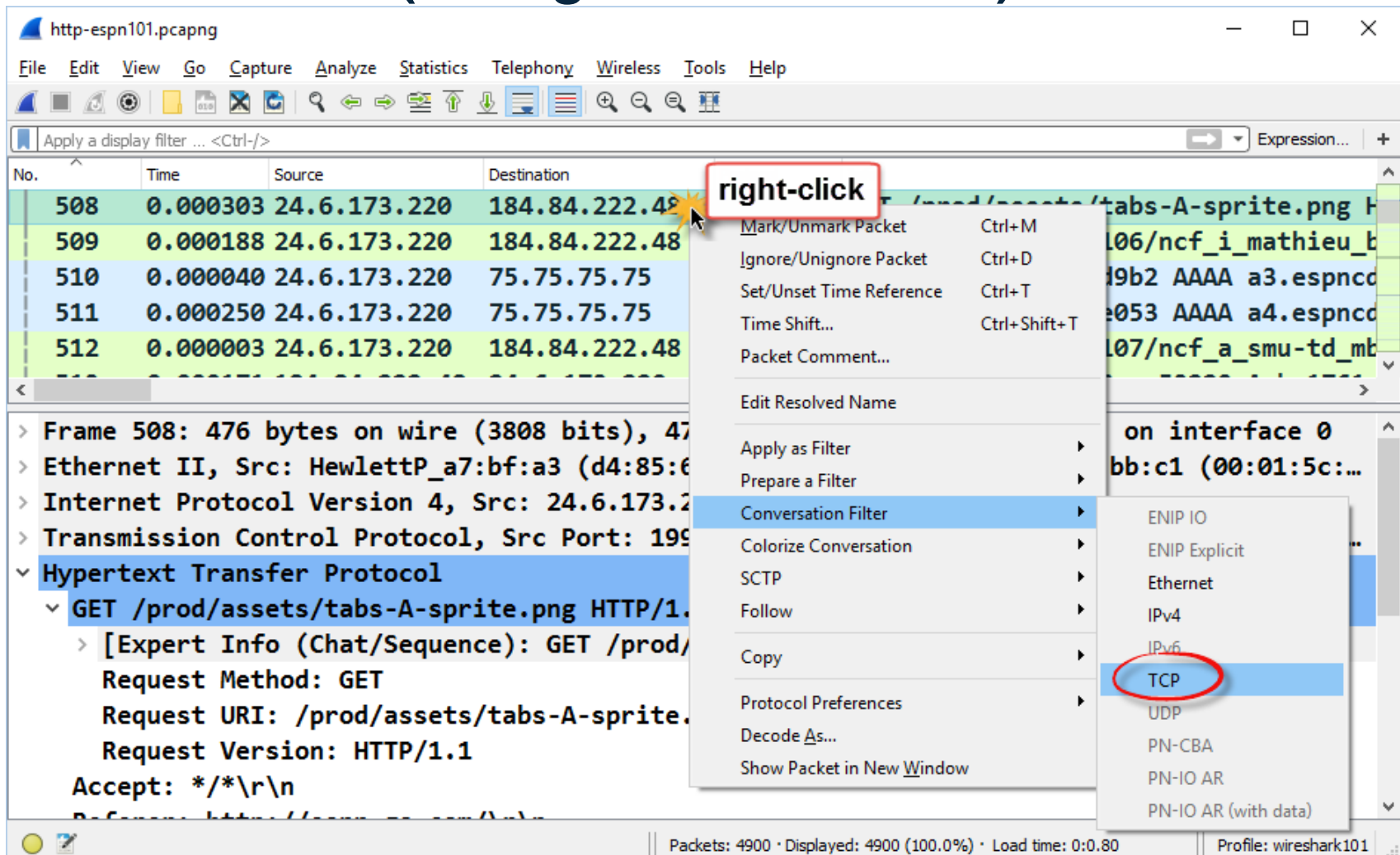
No.	Time	Source	Destination	Protocol	Info
1	0.000000	75.75.75.75	24.6.173.220	DNS	Standard query 0x9ba8 A www.wireshark...
2	0.000000	24.6.173.220	75.75.75.75	DNS	Standard query response 0x9ba8 A www.w...
3	0.000783	75.75.75.75	24.6.173.220	DNS	Standard query 0x8920 AAAA www.wiresha...
4	0.030017	75.75.75.75	24.6.173.220	DNS	Standard query 0x8920 AAAA www.wiresha...
5	0.003284	24.6.173.220	75.75.75.75	DNS	Standard query response 0x8920 AAAA ww...
6	0.000000	174.137.42.75	24.6.173.220	TCP	Seq=0 Win=8192 Len=0 ...
7	0.000000	24.6.173.220	174.137.42.75	TCP	Seq=0 Win=8192 Len=0 ...
8	0.000000	24.6.173.220	174.137.42.75	TCP	ACK] Seq=0 Ack=1 Win=...
9	0.000000	174.137.42.75	24.6.173.220	TCP	Seq=1 Ack=1 Win=65700...
10	0.000000	174.137.42.75	24.6.173.220	HTTP	GET / HTTP/1.1
11	0.000000	24.6.173.220	174.137.42.75	TCP	80 → 42379 [ACK] Seq=1 Ack=292 Win=691...
12	0.000000	24.6.173.220	174.137.42.75	HTTP	HTTP/1.1 200 OK [Unreassembled Packet]

Comparison Operators

Operation	English	Example	Description
<code>==</code>	<code>eq</code>	<code>ip.src == 10.2.2.2</code>	Display all IPv4 traffic from 10.2.2.2
<code>!=</code>	<code>ne</code>	<code>tcp.srcport != 80</code>	Display all TCP traffic from any port except port 80
<code>></code>	<code>gt</code>	<code>frame.time_relative > 1</code>	Display packets that arrived more than 1 second after the previous packet in the trace file
<code><</code>	<code>lt</code>	<code>tcp.window_size < 1460</code>	Display when the TCP receive window size is less than 1460 bytes
<code>>=</code>	<code>ge</code>	<code>dns.count.answers >= 10</code>	Display DNS response packets that contain at least 10 answers
<code><=</code>	<code>lt</code>	<code>ip.ttl <= 10</code>	Display any packets that have 10 or less in the IP Time to Live field
	<code>contains</code>	<code>http contains "GET"</code>	Display all the HTTP client GET requests

Note: Be careful using the `!=` operator.

Filter on a Single TCP or UDP Conversation (the right-click method)



Use Filters to Spot Communication Delays

Calculate

Conversations

Timestamps

setting must be enabled to see this

[Timestamps]

section

urgent pointer: 0

> [SEQ/ACK analysis]

✓ [Timestamps]

[Time since first frame in this TCP stream: 37.248064000 seconds]

[Time since previous frame in this TCP stream: 1.518125000 seconds]

The image shows a Wireshark packet capture window for the file 'http-download101d.pcapng'. The packet list on the left shows four packets. The first and third packets are TCP retransmissions (80 → 18280) with sequence numbers 216.239.11... and 192.168.1.64. The second and fourth packets are TCP ACKs (80 → 18280) with sequence numbers 216.239.11... and 192.168.1.64. The packet details pane on the right shows the 'Timestamps' section expanded, displaying the time since the first frame (37.248064000 seconds) and the time since the previous frame (1.518125000 seconds). The status bar at the bottom indicates 'Packets: 24098 · Displayed: 4 (0.0%) · Load time: 0:0.431 | Profile: wireshark101'.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	216.239.11...	192.168.1.64	TCP	[TCP Retransmission] 80 → 18280...
2	0.885...	216.239.11...	192.168.1.64	TCP	80 → 18280 [ACK] Seq=2249861 A...
3	9.41...	216.239.11...	192.168.1.64	TCP	[TCP Retransmission] 80 → 18280...
4	129637	216.239.11...	192.168.1.64	TCP	80 → 18280 [ACK] Seq=20165661 ...

size value: 5840
 lated window size: 5840]
 w size scaling factor: -1 (unknown)]
 um: 0x6fd5 [validation disabled]
 urgent pointer: 0
 > [SEQ/ACK analysis]
 ✓ [Timestamps]
 [Time since first frame in this TCP stream: 37.248064000 seconds]
 [Time since previous frame in this TCP stream: 1.518125000 seconds]

Time delta from previous frame...is TCP stream (tcp.time_delta | Packets: 24098 · Displayed: 4 (0.0%) · Load time: 0:0.431 | Profile: wireshark101

Use Right-Click to Follow a Stream

The screenshot shows the Wireshark interface with the file `http-esp101.pcapng` open. The packet list on the left shows several packets, with packet 8 selected. A right-click context menu is open over packet 8, and the **Follow** option is highlighted. A red arrow points from the **Follow** option to the **Follow TCP Stream (tcp.stream eq 0)** window that has opened. This window displays the details of the selected TCP stream, showing the **GET / HTTP/1.1** request and the **Accept** header with various media types.

right-click

Follow TCP Stream (tcp.stream eq 0)

GET / HTTP/1.1

Accept: application/x-ms-application, image/jpeg, application/xaml+xml, image/gif, image/pjpeg, application/x-ms-xbap, application/vnd.ms-excel, application/vnd.ms-powerpoint,

Packets: 4900 · Displayed: 7 (0.1%) · Load time: 0:0.101 | Profile: wireshark101

Filter on a Conversation from Wireshark Statistics

Wireshark · Conversations · http-esp101

Ethernet · 1 IPv4 · 37 **IPv6 · 5** **TCP · 63** UDP · 82

Address A	Port A	Address B	Port B	Packets	Bytes	Bytes A → B	Packets B → A	Bytes B → A	Rel Start	Duration	Bits/s A → B	Bits/s
24.6.173.220	19996	184.84.222.88	80	1,855	30 k	1322	1989 k	9.726479000	64.450526	3765		
24.6.173.220	19976	184.84.222.120	80	534								
24.6.173.220	19980	184.84.222.10	80	251								
24.6.173.220	19945	184.84.222.48	80	150								
24.6.173.220	19956	184.84.222.152	80	137								
24.6.173.220	19942	68.71.216.176	80	127								
24.6.173.220	19944	184.84.222.48	80	121	116 k	42						
24.6.173.220	19981	184.84.222.10	80	120	119 k	41						
24.6.173.220	20002	68.71.216.157	80	112	10 k	57						
24.6.173.220	19983	184.84.222.152	80	111	111 k	38	6082	73				
24.6.173.220	19961	74.125.224.59	80	110	103 k	37	3751	73				
24.6.173.220	19943	184.84.222.48	80	90	81 k	34	5823	56				
24.6.173.220	19950	184.84.222.48	80	88	85 k	29	4579	59				
24.6.173.220	19954	184.84.222.48	80	67	62 k	25	4006	42	58 k	3.216284000	56.403650	568
24.6.173.220	19978	184.84.222.120	80	61	59 k	21	1734	40	57 k	7.913036000	61.259194	226

right-click

- Apply as Filter
 - Selected
 - A → B
 - A → B
 - B → A
 - A → Any
 - A → Any
 - Any → A
 - Any → B
 - Any → B
 - B → Any
 - Not Selected
 - ...and Selected
 - ...or Selected
 - ...and not Selected
 - ...or not Selected
- Prepare a Filter
- Find
- Colorize

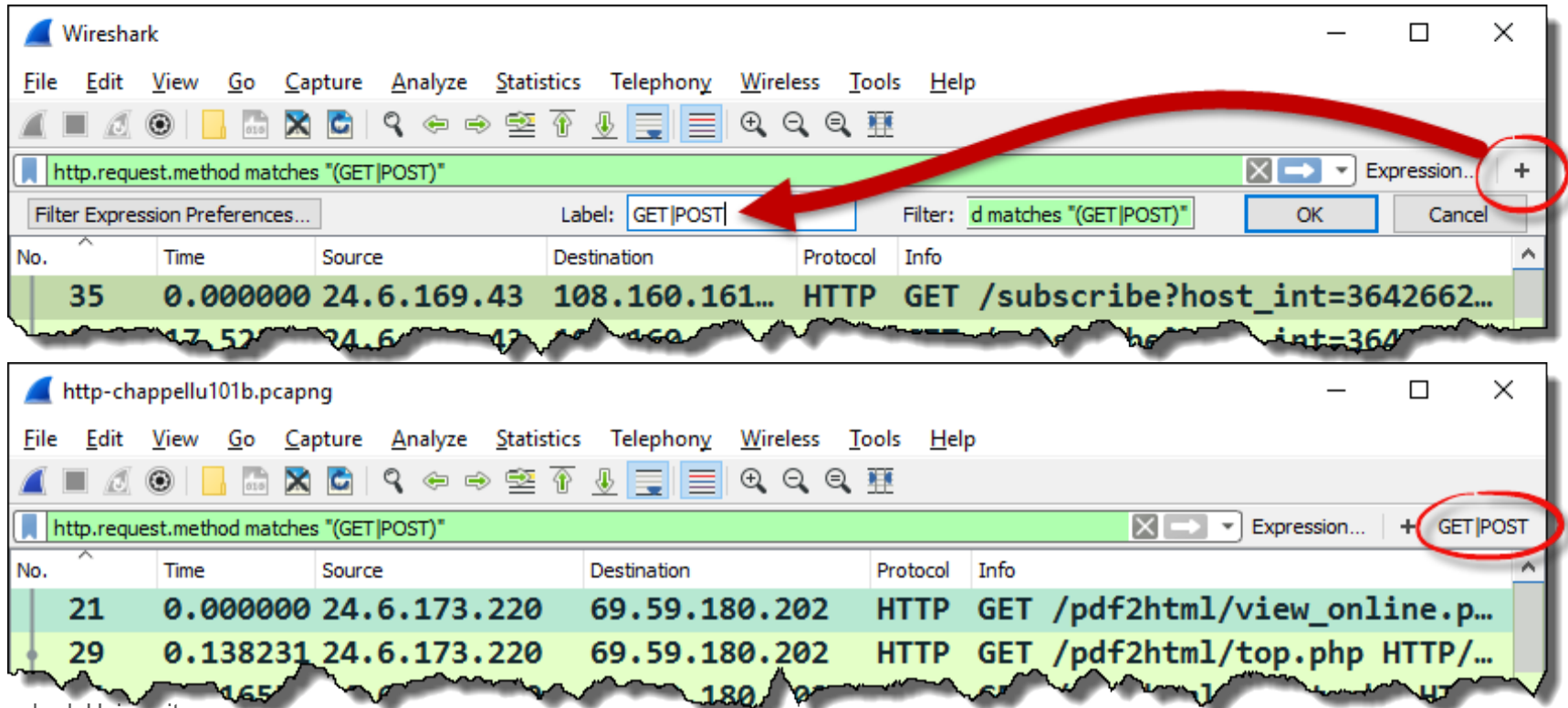
☐ Name resolution ☐ Limit to display filter

Conversation Types

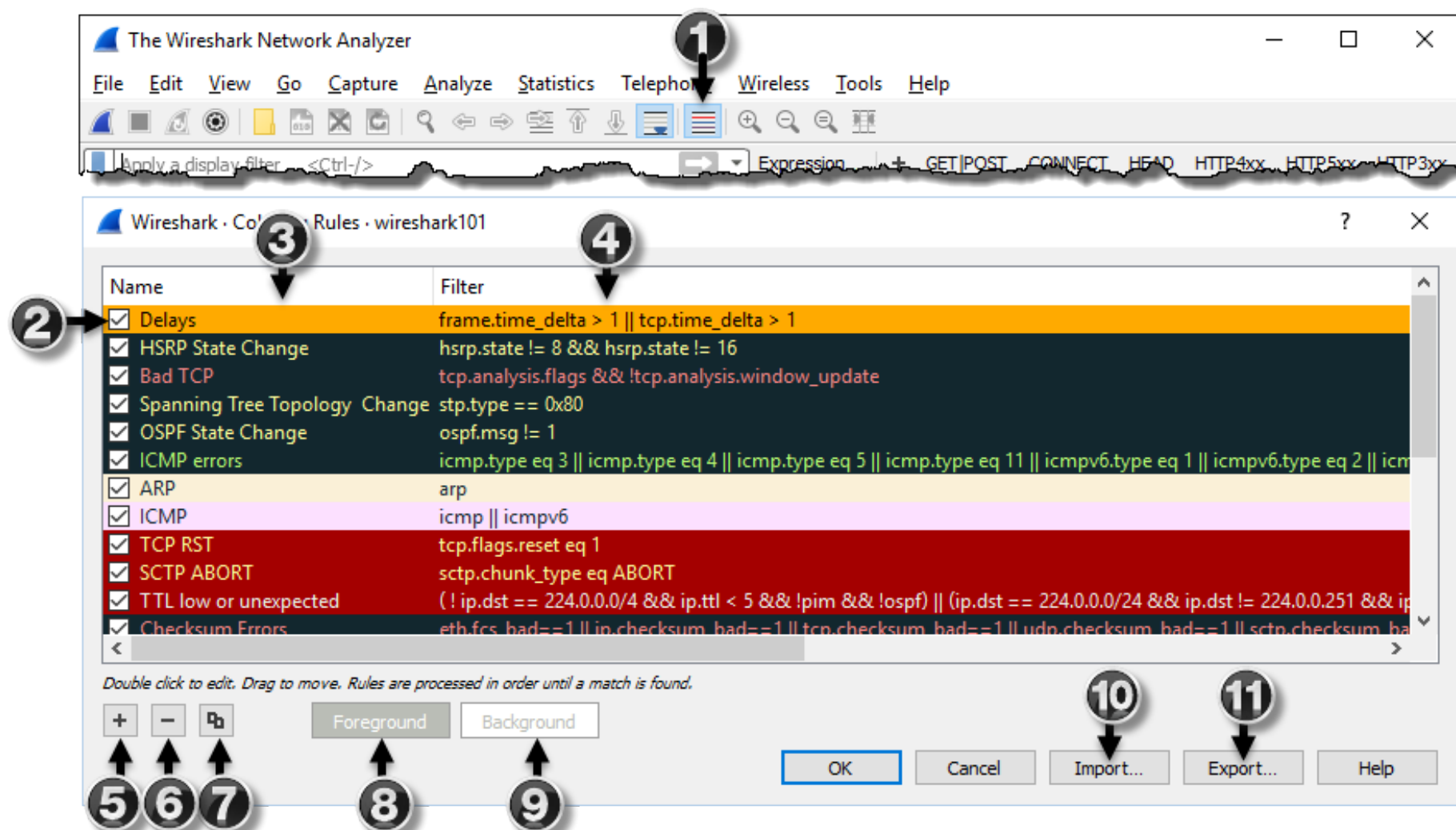
Copy Follow Stream... Graph... Close Help

Turn Your Key Display Filters into Buttons

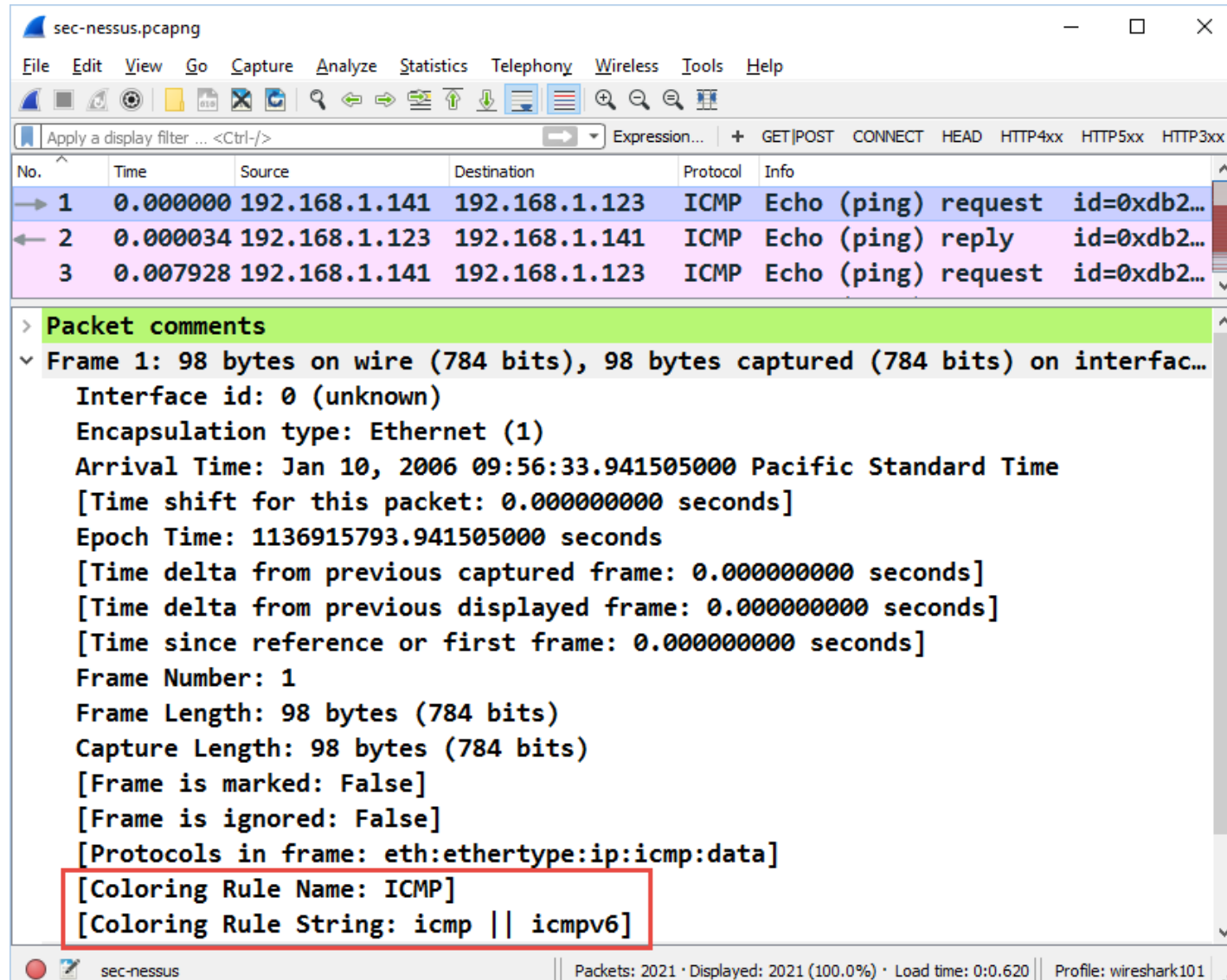
1. Create a display filter and click .
2. Name your Filter Expression button.
(Reorder/edit/disable or delete in **Preferences | Filter Expression.**)



Coloring Rules Interface



Identify Applied Coloring Rules



The image shows a Wireshark window titled "sec-nessus.pcapng". The packet list shows three ICMP Echo (ping) requests. The details pane for the first packet (Frame 1) is expanded, showing various metadata and the applied coloring rule.

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.1.141	192.168.1.123	ICMP	Echo (ping) request id=0xdb2...
2	0.000034	192.168.1.123	192.168.1.141	ICMP	Echo (ping) reply id=0xdb2...
3	0.007928	192.168.1.141	192.168.1.123	ICMP	Echo (ping) request id=0xdb2...

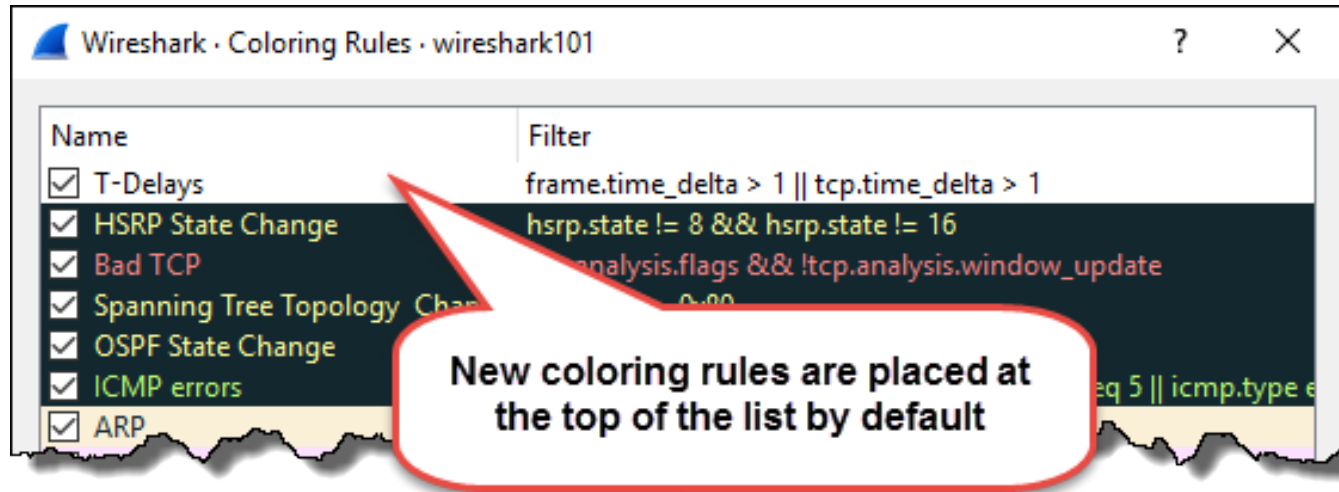
Packet comments

- Frame 1: 98 bytes on wire (784 bits), 98 bytes captured (784 bits) on interfac...
 - Interface id: 0 (unknown)
 - Encapsulation type: Ethernet (1)
 - Arrival Time: Jan 10, 2006 09:56:33.941505000 Pacific Standard Time
 - [Time shift for this packet: 0.000000000 seconds]
 - Epoch Time: 1136915793.941505000 seconds
 - [Time delta from previous captured frame: 0.000000000 seconds]
 - [Time delta from previous displayed frame: 0.000000000 seconds]
 - [Time since reference or first frame: 0.000000000 seconds]
 - Frame Number: 1
 - Frame Length: 98 bytes (784 bits)
 - Capture Length: 98 bytes (784 bits)
 - [Frame is marked: False]
 - [Frame is ignored: False]
 - [Protocols in frame: eth:ethertype:ip:icmp:data]
 - [Coloring Rule Name: ICMP]
 - [Coloring Rule String: icmp || icmpv6]

sec-nessus | Packets: 2021 · Displayed: 2021 (100.0%) · Load time: 0:0.620 | Profile: wireshark101

Build a Coloring Rule to Highlight Delays

```
frame.time_delta > 1 || tcp.time_delta > 1
```



Master the Intelligent Scrollbar

ftp-bounce.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

Apply a display filter ... <Ctrl-/> Expression... GET|POST CONNECT HEAD HTTP4xx HTTP5xx HTTP3xx

No.	Time	Source	Destination	Protocol	Info
30	0.201725	204.181.64.2	24.6.103.134	TCP	64712 → 21 [ACK] Seq=63 Ack=322 ...
31	7.209497	204.181.64.2	24.6.103.134	FTP	Request: TYPE I
32	0.040956	24.6.103.134	204.181.64.2	FTP	Response: 200 Type set to I.
33	0.097859	204.181.64.2	24.6.103.134	FTP	Request: PASV
34	0.126899	24.6.103.134	204.181.64.2	FTP	Response: 227 Entering Passive Mode (192, 168, 1, 100)
35	0.090889	204.181.64.2	24.6.103.134	TCP	64712 → 21 [ACK] Seq=63 Ack=322 ...
36	0.000165	24.6.103.134	204.181.64.2	TCP	1303 → 64444 [SYN, ACK] Seq=1303 Win=0 Len=0
37	0.089448	204.181.64.2	24.6.103.134	TCP	64444 → 1303 [ACK] Seq=1 Ack=1303
38	0.001690	204.181.64.2	24.6.103.134	FTP	Request: STOR in design template...
39	0.133418	24.6.103.134	204.181.64.2	TCP	21 → 64712 [ACK] Seq=21 Ack=64712
40	0.002159	24.6.103.134	204.181.64.2	FTP	Request: PASV
41	0.057583	24.6.103.134	204.181.64.2	TCP	1303 → 64444 [FIN, ACK] Seq=1303 Win=0 Len=0
42	0.000090	24.6.103.134	204.181.64.2	FTP	Response: 425 Error: Possible boot...
43	0.051747	204.181.64.2	24.6.103.134	FTP...	FTP Data: 512 bytes
44	0.000074	24.6.103.134	204.181.64.2	TCP	1303 → 64444 [RST] Seq=2 Win=0 Len=0
45	0.002597	204.181.64.2	24.6.103.134	FTP...	FTP Data: 512 bytes

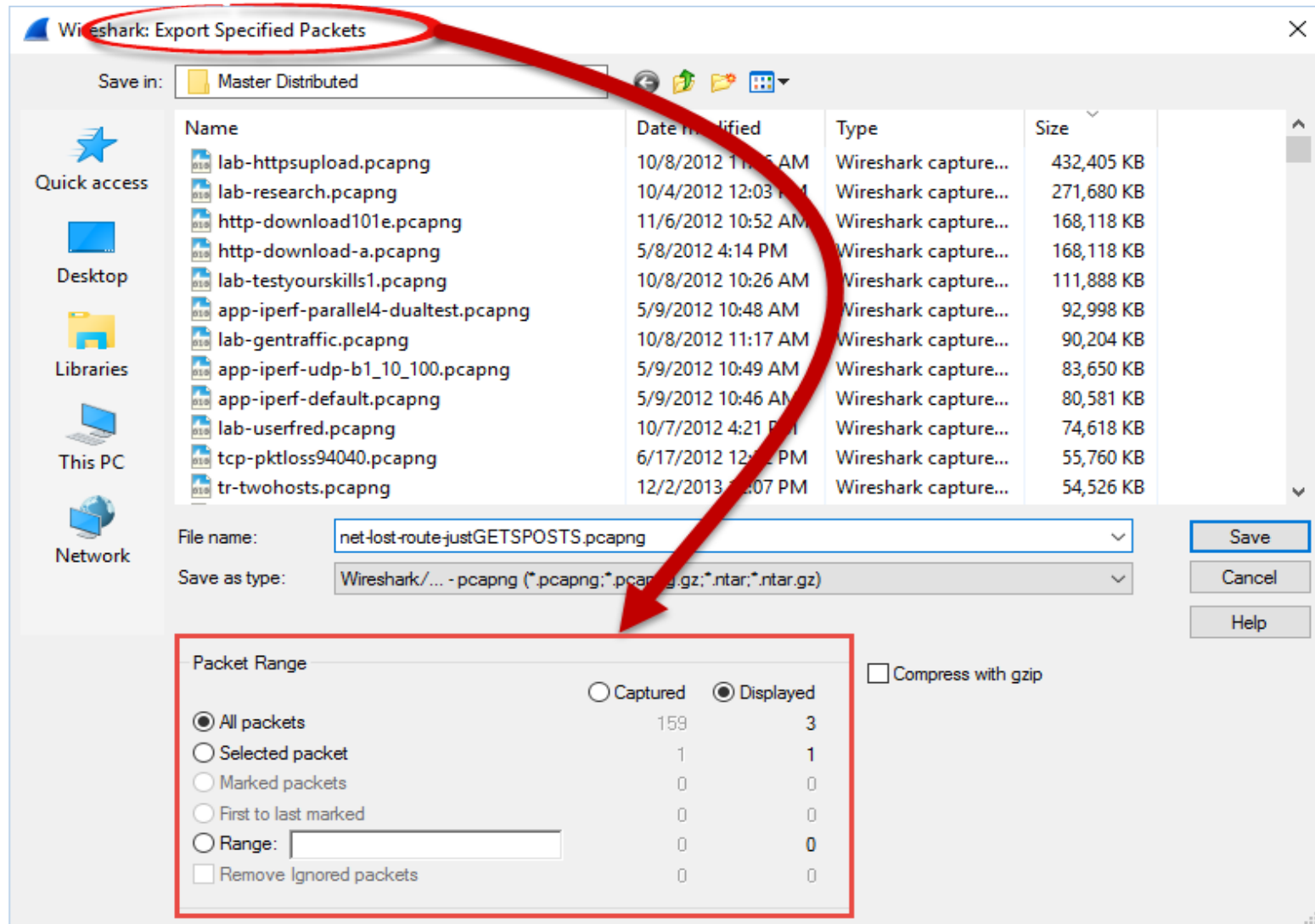
Orange stripe is an FTP command - custom coloring rule

Red stripes are TCP Resets - a default coloring rule

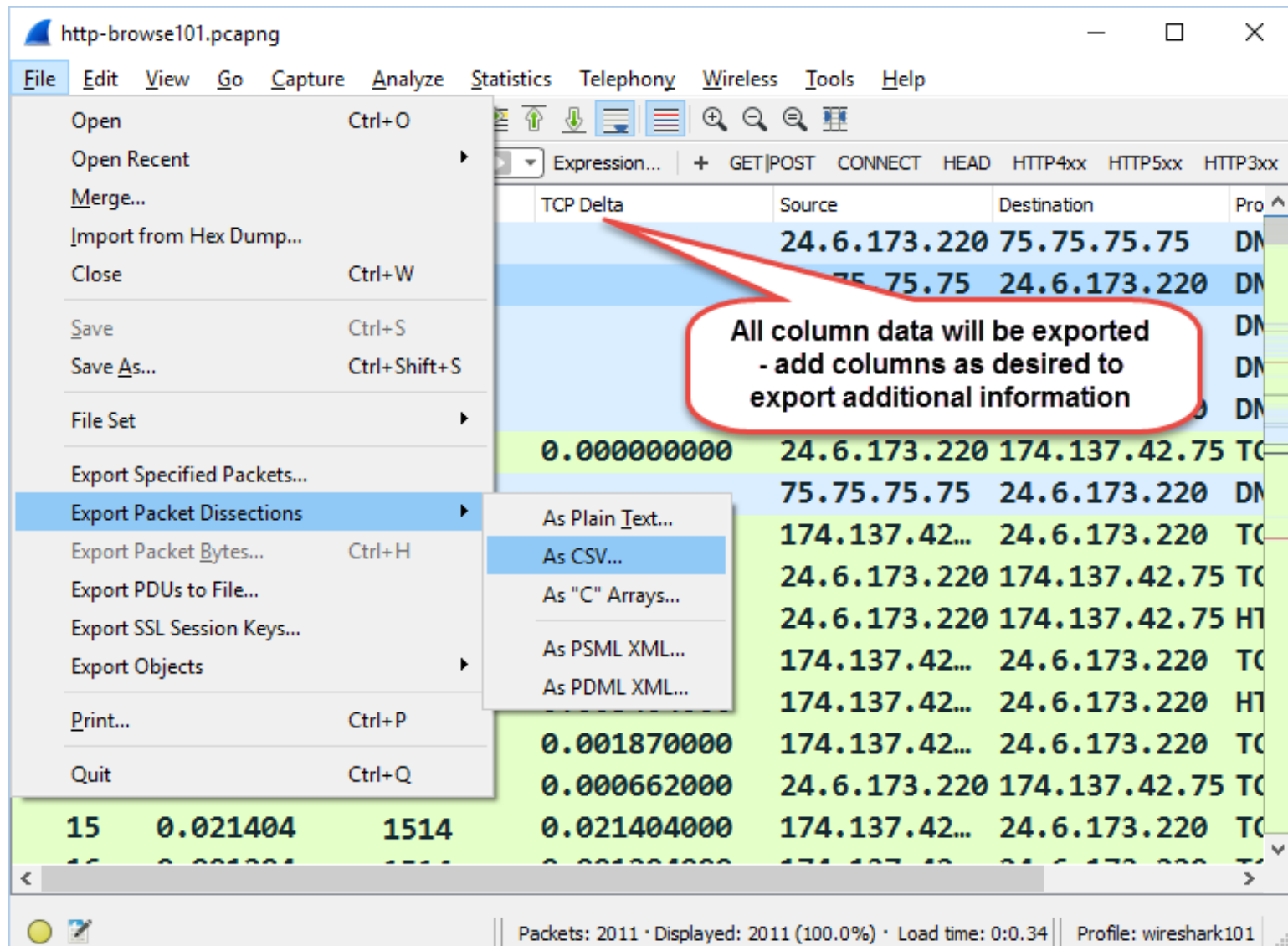
Thumb

Packets: 53 · Displayed: 53 (100.0%) · Load time: 0:0.5 | Profile: wireshark101

Export Packets that Interest You



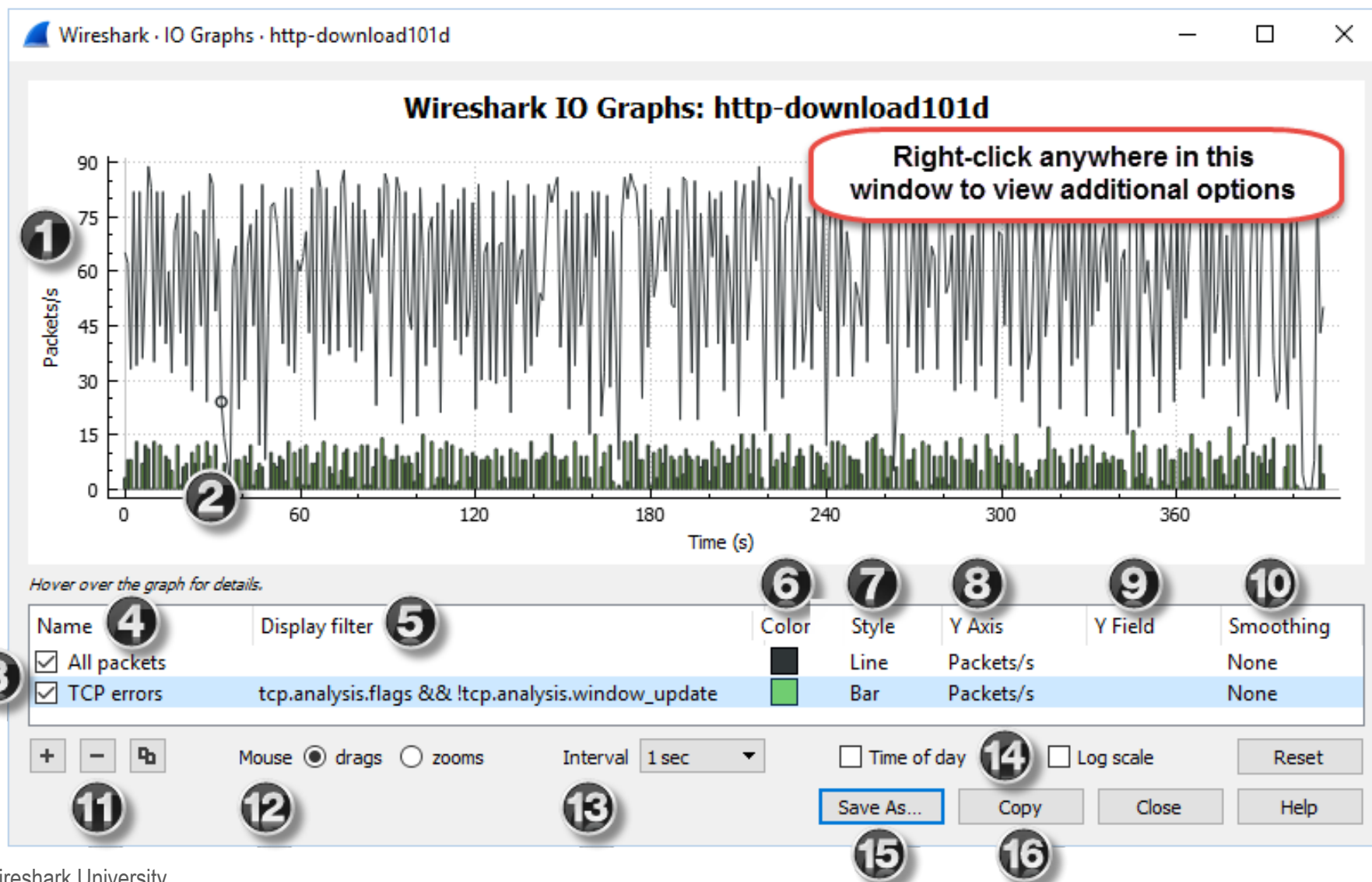
Export Packet Details



Section 5 Skills

**Build and Interpret
Charts and Graphs**

IO Graph Interface



Find Out Who's Talking to Whom

Wireshark · Conversations · http-espn101

Ethernet · 1 IPv4 · 37 IPv6 TCP · 63 UDP · 82

Address A	Address B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B → A	Rel Start
24.6.173.220	184.84.222.88	1,855	2020 k	533	30 k	1322	1989 k	9.726479000
24.6.173.220	184.84.222.48	720	649 k	265	43 k	455	605 k	0.322923000
24.6.173.220	184.84.222.120	613	628 k	195	14 k	418	614 k	6.716176000
24.6.173.220	184.84.222.10	371	382 k	119	7502	252	374 k	7.950911000
24.6.173.220	184.84.222.152	303	286 k	110	25 k	193	261 k	3.261301000
24.6.173.220	68.71.216.176	127	134 k	38	7147	89	127 k	0.168701000
24.6.173.220	74.125.224.59	142	115 k	51	9643	91	105 k	2.843065000
24.6.173.220	184.84.222.16	41	36 k	15	1768	26	35 k	7.951909000
24.6.173.220	184.84.222.75	36	33 k	12	1602	24	32 k	5.377013000
24.6.173.220	138.108.7.20	31	24 k	11	1675	20	23 k	5.436636000
24.6.173.220	68.71.216.171	29	24 k	12	1007	17	23 k	5.192672000
24.6.173.220	75.75.75.75	180	22 k	90	6973	90	15 k	0.000000000
24.6.173.220	68.71.216.157	132	20 k	66	3672	66	16 k	21.802866000
24.6.173.220	184.84.222.137	30	19 k	14	1638	16	17 k	3.270647000

☐ Name resolution ☐ Limit to display filter **Conversation Types**

Copy Follow Stream... Graph... Close Help

Locate the Top Talkers

Wireshark · Conversations · http-misctraffic101

Ethernet · 1 IPv4 · 2 IPv6 TCP · 7 UDP

Address A	Port A	Address B	Port B	Packets	Bytes	Packets A → B	Bytes A → B	Packets B → A	Bytes B
24.6.181.160	1266	107.6.133.250	80	475	533 k	126	8261	349	
24.6.181.160	1260	208.118.237.137	80	127	133 k	37	3086	90	
24.6.181.160	1264	208.118.237.137	80	40	36 k	14	1705	26	
24.6.181.160	1261	208.118.237.137	80	10	2141	5	1174	5	
24.6.181.160	1263	208.118.237.137	80	10	2012	5	1175	5	
24.6.181.160	1262	208.118.237.137	80	10	2011	5	1174	5	
24.6.181.160	1265	208.118.237.137	80	10	1821	5	1169	5	

☐ Name resolution ☐ Limit to display filter Conversation Types

Copy Follow Stream... Graph... Close Help

List Active Applications

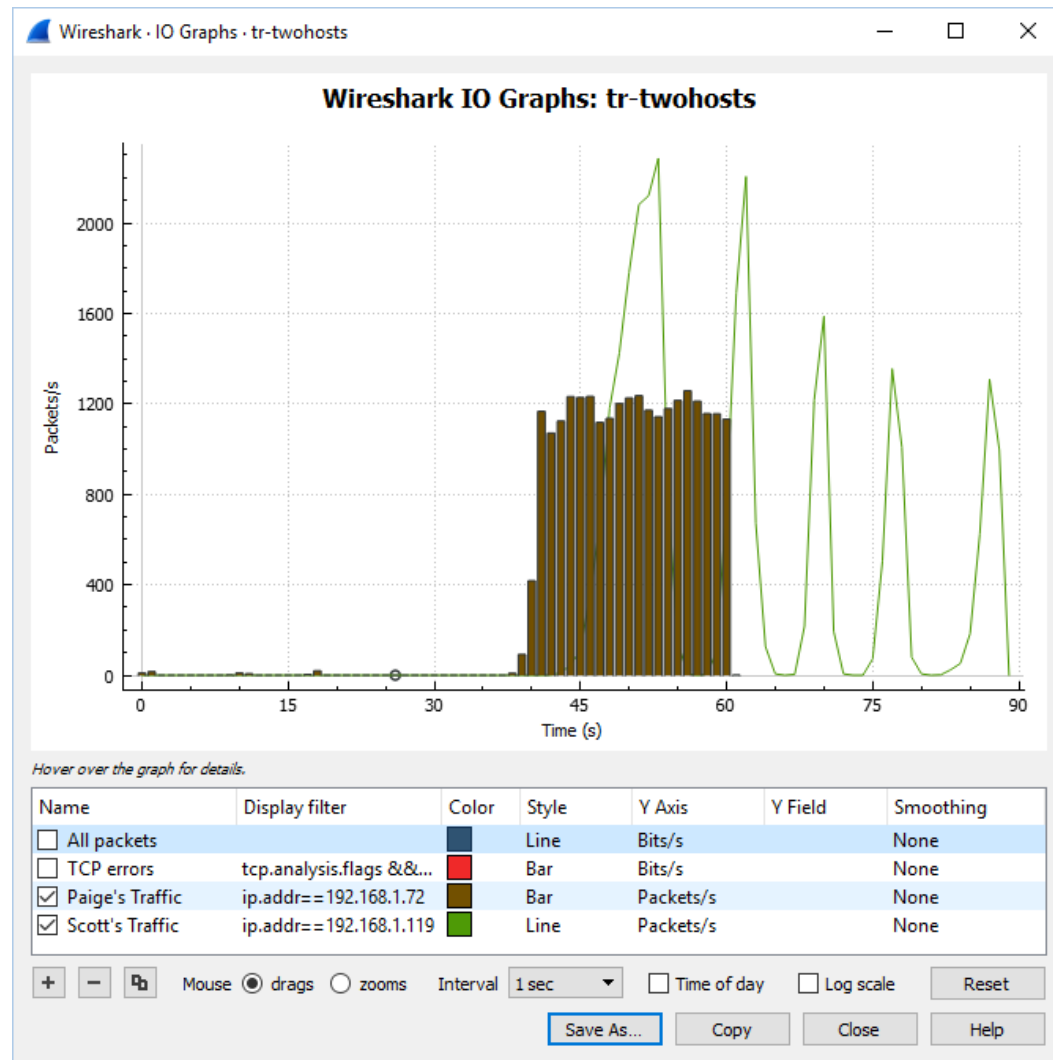
Wireshark · Protocol Hierarchy Statistics · http-browse101b

Protocol	Percent Packets	Packets	Percent Bytes	Bytes	Bits/s	End Packets	End Bytes	End Bits/s
▼ Frame	100.0	195	100.0	107708	108 k	0	0	0
▼ Ethernet	100.0	195	100.0	107708	108 k	0	0	0
▼ Internet Protocol Version 6	8.2	16	1.9	2062	2074	0	0	0
▼ User Datagram Protocol	8.2	16	1.9	2062	2074	0	0	0
Domain Name System	8.2	16	1.9	2062	2074	16	2062	2074
▼ Internet Protocol Version 4	91.8	179	98.1	105646	106 k	0	0	0
▼ Transmission Control Protocol	91.8	179	98.1	105646	106 k	114	59681	60 k
▼ Hypertext Transfer Protocol	33.3	65	42.7	45965	46 k	33	13364	13 k
Portable Network Graphics	0.5	1	1.1	1229	1236	1	1229	1236
Media Type	0.5	1	1.4	1514	1523	1	1514	1523
Line-based text data	2.1	4	4.8	5176	5207	4	5176	5207
JPEG File Interchange Format	2.1	4	5.6	6056	6092	4	6056	6092
▼ Compuserve GIF	11.3	22	17.3	18626	18 k	16	9542	9599
Unreassembled Fragmented Packet	3.1	6	8.4	9084	9139	6	9084	9139

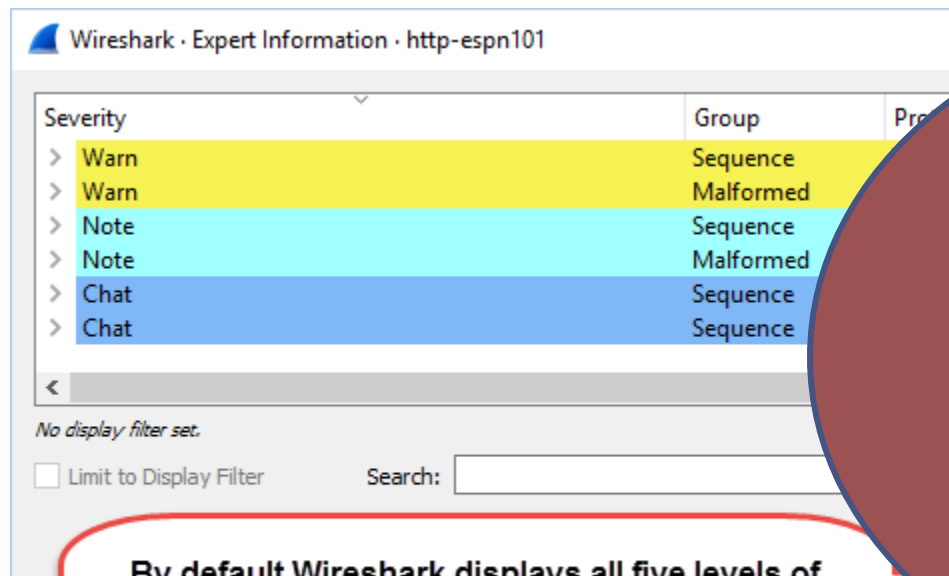
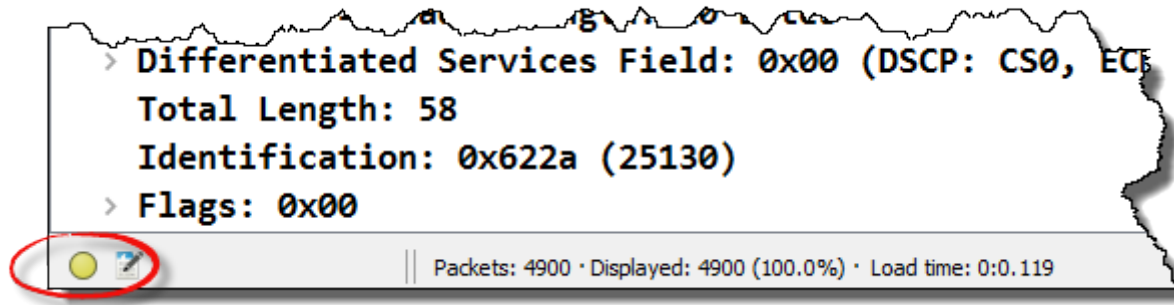
No display filter

Close Copy Help

Graph Application and Host Bandwidth Usage



Identify TCP Errors on the Network



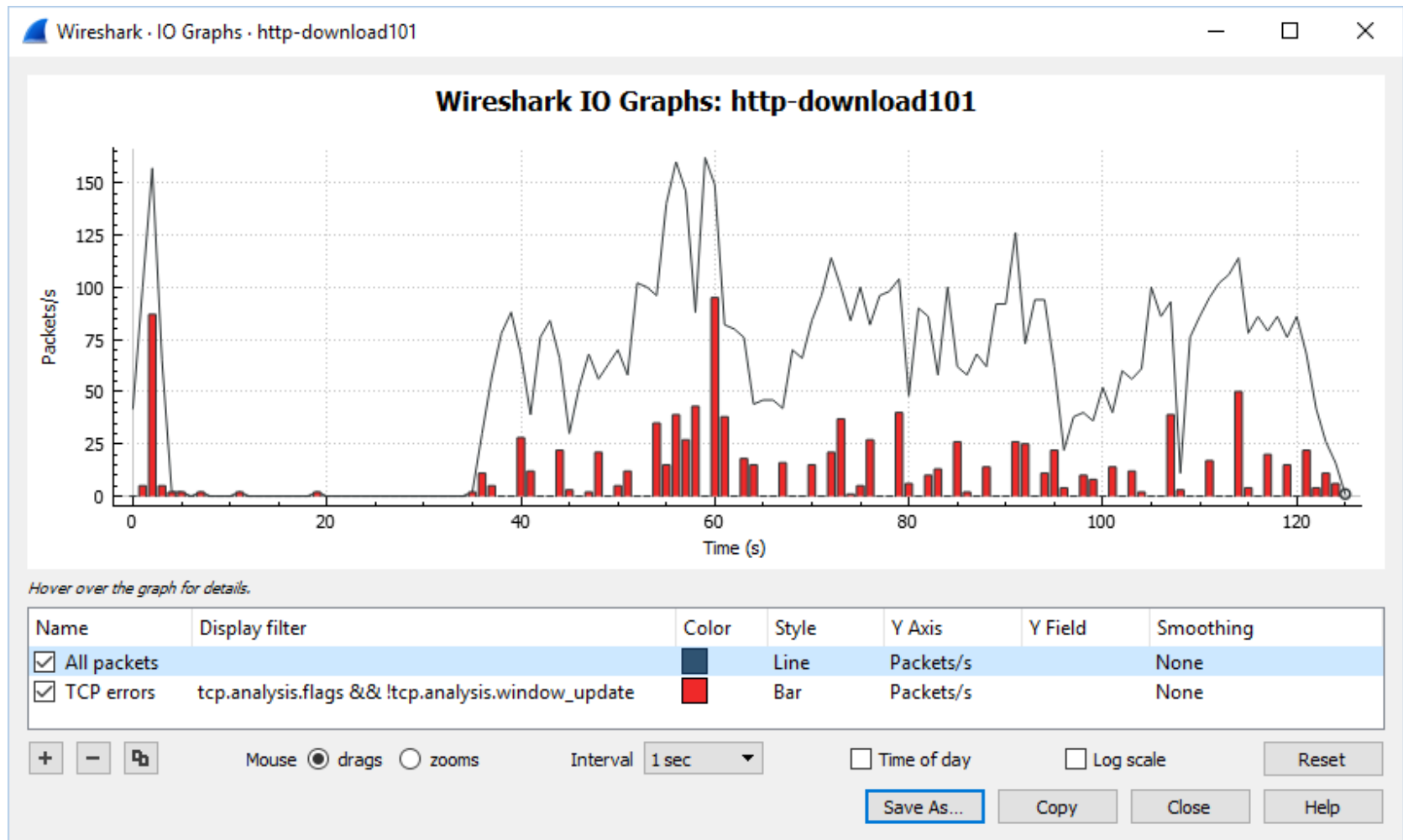
By default Wireshark displays all five levels of severity - in this case there are no Error or Comment items to show

Hoping this is going to change

Understand what the Expert Infos Errors Mean

- Packet Loss, Recovery, and Faulty Trace Files
- Asynchronous or Multiple Path Indications
- Keep-Alive Indication
- Receive Buffer Congestion Indications
- TCP Connection Port Reuse Indication
- Possible Router Problem Indication
- Misconfiguration or ARP Poisoning Indication

Graph Various Network Errors



File and Object Reassembly Options

The image shows the Wireshark interface with the 'File' menu open and a right-click context menu over a packet list. The packet list shows several TCP and HTTP packets. The 'File' menu has 'Export Objects' highlighted, which has opened a sub-menu with 'HTTP...' selected (marked with a '1'). The right-click context menu has 'Follow' selected (marked with a '2'), which has opened a sub-menu with 'TCP Stream' selected (marked with a '3'). The 'Follow' sub-menu also shows 'UDP Stream' (marked with a '3') and 'SSL Stream' (marked with a '4').

File Menu:

- Open (Ctrl+O)
- Open Recent
- Merge...
- Import from Hex Dump...
- Close (Ctrl+W)
- Save (Ctrl+S)
- Save As... (Ctrl+Shift+S)
- File Set
- Export Specified Packets...
- Export Packet Dissections
- Export Packet Bytes... (Ctrl+H)
- Export PDUs to File...
- Export SSL Session Keys...
- Export Objects**
- Print... (Ctrl+P)
- Quit (Ctrl+Q)

Export Objects Sub-menu:

- DICOM...
- HTTP...** (1)
- SMB...
- IFTF

Right-click Context Menu:

- Mark/Unmark Packet (Ctrl+M)
- Ignore/Unignore Packet (Ctrl+D)
- Set/Unset Time Reference (Ctrl+T)
- Time Shift... (Ctrl+Shift+T)
- Packet Comment...
- Edit Resolved Name
- Apply as Filter
- Prepare a Filter
- Conversation Filter
- Colorize Conversation
- SCTP
- Follow** (2)
- Copy
- Protocol Preferences
- Decode As...
- Show Packet in New Window

Follow Sub-menu:

- TCP Stream** (3)
- UDP Stream (3)
- SSL Stream (4)

Packet List:

Destination	Protocol	Info
208.111.1...	TCP	[ACK] Seq=1 Ack=1 Win=65...
208.111.1...	HTTP	sets/314da08c2cc0c65e47e...
24.6.173...	TCP	Ack=1 W...
24.6.173...	TCP	Ack=1 W...
24.6.173...	TCP	Ack=1 W...
208.111.1...	TCP	:1 Win=65...
208.111.1...	TCP	:1 Win=65...
208.111.1...	TCP	:1 Win=65...
208.111.1...	HTTP	03a791286...
208.111.1...	HTTP	ed131030

Packet Details:

.... 0101 = Header Length: 20 bytes

> Differentiated Services Field: 0x00 (DSC)

Total Length: 60

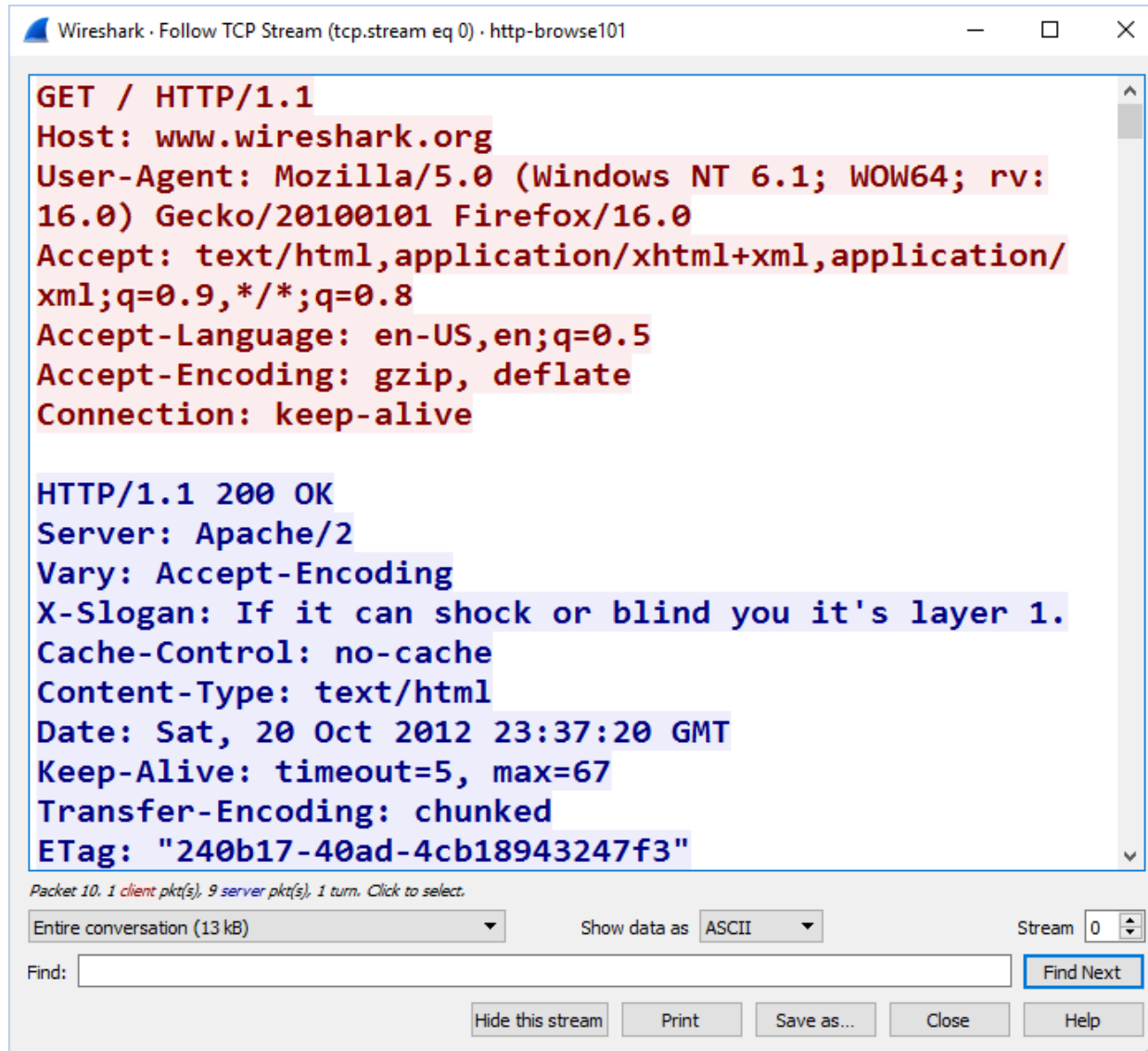
Identification: 0x697e (27006)

> Flags: 0x00

Status Bar: Packets: 6143 • Displayed: 6143 (100.0%) • Load time: 0:0.124 Profile: wireshark101

Reassemble Web Browsing Sessions

p browse101.pcapng



Reassemble a File Transferred via FTP

The screenshot shows the Wireshark network protocol analyzer interface. The main packet list displays several packets, with packet 81 selected. A right-click context menu is open over packet 81, and the 'Follow' option is selected, opening a submenu where 'TCP Stream' is highlighted. The packet details pane shows the structure of packet 81: Frame 81: 1514 bytes on wire (12112 bits), Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol, Src Port: FTP Data (1460 bytes data).

No.	Time	Source	Destination	Protocol	Info
79	0.000944	24.6.173.220	131.246.123.4	FTP	Request: RETR /pub/wireshar...
80	0.210322	131.246.123.4	24.6.173.220	FTP	Response: 150 Opening BINAR...
81	0.026085	131.246.123.4	24.6.173.220	FTP	Data: 1460 bytes
82	0.000984	131.246.123.4	24.6.173.220	FTP	
83	0.000004	131.246.123.4	24.6.173.220	FTP	
84	0.000173	24.6.173.220	131.246.123.4	FTP	
85	0.167047	24.6.173.220	131.246.123.4	FTP	
86	0.044281	131.246.123.4	24.6.173.220	FTP	
87	0.000981	131.246.123.4	24.6.173.220	FTP	
88	0.000002	131.246.123.4	24.6.173.220	FTP	

Frame 81: 1514 bytes on wire (12112 bits)
 Ethernet II, Src: Cadant_31:bb:c1 (00:01:
 Internet Protocol Version 4, Src: 131.246
 Transmission Control Protocol, Src Port:
 FTP Data (1460 bytes data)

right-click

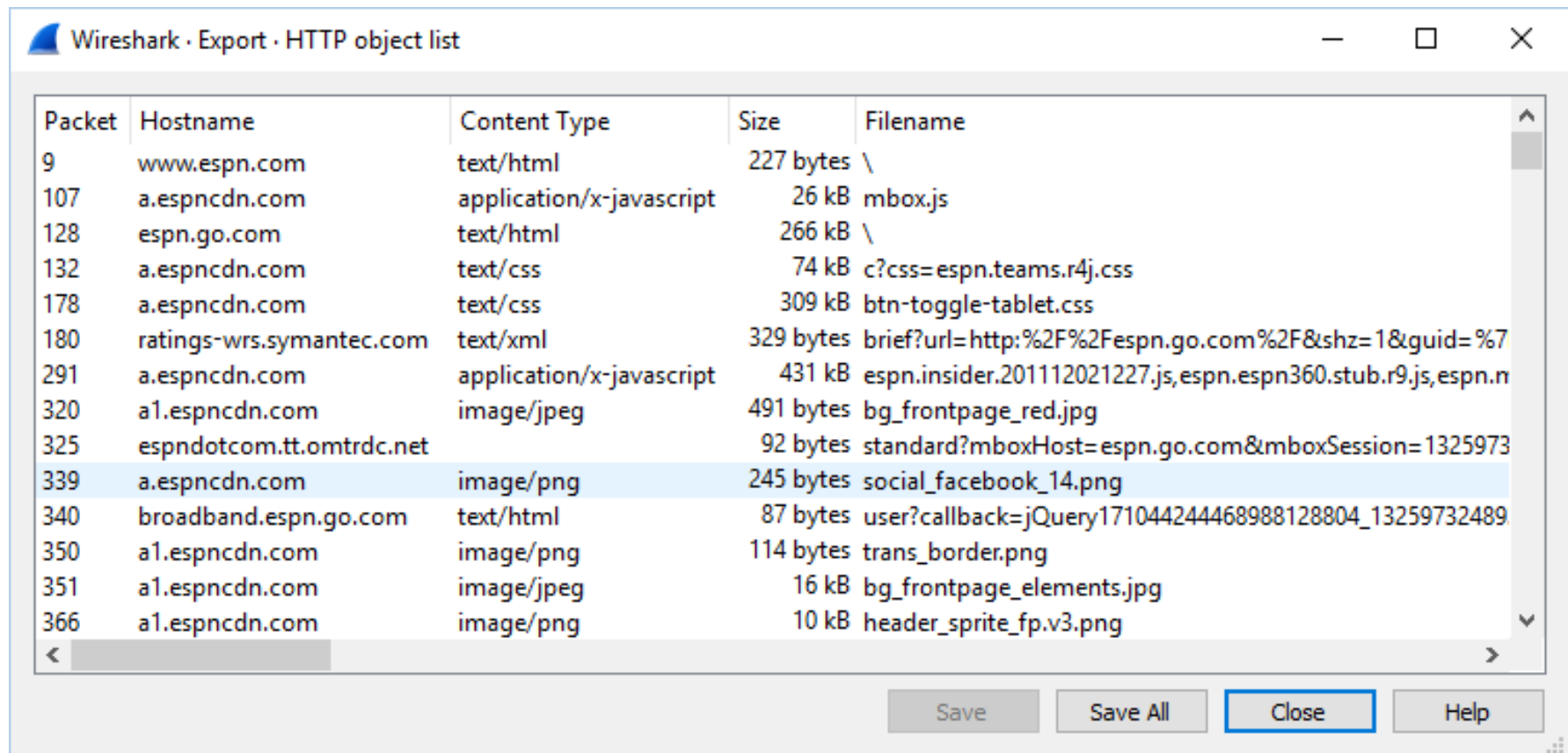
- Mark/Unmark Packet Ctrl+M
- Ignore/Unignore Packet Ctrl+D
- Set/Unset Time Reference Ctrl+T
- Time Shift... Ctrl+Shift+T
- Packet Comment...
- Edit Resolved Name
- Apply as Filter
- Prepare a Filter
- Conversation Filter
- Colorize Conversation
- SCTP
- Follow
- Copy
- Protocol Preferences
- Decode As...
- Show Packet in New Window

bits) on i...
 a7:bf:a3 (d...
 66 (64666)...
 TCP Stream
 UDP Stream
 SSL Stream

Packets: 22479 · Displayed: 22479 (100.0%) · Load time: 0:0.273 | Profile: wireshark101

Export HTTP Objects Transferred in a Web Browsing Session

Enable *Allow subdissector to reassemble TCP stream* (TCP preference).



File and Packet Annotation Options

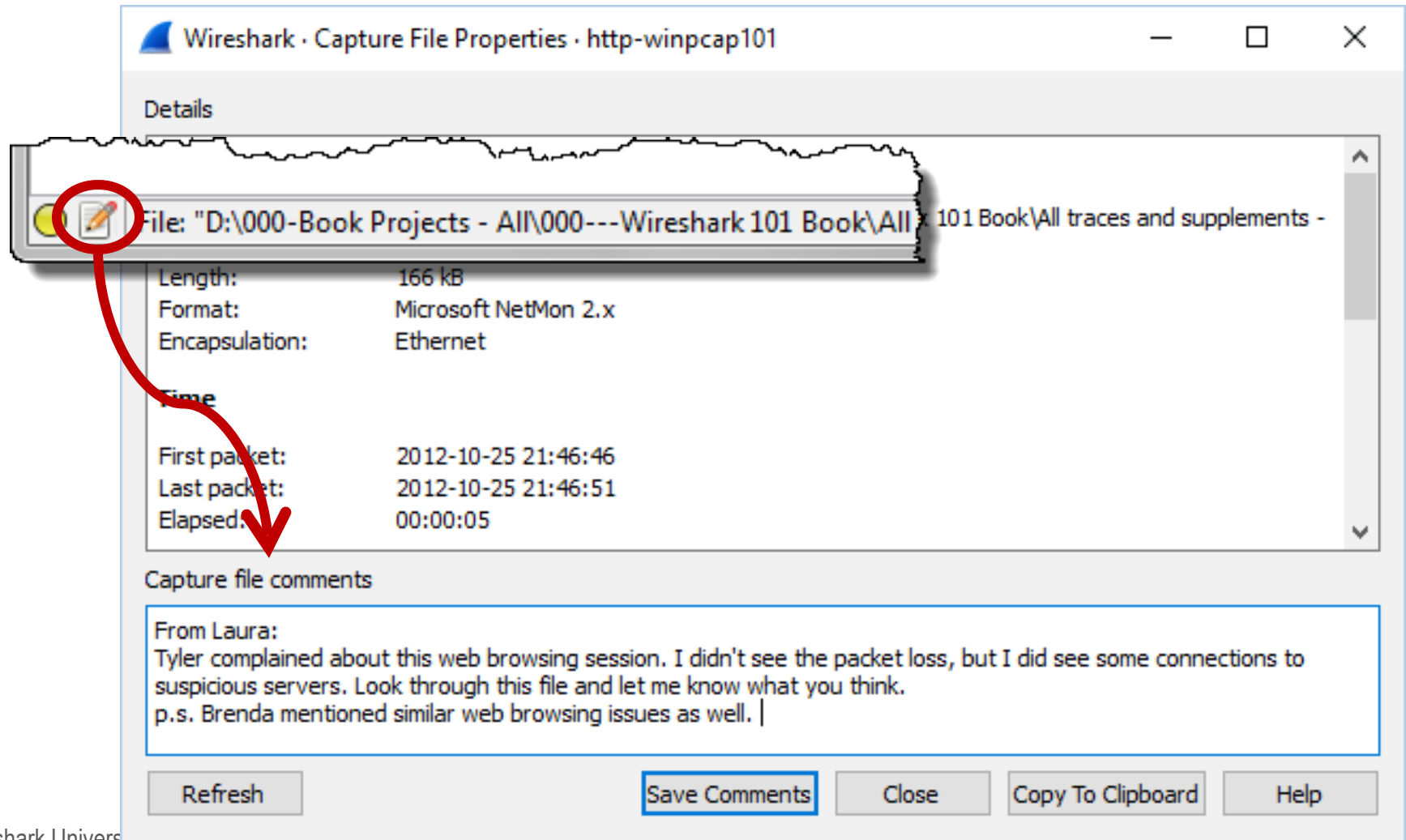
The screenshot illustrates the process of adding a packet comment in Wireshark. The interface is divided into several panes:

- Top Pane:** Shows the packet list. Packet 8 is selected, and a right-click context menu is open. The menu includes options like 'Mark/Unmark Packet', 'Ignore/Unignore Packet', and 'Packet Comment...'. A red box highlights the 'right-click' action, and a circled '8' marks the 'Packet Comment...' option.
- Bottom Pane:** Displays the packet details for the selected packet. It shows the 'Packet comments' section with a list of comments. A circled '4' marks the first comment: 'This packet shows the'. Below this, the packet details for the selected packet are shown, including 'Source Port: 6413', 'Destination Port: 80', 'Sequence number: 1', and 'Acknowledgment number:'. A circled '5' marks the 'Acknowledgment number:' label, and a circled '6' marks the 'Acknowledgment number:' value.
- Right Pane:** Shows the 'Expert Information' pane. It displays a list of comments with their severity and group. A circled '7' marks the first comment: '8: This packet shows the first GET request to the site.' Below this, the packet details for the selected packet are shown, including 'Source Port: 6413', 'Destination Port: 80', 'Sequence number: 1', and 'Acknowledgment number:'. A circled '5' marks the 'Acknowledgment number:' label, and a circled '6' marks the 'Acknowledgment number:' value.

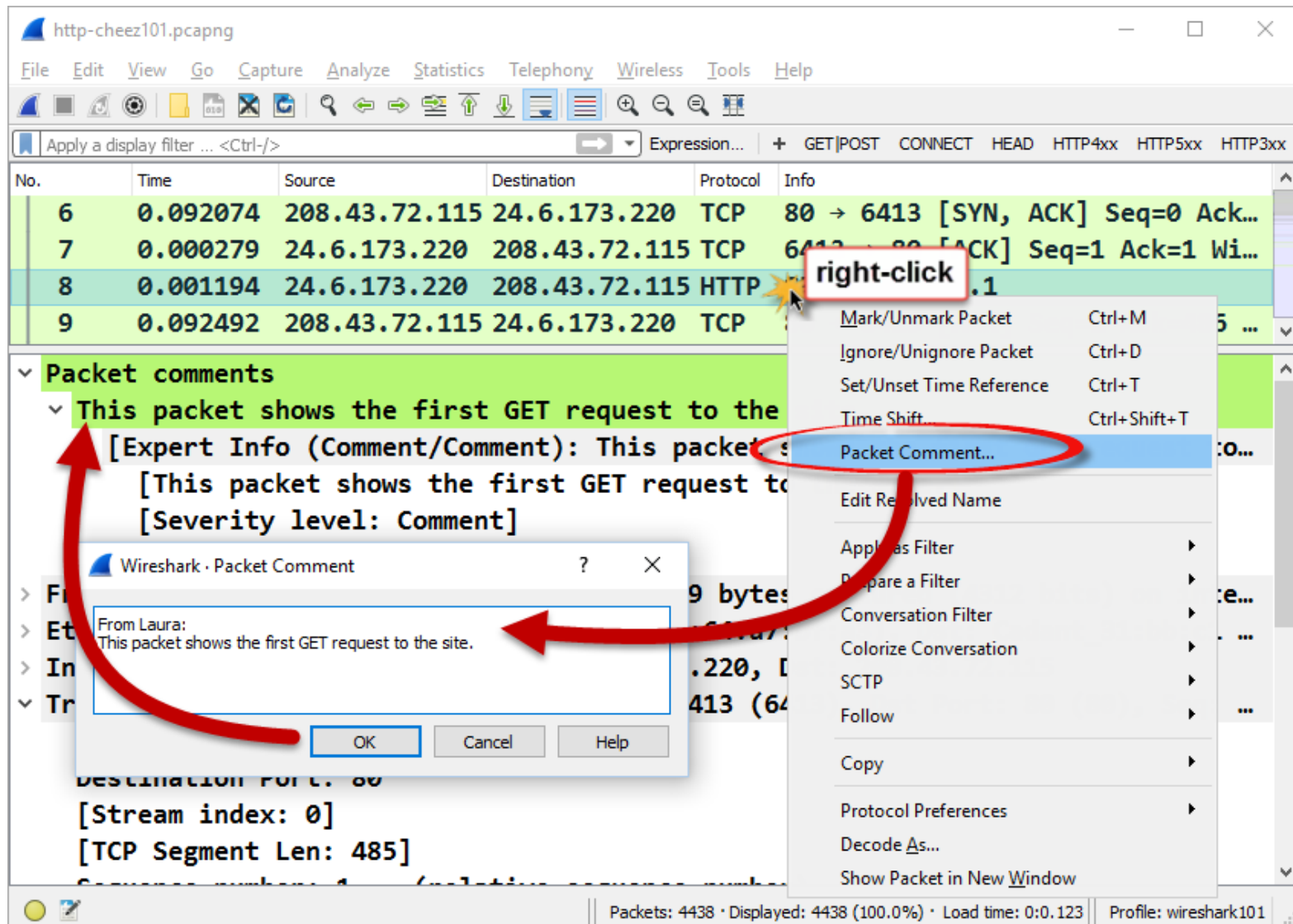
The 'Expert Information' pane also includes a search bar and a 'Show...' button.

Add Your Comments to Trace Files

Only *.pcapng* file format supports comments.



Add Comments to Individual Packets



Export Packet Comments for a Report

The image shows the Wireshark network protocol analyzer interface. The top menu bar includes File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Wireless, Tools, and Help. Below the menu is a toolbar with various icons. The main display area is divided into three panes. The top pane shows a list of captured packets with columns for No., Time, Source, Destination, Protocol, and Comment. The bottom pane shows the details of the selected packet (No. 1), including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The middle pane shows the packet's raw data in hexadecimal and ASCII. The 'pkt_comment' filter is applied to the packet list, and the 'Packet comments' section is expanded, showing a comment for the selected packet.

sec-suspicious101.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

pkt_comment

Expression... + GET|POST CONNECT HEAD HTTP4xx HTTP5xx HTTP3xx

No.	Time	Source	Destination	Protocol	Comment
1	0.000000	24.6.173.220	74.125.224.84	HTTP	This is the original search query for the
5	0.062672	74.125.224.84	24.6.173.220	HTTP	In this response, the server sends numero
7	0.475050	24.6.173.220	74.125.224.84	HTTP	Now we clicked on the image load the expa
12	0.043454	74.125.224.84	24.6.173.220	HTTP	We get the expanded image through Google

Packet comments

- This is the original search query for the "Peter Lik for sale" images.
 - [Expert Info (Comment/Comment): This is the original search query for the "Peter Li...
 - [This is the original search query for the "Peter Lik for sale" images.]
 - [Severity level: Comment]
 - [Group: Comment]

> Frame 1: 1097 bytes on wire (8776 bits), 1097 bytes captured (8776 bits) on interface 0

> Ethernet II, Src: HewlettP_a7:bf:a3 (d4:85:64:a7:bf:a3), Dst: Cadant_31:bb:c1 (00:01:5c...

> Internet Protocol Version 4, Src: 24.6.173.220, Dst: 74.125.224.84

> Transmission Control Protocol, Src Port: 50262 (50262), Dst Port: 80 (80), Seq: 1, Ack:

Packets: 172 · Displayed: 19 (11.0%) · Load time: 0:0.6 | Profile: wireshark101

Command-Line Tools Key Options

EDITCAP

<code>editcap -h</code>	View Editcap parameters.
<code>editcap -i 360 big.pcapng 360secs.pcapng</code>	Split <i>big.pcapng</i> into separate <i>360secs*.pcapng</i> files with up to 360 seconds of traffic in each file.
<code>editcap -c 500 big.pcapng 500pkts.pcapng</code>	Split <i>big.pcapng</i> into separate <i>500pkts*.pcapng</i> files with up to 500 packets in each file.

MERGECAP

<code>mergcap -h</code>	View Mergecap parameters.
<code>mergcap -w merged.pcapng files*.pcapng</code>	Merge <i>files*.pcapng</i> into a single file called <i>merged.pcapng</i> (merge based on packet timestamps).
<code>mergcap -a -w ab.pcapng a.pcapng b.pcapng</code>	Merge <i>a.pcapng</i> and <i>b.pcapng</i> into a single file called <i>ab.pcapng</i> (merge based on the order files are listed).

TSHARK

<code>tshark -h</code>	View Tshark parameters.
<code>tshark -D</code>	List the available capture interfaces that can be used with the <code>-i</code> parameter.
<code>tshark -i2 -f "tcp" -w tcp.pcapng</code>	Capture only TCP-based traffic on interface 2 and save it to <i>tcp.pcapng</i> .
<code>tshark -i1 -Y "ip.addr==10.2.1.1"</code>	Capture all traffic on interface 1, but only display traffic to or from 10.2.1.1.
<code>tshark -r "myfile.pcapng" -Y "http.host contains ".ru" -w myfile-ru.pcapng</code>	Open a trace file called <i>myfile.pcapng</i> and apply a display filter for the value <code>"http.host contains ".ru"</code> in the HTTP host field – save the results to a file called <i>myfile-ru.pcapng</i> .

Split a Large Trace File into a File Set

- Use `capinfos <filename>` to obtain file information first.

Split based on packet count

- `editcap -c 1000 a.pcapng
a1000set.pcapng`

Split based on time (seconds)

- `editcap -i 360 b.pcapng
b360set.pcapng`

Merge Trace Files

List all Mergecap parameters

- `mergecap -h`

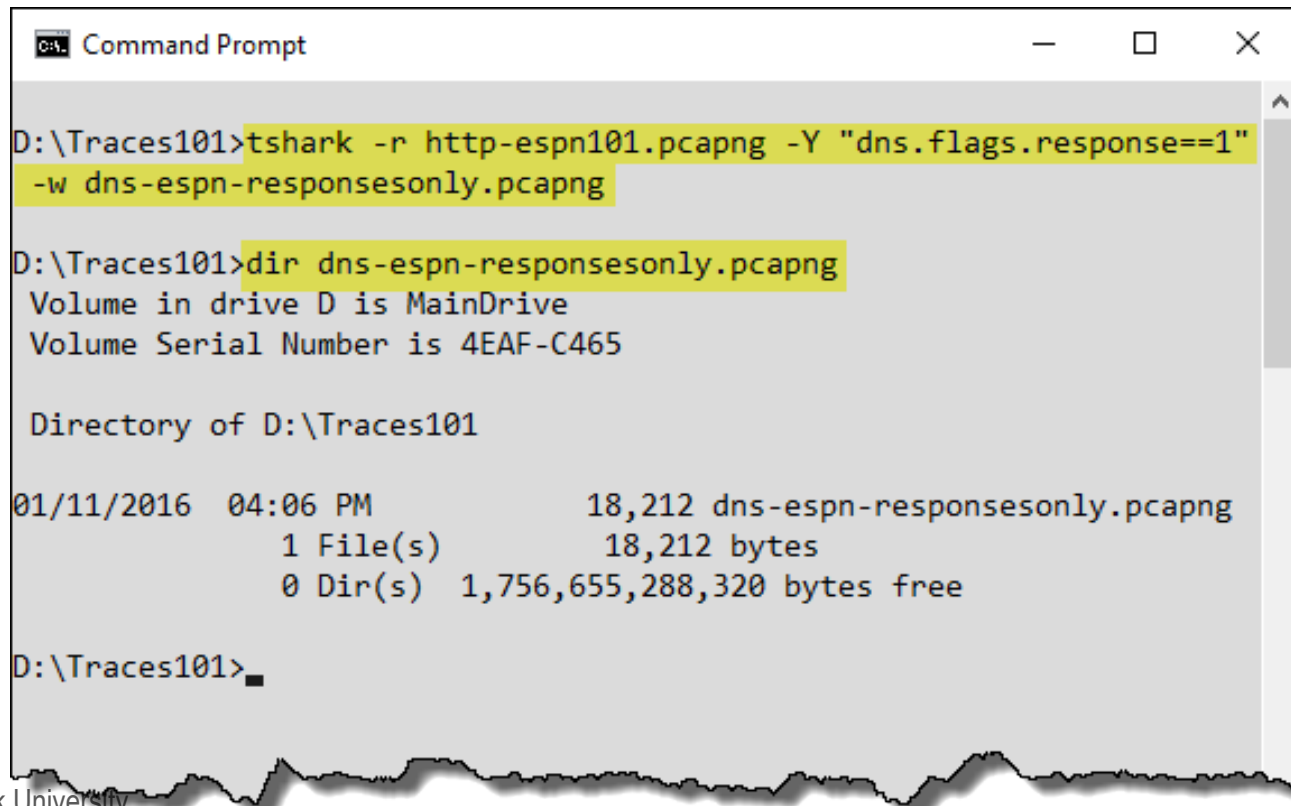
Use Wildcards when Merging

- `mergecap -w c.pcapng c30set*.*`

Capture Traffic at Command Line

Tshark Examples

- `tshark -h`
- `tshark -D`
- `tshark -c 100 -w 100.pcapng`



```
Command Prompt

D:\Traces101>tshark -r http-espn101.pcapng -Y "dns.flags.response==1"
-w dns-espn-responsesonly.pcapng

D:\Traces101>dir dns-espn-responsesonly.pcapng
Volume in drive D is MainDrive
Volume Serial Number is 4EAF-C465

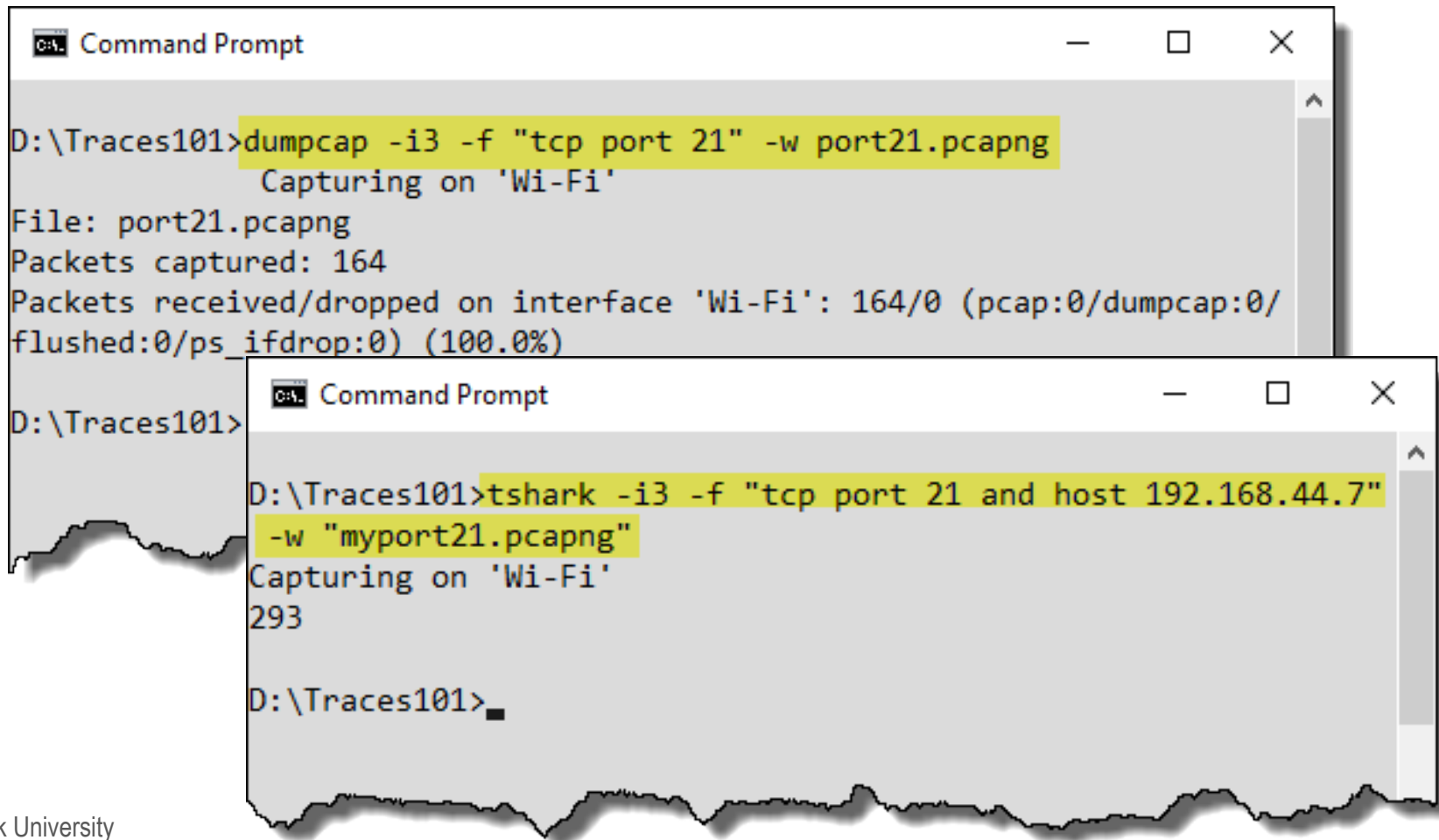
Directory of D:\Traces101

01/11/2016  04:06 PM                18,212 dns-espn-responsesonly.pcapng
               1 File(s)                18,212 bytes
               0 Dir(s)  1,756,655,288,320 bytes free

D:\Traces101>
```

Use Capture Filters during Command-Line Capture

Use the **-f** parameter



The image shows two overlapping Windows Command Prompt windows. The top window displays the execution of the `dumpcap` command with a capture filter. The bottom window displays the execution of the `tshark` command with a capture filter. Both windows show the capture status and statistics.

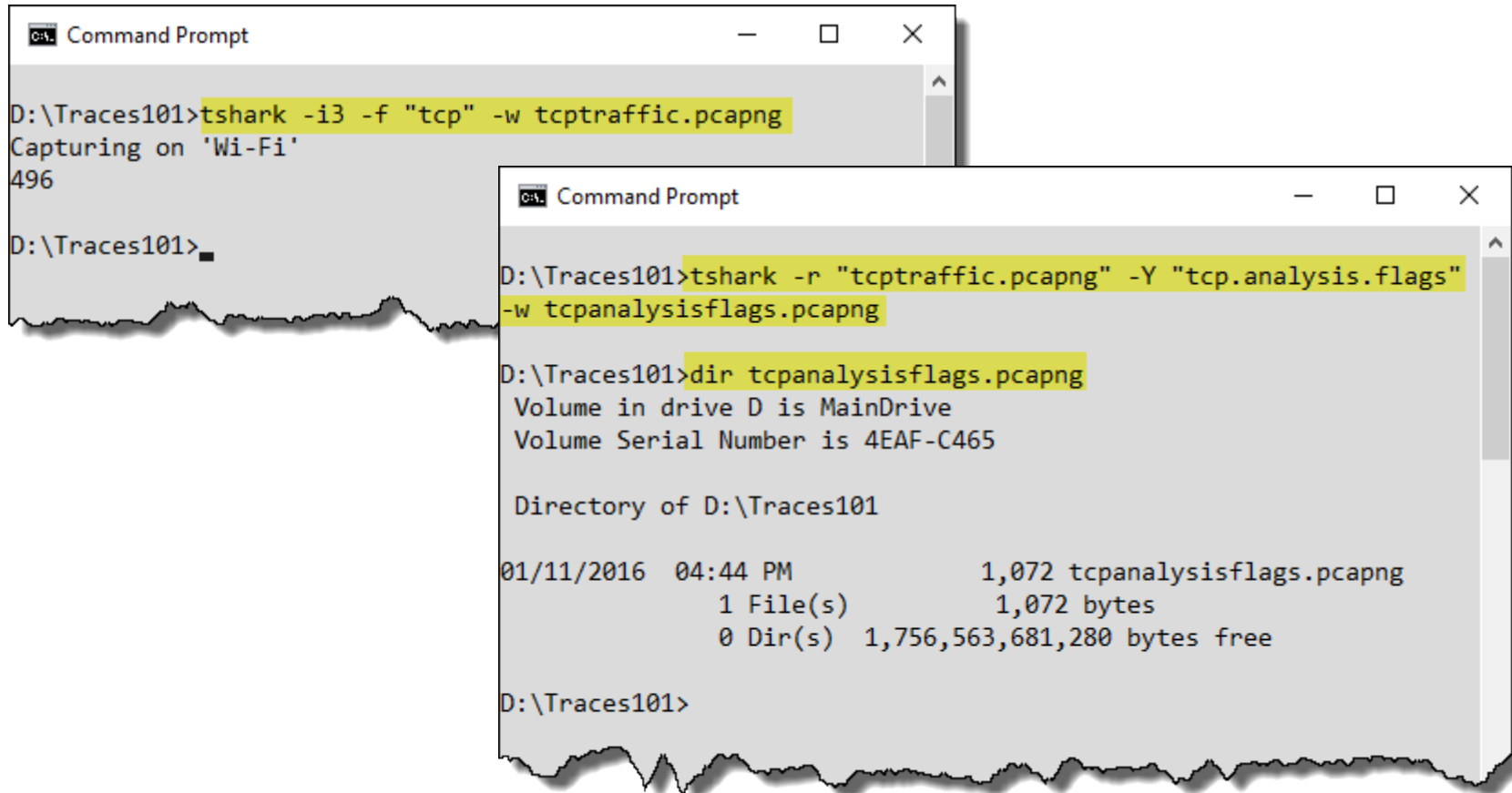
```
D:\Traces101>dumpcap -i3 -f "tcp port 21" -w port21.pcapng
Capturing on 'Wi-Fi'
File: port21.pcapng
Packets captured: 164
Packets received/dropped on interface 'Wi-Fi': 164/0 (pcap:0/dumpcap:0/
flushed:0/ps_ifdrop:0) (100.0%)

D:\Traces101>tshark -i3 -f "tcp port 21 and host 192.168.44.7"
-w "myport21.pcapng"
Capturing on 'Wi-Fi'
293

D:\Traces101>
```

Use Display Filters during Command-Line Capture

Consider a two-step process if you want to capture, apply a display filter, and save the trace file



```
Command Prompt
D:\Traces101>tshark -i3 -f "tcp" -w tcptraffic.pcapng
Capturing on 'Wi-Fi'
496

D:\Traces101>

Command Prompt
D:\Traces101>tshark -r "tcptraffic.pcapng" -Y "tcp.analysis.flags"
-w tcpanalysisflags.pcapng

D:\Traces101>dir tcpanalysisflags.pcapng
Volume in drive D is MainDrive
Volume Serial Number is 4EAF-C465

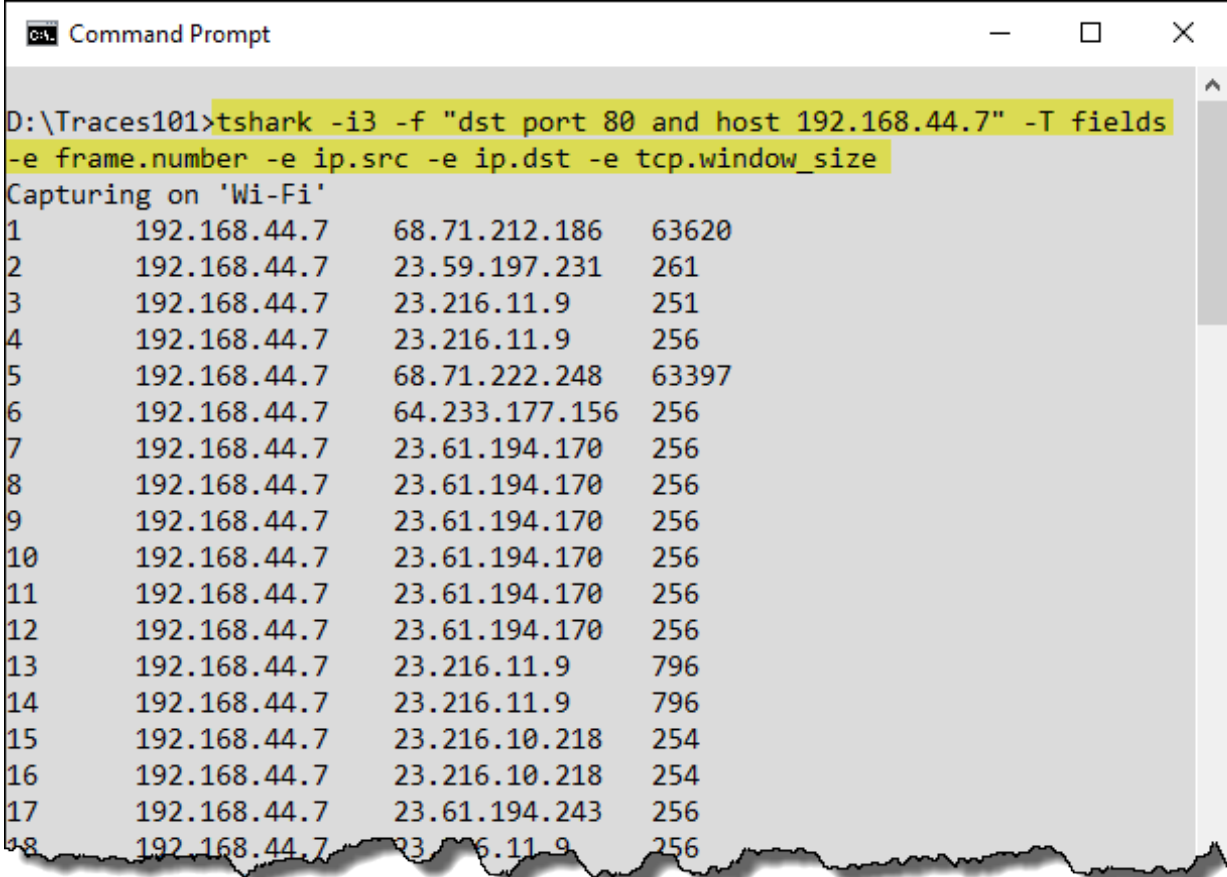
Directory of D:\Traces101

01/11/2016  04:44 PM                1,072 tcpanalysisflags.pcapng
               1 File(s)                1,072 bytes
               0 Dir(s)  1,756,563,681,280 bytes free

D:\Traces101>
```

Use Tshark to Export Specific Field Values and Statistics from a Trace File

-T fields -e <field name>



The screenshot shows a Windows Command Prompt window with the following command and output:

```
D:\Traces101>tshark -i3 -f "dst port 80 and host 192.168.44.7" -T fields
-e frame.number -e ip.src -e ip.dst -e tcp.window_size
Capturing on 'Wi-Fi'
```

frame.number	ip.src	ip.dst	tcp.window_size
1	192.168.44.7	68.71.212.186	63620
2	192.168.44.7	23.59.197.231	261
3	192.168.44.7	23.216.11.9	251
4	192.168.44.7	23.216.11.9	256
5	192.168.44.7	68.71.222.248	63397
6	192.168.44.7	64.233.177.156	256
7	192.168.44.7	23.61.194.170	256
8	192.168.44.7	23.61.194.170	256
9	192.168.44.7	23.61.194.170	256
10	192.168.44.7	23.61.194.170	256
11	192.168.44.7	23.61.194.170	256
12	192.168.44.7	23.61.194.170	256
13	192.168.44.7	23.216.11.9	796
14	192.168.44.7	23.216.11.9	796
15	192.168.44.7	23.216.10.218	254
16	192.168.44.7	23.216.10.218	254
17	192.168.44.7	23.61.194.243	256
18	192.168.44.7	23.61.11.9	256

Continue Learning about Wireshark and Network Analysis

- Visit **www.wiresharkbook.com** (other Wireshark books and links to related tools).
- Visit **www.wireshark.org** to sign up for the Wireshark-Announce mailing list (new Wireshark version information).
- Sign up for the newsletter at **www.chappellU.com** to participate in free online Wireshark events.
- Practice capturing your own traffic.
- Continue customizing Wireshark by adding new profiles and new display filters, coloring rules, and Filter Expression buttons.
- Share your customized settings with other IT team members to create a master profile that improves your team's network analysis efficiency.

Course Conclusion

Filtering Slides

IPv4/IPv6 Capture Filters

<code>host 10.3.1.1</code>	Capture traffic to/from 10.3.1.1
<code>host 2406:da00:ff00::6b16:f02d</code>	Capture traffic to/from the IPv6 address 2406:da00:ff00::6b16:f02d
<code>not host 10.3.1.1</code>	Capture all traffic except traffic to/from 10.3.1.1
<code>src host 10.3.1.1</code>	Capture traffic from 10.3.1.1
<code>dst host 10.3.1.1</code>	Capture traffic to 10.3.1.1
<code>host 10.3.1.1 or host 10.3.1.2</code>	Capture traffic to/from 10.3.1.1 and any host it is communicating with and traffic to/from 10.3.1.2 and any host it is communicating with
<code>host www.espn.com</code>	Capture traffic to/from any IP address that resolves to www.espn.com (this will only work if the host name can be resolved by Wireshark prior to capture)

Subnet Capture Filters

<code>net 10.3.0.0/16</code>	Capture traffic to/from any host on network 10.3.0.0
<code>net 10.3.0.0 mask 255.255.0.0</code>	Same result as previous filter
<code>ip6 net 2406:da00:ff00::/64</code>	Capture traffic to/from any host on network 2406:da00:ff00:0000 (IPv6)
<code>not dst net 10.3.0.0/16</code>	Capture all traffic except traffic to an IP address starting with 10.3
<code>dst net 10.3.0.0/16</code>	Capture traffic to any IP address starting with 10.3
<code>src net 10.3.0.0/16</code>	Capture traffic from any IP address starting with 10.3
<code>net 10.3.0.0/16</code>	Capture traffic to/from any host on network 10.3.0.0

Broadcast and Multicast Capture Filters

<code>ip broadcast</code>	Capture traffic to 255.255.255.255
<code>ip multicast</code>	Capture traffic to 224.0.0.0 through 239.255.255.255 (also catches traffic to 255.255.255.255 unless you add and not ip broadcast)
<code>dst host ff02::1</code>	Capture traffic to the IPv6 multicast address for all hosts
<code>dst host ff02::2</code>	Capture traffic to the IPv6 multicast address for all routers

MAC Address Capture Filters

<code>ether host 00:08:15:00:08:15</code>	Capture traffic to or from 00:08:15:00:08:15
<code>ether src 02:0A:42:23:41:AC</code>	Capture traffic from 02:0A:42:23:41:AC
<code>ether dst 02:0A:42:23:41:AC</code>	Capture traffic to 02:0A:42:23:41:AC
<code>not ether host 00:08:15:00:08:15</code>	Capture traffic to or from any MAC address except for traffic to or from 00:08:15:00:08:15

Capture Traffic for a Specific Application

<code>port 53</code>	Capture UDP/TCP traffic to or from port 53 (typically DNS traffic)
<code>not port 53</code>	Capture all UDP/TCP traffic except traffic to or from port 53
<code>port 80</code>	Capture UDP/TCP traffic to or from port 80 (typically HTTP traffic)
<code>udp port 67</code>	Capture UDP traffic to or from port 67 (typically DHCP traffic)
<code>tcp dst port 21</code>	Capture TCP traffic to port 21 (typically the FTP command channel)
<code>portrange 1-80</code>	Capture UDP/TCP traffic to or from ports from 1 through 80
<code>tcp portrange 1-80</code>	Capture TCP traffic to or from ports from 1 through 80

Combine Port-Based Capture Filters

<code>port 20 or port 21</code>	Capture all UDP/TCP traffic to or from port 20 or port 21 (typically FTP data and command ports)
<code>host 10.3.1.1 and port 80</code>	Capture UDP/TCP traffic to or from port 80 that is being sent to or from 10.3.1.1
<code>host 10.3.1.1 and not port 80</code>	Capture UDP/TCP traffic to or from 10.3.1.1 except traffic to or from port 80
<code>udp src port 68 and udp dst port 67</code>	Capture all UDP traffic from port 68 to port 67 (typically traffic sent from a DHCP client to a DHCP server)
<code>udp src port 67 and udp dst port 68</code>	Capture all UDP traffic from port 67 to port 68 (typically traffic sent from a DHCP server to a DHCP client)
<code>port 20 or port 21</code>	Capture all UDP/TCP traffic to or from port 20 or port 21 (typically FTP data and command ports)
<code>host 10.3.1.1 and port 80</code>	Capture UDP/TCP traffic to or from port 80 that is being sent to or from 10.3.1.1

Capture Specific ICMP Traffic

<code>icmp</code>	Capture all ICMP packets.
<code>icmp[0]=8</code>	Capture all ICMP Type 8 (Echo Request) packets.
<code>icmp[0]=17</code>	Capture all ICMP Type 17 (Address Mask Request) packets.
<code>icmp[0]=8 or icmp[0]=0</code>	Capture all ICMP Type 8 (Echo Request) packets or ICMP Type 0 (Echo Reply) packets.
<code>icmp[0]=3 and not icmp[1]=4</code>	Capture all ICMP Type 3 (Destination Unreachable) packets except for ICMP Type 3/Code 4 (Fragmentation Needed and Don't Fragment was Set) packets.
<code>icmp</code>	Capture all ICMP packets.
<code>icmp[0]=8</code>	Capture all ICMP Type 8 (Echo Request) packets.

Apply Display Filters based on an IP Address, Range of Addresses, or Subnet

Address Filter Type	Filter Example
Single IPv4 Address	<code>ip.addr==10.3.1.1</code>
Single IPv6 Address	<code>ipv6.addr==2406:da00:ff00::6b16:f02d</code>
Host Name*	<code>ip.host==www.wireshark.org</code>
Range of Addresses	<code>ip.addr > 10.3.0.1 && ip.addr < 10.3.0.5</code>
Subnet (IPv4)	<code>ip.addr==10.3.0.0/16</code>
Subnet (IPv6)	<code>ipv6.addr >= fe80:: && ipv6.addr < fec0::</code>

* You must enable Wireshark's **Resolve network (IP) addresses** setting (**Edit | Preferences | Name Resolution**) in order to use this display filter.

Expand Display Filters with Multiple Include and Exclude Conditions (using Operators)

Operator	English	Example	Description
<code>&&</code>	and	<code>ip.src==10.2.2.2 && tcp.port==80</code>	View all IPv4 traffic from 10.2.2.2 that is to or from port 80
<code> </code>	or	<code>tcp.port==80 tcp.port==443</code>	View all TCP traffic to or from ports 80 or 443
<code>!</code>	not	<code>!arp</code>	View all traffic except ARP traffic
<code>!=</code>	ne	<code>tcp.flags.syn != 1</code>	View TCP frames that do not have the TCP SYN flag (synchronize sequence numbers) set to 1

Why Didn't my Filter Work?

Incorrect `ip.addr != 10.2.2.2`

Display packets that do not have 10.2.2.2 in the IP source address field *or* IP destination address field.

Correct `!ip.addr == 10.2.2.2`

Display packets that do not have 10.2.2.2 in the IP source address field and also does not have 10.2.2.2 in the destination address field.

Incorrect `!tcp.flags.syn==1`

Display all packets that do not have a TCP SYN bit set to 1 (regardless of whether they are a TCP packet or not)

Correct `tcp.flags.syn != 1`

This filter will only display TCP packets that contain a SYN set to 0.

Use Parentheses to Change Filter Meaning

```
(tcp.port==80 && ip.src==10.2.2.2) || tcp.flags.syn==1
```

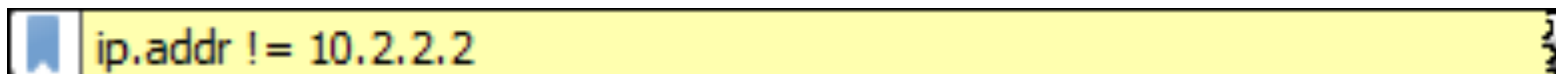
Diagram illustrating the filter expression: `(tcp.port==80 && ip.src==10.2.2.2) || tcp.flags.syn==1`. The expression is enclosed in parentheses, indicating that the first part `(tcp.port==80 && ip.src==10.2.2.2)` is evaluated first, followed by the `||` operator, and then the second part `tcp.flags.syn==1`. The diagram uses a red bracket for the first part, a green bracket for the second part, and a grey line with "OR" to indicate the logical OR operation.

```
tcp.port==80 && (ip.src==10.2.2.2 || tcp.flags.syn==1)
```

Diagram illustrating the filter expression: `tcp.port==80 && (ip.src==10.2.2.2 || tcp.flags.syn==1)`. The expression is enclosed in parentheses, indicating that the second part `(ip.src==10.2.2.2 || tcp.flags.syn==1)` is evaluated first, followed by the `&&` operator, and then the first part `tcp.port==80`. The diagram uses a red bracket for the first part, a green bracket for the second part, and a grey line with "OR" to indicate the logical OR operation.

Determine Why Your Display Filter Area is Yellow

Yellow Background: This filter may not work as expected.

A horizontal bar representing a Wireshark display filter. It has a yellow background and a blue bookmark icon on the left. The text inside is `ip.addr != 10.2.2.2`.

`ip.addr != 10.2.2.2`

Green Background: The syntax is correct, but it doesn't ensure the logic is correct.

A horizontal bar representing a Wireshark display filter. It has a green background and a blue bookmark icon on the left. The text inside is `arp && bootp && tcp`.

`arp && bootp && tcp`

Red Background: This filter will not work – there is a syntax error.

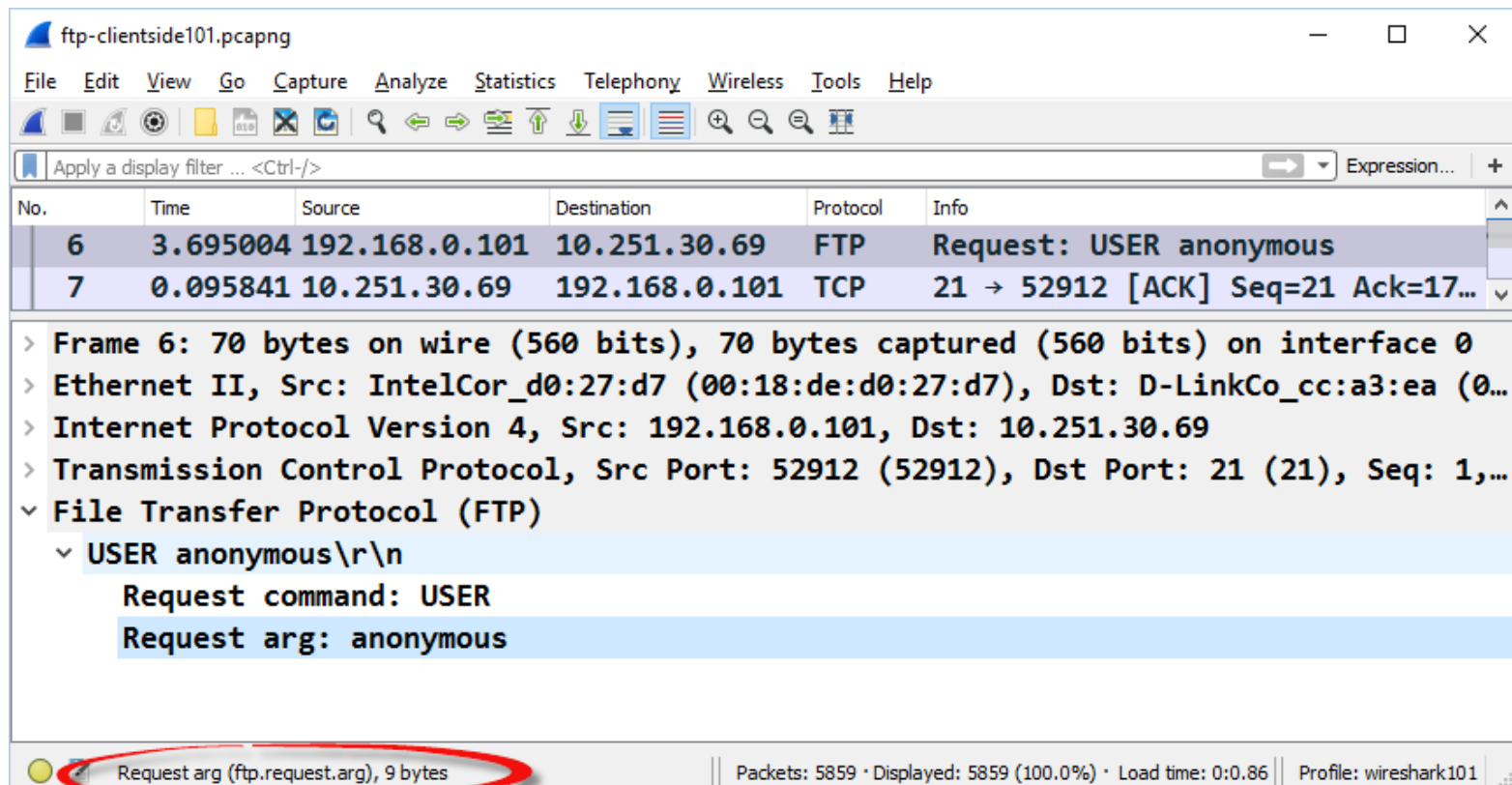
A horizontal bar representing a Wireshark display filter. It has a red background and a blue bookmark icon on the left. The text inside is `!ip.addr=10.2.2.2`.

`!ip.addr=10.2.2.2`

Filter on a Keyword in a Trace File

Use contains for a general filter

Use matches for a Regex filter



ftp-clientside101.pcapng

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

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

No.	Time	Source	Destination	Protocol	Info
6	3.695004	192.168.0.101	10.251.30.69	FTP	Request: USER anonymous
7	0.095841	10.251.30.69	192.168.0.101	TCP	21 → 52912 [ACK] Seq=21 Ack=17...

> Frame 6: 70 bytes on wire (560 bits), 70 bytes captured (560 bits) on interface 0

> Ethernet II, Src: IntelCor_d0:27:d7 (00:18:de:d0:27:d7), Dst: D-LinkCo_cc:a3:ea (0...

> Internet Protocol Version 4, Src: 192.168.0.101, Dst: 10.251.30.69

> Transmission Control Protocol, Src Port: 52912 (52912), Dst Port: 21 (21), Seq: 1,...

▼ File Transfer Protocol (FTP)

▼ USER anonymous\r\n

Request command: USER

Request arg: anonymous

Request arg (ftp.request.arg), 9 bytes

Packets: 5859 · Displayed: 5859 (100.0%) · Load time: 0:0.86 | Profile: wireshark101

ftp.request.arg contains "anonymous"

Using matches for Regex Filters

Consider case sensitivity

`ftp.request.arg` matches `"anonymous"`

`ftp.request.arg` matches `"(?i)anonymous"`

Consider variable characters

`frame` matches `"building[Aa]eng"`

`frame` matches `"building[AaBb]eng"`

`frame` matches `"(?i)(cat|dog)"`

Use Wildcards in Display Filters

`ftp.request.arg` matches `"me.r"`

“.” indicates any character except a carriage return or line feed

`ftp.request.arg` matches `"me..r"`

Now we're looking for any two characters between `me` and `r`

`ftp.request.arg` matches `"me.{1,3}r"`

`{#,#}` indicates minimum and maximum number of repeating characters

Challenge Slides

Section 0 Challenge

Open **challenge101-0.pcapng** and use the techniques covered in this Section to answer these Challenge questions. The answer key is located in Appendix A.

We will focus on what you can learn about communications based on the main Wireshark view.

- Question 0-1. How many packets are in this trace file?
- Question 0-2. What IP hosts are making a TCP connection in frames 1, 2, and 3?
- Question 0-3. What HTTP command is sent in frame 4?
- Question 0-4. What is the length of the largest frame in this trace file?
- Question 0-5. What protocols are seen in the **Protocol** column?
- Question 0-6. What responses are sent by the HTTP server?
- Question 0-7. Is there any IPv6 traffic in this trace file?

Section 1 Challenge

Open **challenge101-1.pcapng** and use the techniques covered in this Section to answer these Challenge questions. The answer key is located in Appendix A.

Important: This trace file includes an HTTP communication running over a non-standard port number. Before you can answer these questions, you must force Wireshark to dissect this traffic as HTTP.

Question 1-1. In which frame number does the client request the default web page ("/")?

Question 1-2. What response code does the server send in frame 17?

Question 1-3. What is the largest TCP delta value seen in this trace file?

Question 1-4. How many SYN packets arrived after at least a 1 second delay?

Section 2 Challenge

This challenge requires access to the Internet. You will capture traffic to a web site and analyze your findings. The answer key is located in Appendix A.

First, configure Wireshark to capture only traffic to and from your MAC address and port 80, and save the traffic to a file named **mybrowse.pcapng**. Then ping and browse to www.chappellU.com. Stop the capture and examine the trace file contents.

Question 2-1. Did you capture any ICMP traffic?

Question 2-2. What protocols are listed for your browsing session to www.chappellU.com?

Now configure Wireshark to capture all your ICMP traffic, and save your traffic to a file called **myicmp.pcapng**. Again, ping and browse to www.chappellU.com. Stop the capture and examine the trace file contents.

Question 2-3. How many ICMP packets did you capture?

Question 2-4. What ICMP Type and Code numbers are listed in your trace file?

Section 3 Challenge

Open **challenge101-3.pcapng** and use your display filter and coloring rule skills to locate traffic based on addresses, protocols and keywords to answer these Challenge questions.

You will practice your display filter to locate traffic based on addresses, protocols, and keywords.

- | | |
|---------------|--|
| Question 3-1. | How many frames travel to or from 80.78.246.209? |
| Question 3-2. | How many DNS packets are in this trace file? |
| Question 3-3. | How many frames have the TCP SYN bit set to 1? |
| Question 3-4. | How many frames contain the string “set-cookie” in upper case or lower case? |
| Question 3-5. | How many frames contain a TCP delta time greater than 1 second? |

Section 4 Challenge

Open **challenge101-4.pcapng** and use your packet coloring and export skills in this Section to answer these Challenge questions.

Question 4-1. What coloring rule does frame 170 match?

Question 4-2. Temporarily color TCP stream 5 with a light blue background and apply a filter on this traffic. How many packets match your filter?

Question 4-3. Create and apply a coloring rule for TCP delta delays greater than 100 seconds. How many frames match this coloring rule?

Question 4-4. Export this filtered TCP delta information in CSV format. Using a spreadsheet program, what is the average TCP delta time?

Section 5 Challenge

Open **challenge101-5.pcapng** and use the techniques covered in this Section to answer these Challenge questions.

- Question 5-1. Create an IO Graph for this trace file. What is the highest packets-per-second value seen in this trace file?
- Question 5-2. What is the highest bits-per-second value seen in this trace file?
- Question 5-3. How many TCP conversations are in this trace file?
- Question 5-4. How many times has “Previous segment not captured” been detected in this trace file?
- Question 5-5. How many retransmissions and fast retransmissions are seen in this trace file?

Section 6 Challenge

Open *challenge101-6.pcapng* and use the techniques covered in this Section to answer these Challenge questions. The answer key is located in Appendix A.

- Question 6-1. What two .jpg files can be exported from this trace file?
- Question 6-2. On what HTTP server and in what directory does next-active.png reside?
- Question 6-3. Export *booksmall.png* from this trace file. What is in the image?
- Question 6-4. Reassemble TCP stream 7. What type of browser is the client using in this stream?

Section 7 Challenge

Open ***challenge101-7.pcapng*** and use the techniques covered in this Section to answer these Challenge questions. The answer key is located in Appendix A.

- | | |
|---------------------------------|---|
| Question 7-1. | What information is contained in the trace file annotation? |
| Question 7-2. | What packet comments are contained in this trace file? |
| Question 7-3.
did you alter? | Add a comment to the POST message in this trace file. What packet |

Section 8 Challenge

Use *challenge101-8.pcapng* and the command-line tool techniques covered in this Section to answer these Challenge questions. The answer key is located in Appendix A.

Question 8-1. What Tshark parameter should you use to list active interfaces on your Wireshark system?

Question 8-2. Using Tshark to extract protocol hierarchy information, how many UDP frames are in *challenge101-8.pcapng*?

Question 8-3. Use Tshark to export all DNS packets from *challenge101-8.pcapng* to a new trace file called *ch8dns.pcapng*. How many packets were exported?

Lab Slides

Starting at Lab 4

Lab 4

[http-disney101.pcapng](#)

Add the HTTP Host Field as a Column

During a browsing session, an HTTP client sends requests for HTTP objects to one or more HTTP servers. In each of the requests, the client specifies the name or the IP address of the target HTTP server. This can be very revealing.

Note: All frames from 24.6.173.220 will appear with a black background and red foreground if Wireshark is set to validate IP header checksums. You will disable this feature in Lab 6.

Lab 5

[http-pcaprnet101.pcapng](http://pcaprnet101.pcapng)

Set Key Wireshark Preferences (IMPORTANT LAB)

Wireshark offers several key preference settings to enhance your analysis sessions. In this lab you will use the Edit Preferences button on the main toolbar and the right-click method to view and change the preference settings.

These are the settings we will view and alter in this lab:

- Increase the number of display filters that Wireshark remembers.
- Increase the number of recently opened files that Wireshark remembers.
- Ensure IP, UDP, and TCP checksum validations are disabled.
- Enable the TCP *Calculate conversation timestamps* setting.
- Enable the TCP *Track number of bytes in flight* setting.
- Disable the TCP *Allow subdissector to reassemble TCP streams* setting.

Lab 6

Create a New Profile Based on the Default Profile

Profiles enable you to work with customized settings to be more efficient when analyzing traffic.

In this lab you will create a new profile called “**wireshark101.**”

You will base it on your Default profile to ensure any previously created settings will be copied over to your new profile.

Lab 7

[httpdnsprofile2.zip](#)
and
[dns-nmap101.pcapng](#)

Import a DNS/HTTP Errors Profile

Once you've created that fabulous profile that detects various types of HTTP or DNS problems perhaps, consider installing that profile on your other Wireshark systems.

Since Wireshark bases profiles on text files, this is a simple process.

Lab 8

<http://slow101.pcapng>

Spot Path and Server Latency Problems

Let's practice using these two columns to detect latency.

In this lab you will set the Time column to **Seconds Since Previous Displayed Packet** and add the **TCP Delta** column.

You may have some of these columns set already if you followed along with the previous section in your Student Manual.

Lab 9

Capture to File Sets

In this lab you will get a chance to practice capturing to file sets using an auto-stop condition.

Lab 10

Use a Ring Buffer to Conserve Drive Space

In this lab exercise, we will set up a ring buffer to ensure we see the most recent traffic.

We will create a problem and manually stop the capture to analyze the issue.

Lab 11

Capture Only Traffic to or from Your IP Address

In this lab you will determine your current IP address and apply a capture filter for that traffic.

Lab 12

Capture Only Traffic to or from Everyone Else's MAC Address

In this lab you will determine your current MAC address and apply a capture filter that filters out your traffic—you are interested in everyone else's traffic only.

If you have a dual-stack host, it is much more effective to make a single filter based on your MAC address than to make a more complex filter based on your IPv4 and IPv6 addresses.

Lab 13

Create, Save and Apply a DNS Capture Filter

In this exercise you will use several skills learned in this Section. You will configure Wireshark to capture only DNS traffic and save that traffic to a file called **mydns101.pcapng**.

Lab 14

[http-sfgate101.pcapng](#)

Use Auto-Complete to Find Traffic to a Specific HTTP Server

In this lab we use Wireshark's auto-complete feature to filter on specific HTTP communications.

Ultimately, we are interested in client requests to a particular server.

This trace file, *http-sfgate101.pcapng*, was captured as someone browsed a web site and then filled in a feedback form on that site asking about iPad support.

Lab 15

Use a Default Filter as a “Seed” for a New Filter

You can use the default display filters as a template to create and save new custom display filters.

This method helps you remember the display filter syntax and ensures that the syntax is correct. We will create a display filter for all traffic to or from your IP address.

Lab 16

[http-disney101.pcapng](#)

Filter on HTTP Traffic the Right Way

This is a quick lab.

We will just compare the results from applying two different display filters to the traffic.

We will use `http` and then we will replace it with the proper filter for this web browsing traffic.

Lab 17

mybackground101.pcapng

Filter on Traffic to or from Online Backup Subnets

In this lab, we will apply a subnet display filter to examine traffic to or from a backup server for Memeo which offers an online backup product. This traffic runs in the background, constantly checking in with the server.

Lab 18

[http-errors101.pcapng](#)

Filter on DNS Name Errors or HTTP 404 Responses

In this lab we will look for specific DNS or HTTP error responses using the right-click method.

This is a great filter that you may want to save.

Lab 19

[gen-startupchatty101.pcapng](#)

Detect Background File Transfers on Startup

There may be a number of background processes that run when you start up your machine.

Some of these may update your virus detection mechanism, your operating system, or applications.

In this lab, you will detect and filter on the most active conversation of a host that is just starting up.

Lab 20

general101b.pcapng

Locate TCP Connection Attempts to a Client

Client processes send TCP connection requests to server processes. There are very few reasons to allow incoming TCP connections to user machines on your network (as they typically won't be running server processes).

In this lab we will create a display filter that detects incoming TCP connection attempts to anyone on a particular subnet.

We will focus on subnet 24.6.0.0/16.

Lab 21

[http-pictures101.pcapng](http://pictures101.pcapng)

Filter to Locate a Set of Key Words in a Trace File

In this lab we will use the matches operator to find the keywords *sombrero* or *football* in upper case or lower case anywhere in a trace file.

Lab 22

[http-pictures101.pcapng](http://pictures101.pcapng)

Filter with Wildcards between Words

In this lab we will use the matches operator to find the keywords *baby* and *smiling* in a trace file. We will see how the repeating character option settings can affect what matches your filter.

Our display filter `ftp.request.arg matches "me.{1,3}r"` would look for the "." up to three times between the "me" and "r" as mentioned in this section.

This time we will look for the keywords *baby* and *smiling* with up to three characters separating the words.

Lab 23

dfilters_sample.txt

Import Display Filters into a Profile

In this lab you will import a set of display filters from your student USB stick (or the *wiresharkbook.com* website) into your existing display filter file (*dfilters*).

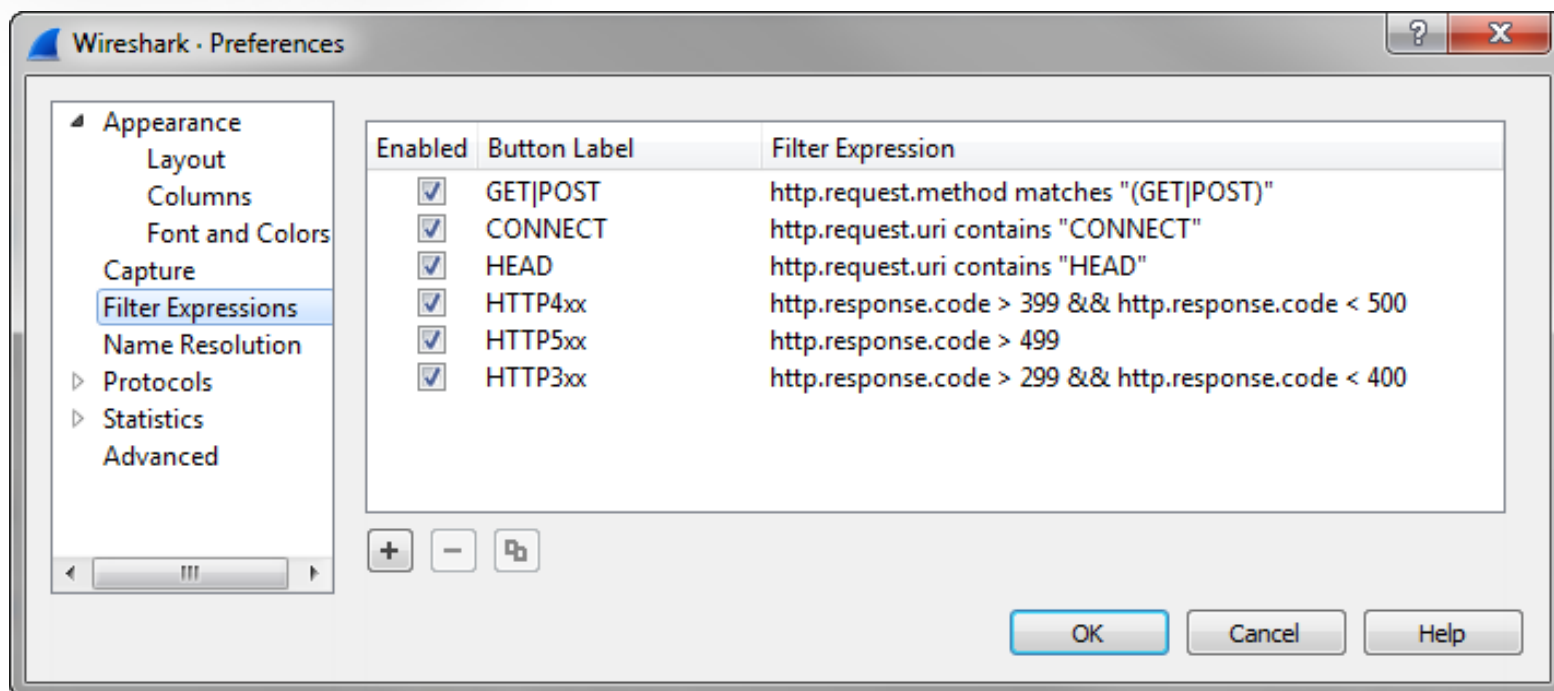
Use this same technique if you want to move display filters from one profile to another on a single host or other Wireshark systems.

Lab 24

http-chappellu101b.pcapng
and
filterexpressions101.txt

Create and Import HTTP Filter Expression Buttons

We will begin by creating a single Filter Expression button and then we'll import the set of Filter Expression buttons shown below into your *wireshark101* profile.



Lab 25

[http-sfgate101.pcapng](http://sfgate101.pcapng)

Add a Column to Display Coloring Rules in Use

Adding a column to identify coloring rules is a great idea when you are new to Wireshark or you just aren't familiar with the coloring rules set.

Lab 26

ftp-crack101.pcapng

Build a Coloring Rule to Highlight FTP User Names, Passwords, etc.

In this lab you will create a coloring rule to call your attention to FTP request arguments, including those associated with USER, PASS, TYPE, SIZE, MDTM, RETR, and CWD commands.

Lab 27

[http-browse101d.pcapng](http://browse101d.pcapng)

Create Temporary Conversation Coloring Rules

In this lab, you will apply three temporary coloring rules to differentiate TCP conversations. When you scroll through the trace file, you will be able to easily see when an earlier conversation begins to surface.

Lab 28

[net-lost-route.pcapng](#)

Create Temporary Conversation Coloring Rules

In this lab we will create a new coloring rule to identify TCP retransmissions. TCP retransmissions are a sign of packet loss on a network and are part of Wireshark's TCP analysis flagged packets. We'd like to just look at the Intelligent Scrollbar to know if retransmissions (packet loss indications) are seen.

Lab 29

[http-misctraffic101.pcapng](http://misctraffic101.pcapng)

Export a Single TCP Conversation

When you are focused on a specific application or a specific file download, it helps to extract conversations into separate trace files. In this lab, you will create and extract a new trace file after locating traffic from an executable file download process.

Lab 30

[http-au101b.pcapng](http://au101b.pcapng)

Export a List of HTTP Host Field Values from a Trace File

In this lab, you will alter the Packet List pane to display the HTTP Host field before exporting information to CSV format.

Lab 31

[http-misctrffic101.pcapng](#)

Filter on the Most Active TCP Conversation

Pulling out the most active conversation is a common network analysis task when trace files contain tens or even hundreds of conversations.

Lab 33

[http-browse101c.pcapng](http://browse101c.pcapng)

Set up GeoIP to Map Targets Globally

Wireshark can use the MaxMind GeoLite database files to list the country, city, AS (Autonomous System) number, latitude, and longitude of an IP address and map IPv4 and IPv6 addresses on a map of the earth. In this lab, you will configure Wireshark to use this database and map IP addresses seen in a trace file.

Lab 34

general101c.pcapng

Detect Suspicious Protocols or Applications

When you are concerned that there may be a security issue in your trace file, open the Protocol Hierarchy window first.

Look for suspicious applications or protocols and the dreaded “data” under IP, UDP, or TCP.

Lab 35

[http-espn101.pcapng](#)

Compare Traffic to/from a Subnet to Other Traffic

In this lab you will compare all the traffic to or from subnet 184.0.0.0/8 to all other traffic. To do this, you will use two IP address filters—one inclusion filter and one exclusion filter.

Lab 35

[http-download101.pcapng](#)

Identify an Overloaded Client

In this lab we use the Expert Infos window to identify the cause of poor network performance. Not only is the client overloaded in this trace file, but there is packet loss along the path as well.

Lab 36

[general101d.pcapng](#)

Detect and Graph File Transfer Problems

In this lab we examine a file transfer process that takes place over TCP. Before we can consider troubleshooting the application itself, we must rule out TCP problems.

Lab 37

[http-wiresharkdownload101.pcapng](http://wiresharkdownload101.pcapng)

Use Reassembly to Find a Web Site's Hidden HTTP Message

It is not unusual to have numerous “hidden” messages sent to your browser when you hit a web site. In this lab you will analyze a trace file that contains two hidden messages. Afterwards, visit the same web site again to catch other interesting messages.

Lab 38

ftp-clientside101.pcapng

Extract a File from an FTP File Transfer

In this lab you will follow an FTP data stream to reassemble the file that was transferred.

First you will reassemble the command channel traffic to see the client login and file retrieval commands, and then you will reassemble the data transfer channel traffic to view the file transferred.

Lab 39

[http-college101.pcapng](http://college101.pcapng)

Carve Out an HTTP Object from a Web Browsing Session

In this lab, you will open a trace file that contains a web browsing session.

Using the **File | Export Objects** process, you will extract one of the images transferred during the web browsing session.

Lab 40

sec-suspicious101.pcapng

Read Analysis Notes in a Malicious Redirection Trace File

It can be a blessing to have notes inside the trace file to assist other analysts (or even you) in following along with the traffic flow.

In this lab you will examine the notes left in a trace file that contains unusual communications.

Lab 41

sec-suspicious101.pcapng

Export Malicious Redirection Packet Comments

We will use the *sec-suspicious101.pcapng* trace file again in this lab.

We will use a two-step process for comment export.

First we will prepare the trace file to export the field information we are most interested in. We will export the fields in text format.

Unlike in the previous section, we will export the packet comments using the Packet summary line.

Lab 42

<http://download101c.pcapng>

Split a File and Work with Filtered File Sets

You will be working with *http-download101c.pcapng* in this lab. This trace file is only 27 MB, but we will use it to practice splitting a file.

After splitting the file, we will move through the file set while a display filter is applied.

Wireshark automatically applies the display filter to each file as it is opened.

Lab 43

[http-downloadc5000*.pcapng](#)

Merge a Set of Files using a Wildcard

In this lab you will merge the six-file *http-downloadc5000*.pcapng* set that you created in Lab 42.

You will use a wildcard to make this process a bit easier and less error-prone.

Lab 44

Use Tshark to Capture to File Sets with an Autostop Condition

In this lab, you will get a chance to use Tshark with various parameters. We'll define file set "next file" parameters and include an autostop condition for unattended capture.

Lab 45

[http-esp101.pcapng](#)

Use Tshark to Extract HTTP GET Requests

In this lab you will use the `-r` parameter to read a trace file and then apply a display filter with the `-R` parameter.

Finally you will save a trace file that contains only the HTTP GET requests.

Lab 46

Use Tshark to Extract HTTP Host Names and IP Addresses

In this lab we will use a combination of display filters and field names to create a file that contains both the IP addresses and host names of HTTP servers contacted on the network.