



Wireshark® Workbook 1

Practice, Challenges, and Solutions

Laura Chappell

Founder, Chappell University™

Creator of the WCNA Certification Program

(formerly referred to as the Wireshark Certified Network Analyst program)

Edited by James Aragon

*Always ensure you have proper authorization
before you listen to or capture network traffic.*

Protocol Analysis Institute, Inc.
59 Damonte Ranch Parkway, #B340
Reno, NV 89521 USA

Chappell University
info@chappellU.com
www.chappellU.com

Copyright Notice

Copyright 2020, Protocol Analysis Institute, Inc., dba Chappell University. All rights reserved. No part of this book, or related materials, may be reproduced or transmitted in any form, by any means (electronic, photocopying, recording, or otherwise) without the prior written permission of the publisher.

To arrange bulk purchase discounts for sales promotions, events, training courses, or other purposes, please contact Chappell University (info@chappellU.com).

ISBN10: 1-893939-63-4

ISBN13: 978-1-893939-64-6

(Version 1.0a)

Distributed worldwide for Chappell University through Protocol Analysis Institute, Inc. Protocol Analysis Institute, Inc. is the educational materials distributor for Chappell University.

For general information on Chappell University or Protocol Analysis Institute, Inc., including information on corporate licenses, updates, future titles, or courses, contact the Protocol Analysis Institute, Inc., at info@chappellu.com.

For authorization to photocopy items for corporate, personal, or educational use, contact Protocol Analysis Institute, Inc., at info@chappellu.com.

Trademarks. All brand names and product names used in this book and related documents are trade names, service marks, trademarks, or registered trademarks of their respective owners. Wireshark and the “fin” logo are registered trademarks of the Wireshark Foundation. At the time this book was written, the Wireshark Foundation, Inc., only had Riverbed senior management as Officers and members of the Board of Directors. I guess Wireshark is owned by Riverbed. Sigh. Shame on you, Riverbed.

Limit of Liability/Disclaimer of Warranty. The author and publisher have used their best efforts in preparing this book and the related materials. Protocol Analysis Institute, Inc., Chappell University, and the author(s) make no representations or warranties of merchantability of fitness for a particular purpose. Protocol Analysis Institute, Inc., and Chappell University assume no liability for any damages caused by following the instructions or using the techniques or tools listed in this book and related materials. Protocol Analysis Institute, Inc., Chappell University, and the author(s) make no representations or warranties that extend beyond the descriptions contained in this paragraph. No warranty may be created or extended by sales representatives or written sales materials. The accuracy or completeness of the information provided herein and the opinions stated herein are not guaranteed or warranted to produce any particular result and the advice and strategies contained herein may not be suitable for every individual. Protocol Analysis Institute, Inc., Chappell University, and the author(s) shall not be liable for any loss of profit or any other damages, including without limitation, special, incidental, consequential, or other damages.

Copy Protection. In all cases, reselling or duplication of this book and related materials without explicit written authorization is expressly forbidden. We will find you, ya know. So don't steal or plagiarize this book.

Index

- .csv format, exporting to, **241, 244, 310, 329**
- .gzip file decompression, **247**
- .pcap format warning, **76**
- .pcapng format, **74**
- 1 window size scaling factor, **65, 341**
- advanced preference settings, **129**
- Akamai, detect use of, **168, 291**
- Amazon Web Services, **191**
- analysis of Yahoo search, **169**
- Apple devices, locating, **158**
- Apple Push Notification (APN) traffic, **213**
- application receiving errors, **295**
- browse the code, **111**
- bug #8997, VSS-Monitoring Ethernet trailer, **116**
- Bugzilla, **112**
- buttons
 - dfilter_buttons* file, **289**
 - DNSErr, **193**
 - HTTPErrors, **46**
 - TCPAnalysis, **143**
- Capture File Properties* window
 - average bits/s, **27**
 - building a report based upon, **78**
 - capture tool detection, **27**
 - statistics, **27**
- capture location
 - determine the, **239**
 - response time accuracy, **11**
 - use TCP handshake to determine, **11**
- Classless Interdomain Routing (CIDR) in display filtering, **28**
- Cloudflare Content Delivery Network, **247**
- coloring information in *Frame* section, **77**
- columns
 - [TCP Flags]* summary line, **92**
 - adding *iRTT* column, **29**
 - Apply as Column*, **10, 15, 44**
 - display multiple fields in one column, **75**
 - for TCP stream index information, **32**
 - for UDP stream index information, **30**
 - hiding, **19**
 - HTTP *Host* field, **302**
 - HTTP *Server* field, **247**
 - http.server*, **324**
 - http.time*, **243**
 - Info (TCP handshake analysis), **21**
 - initial Round Trip Time (iRTT)*, **133**
 - length* field, **116**
 - renaming, **44**
 - sort order indicator, **24**
 - sorting, **10, 24**
- comments
 - adding packet, **78**
 - adding trace file, **27, 78**
 - building a report based upon, **78**
- content delivery service, identifying the, **247**
- copying field values*, **231**
- delays. See time
- display filters
 - !(tcp.option_kind==2)*, **83**
 - !=* use, **165**
 - !tcp.analysis.keep_alive && tcp.len > 0*, **245**
 - !tcp.port==80*, **58**
 - !tcp.port==x*, **212**
 - (?i)* in earlier versions for regex, **284**
 - (?-i)* to remove case insensitivity, **284**

346 Index

- applied to the conversation window, **260**
- bootp, **160**
- bootp (older), **160**
- bootp.option.hostname (older), **160**
- case insensitive, **284**
- casting a wide net, **101**
- check the Status Bar count, **129**
- detect missing TCP options, **83**
- dhcp, **160**
- dhcp.option.hostname, **160**
- dns contains "wiresharkbook", **101**
- dns.a==x, **174**
- dns.flags.rcode != 0, **165**
- dns.flags.rcode > 0, **165, 193, 287**
- dns.flags.recdesired==0, **197**
- dns.flags.response==0, **220, 283**
- dns.flags.response==1, **99, 101, 281, 283**
- dns.id==x, **224**
- dns.qry.name contains "x", **194**
- dns.qry.type==1, **221, 293**
- dns.qry.type==28, **225, 294**
- dns.resp.ttl < 5, **292**
- dns.resp.type==5, **198**
- don't use != with combo fields, **58**
- eth.addr==x, **159**
- field existence, **57, 258**
- follow streams filters, **47**
- for duplicate SYN/ACKs, **137**
- for IPv6 addresses, **204**
- frame contains "x", **179, 207**
- frame.number in {x x}, **325**
- ftp, **205**
- ftp.request.command in {x x}, **210**
- ftp.request.command=="x", **208**
- ftp.request.command=="x" filter, **207**
- http, **99**
- http contains "GET", **172, 176**
- http contains "X-", **45**
- http.content_length > 500000, **45**
- http.host, **14**
- http.host contains "x", **306**
- http.host matches "x", **306**
- http.referer, **308**
- http.request, **14, 99, 170, 179**
- http.request.method, **308**
- http.request.method=="GET", **167**
- http.request.method=="x", **306**
- http.request.uri, **17, 41**
- http.request.uri matches "x", **304**
- http.request_in, **57**
- http.response, **13, 158**
- http.response.code > 399, **19, 46, 63**
- http.response.code > 399 &&
 - http.response.code < 500, **63**
- http.response.code > 499, **63**
- http.response.in || http.request_in, **57**
- http.response_in, **57**
- http.server contains "Apache", **45**
- http.server matches "Apache", **45**
- http.time, **23**
- ip.dst in {x x}, **317**
- ip.flags.df==0, **33**
- ip.flags.mf==0 || ip.frag_offset > 0, **33**
- ip.src filtering, **28**
- ip.src==192.0.0.0/8, **28**
- ipv6, **86**
- lazy filters, **28, 190, 288, 323**
- matches operator, **190, 284**
- membership operator "in", **55**
- mixing && and ||, **86, 160**
- parentheses, automatically applied, **223**
- Prepare a filter, **33, 159, 220, 221, 223, 292, 294**
- problem with dns contains
 - "webaddress", **102**
- ssl.handshake (old), **59**
- subnet filtering, **28**
- SYN/ACK packets, **330**
- tcp.analysis listing, **128**
- tcp.analysis.duplicate_ack, **64, 258, 260, 269**
- tcp.analysis.duplicate_ack_frame, **258**
- tcp.analysis.flags, **143**
- tcp.analysis.initial_rtt, **29, 231, 334**

- tcp.analysis.keep_alive*, 248
- tcp.analysis.out_of_order*, 128, 142
- tcp.dstport in {80 443}*, 55
- tcp.flags.fin==1 || tcp.flags.reset==1*, 92
- tcp.flags.syn==1*, 71
- tcp.flags.syn==1 && tcp.flags.ack==0*, 16, 21, 39, 86, 126, 211, 301, 330
- tcp.flags.syn==1 && tcp.flags.ack==1*, 22, 127, 241, 330
- tcp.flags.syn==1 && tcp.len > 0*, 88
- tcp.flags==0x002*, 16, 39
- tcp.nxtseq==x*, 145, 147, 148
- tcp.option_kind==3*, 79
- tcp.option_kind==34*, 88
- tcp.option_kind==4*, 82, 339
- tcp.option_kind==8*, 83
- tcp.options.sack_le*, 339
- tcp.options.wscale.shift*, 89
- tcp.port==x*, 58, 59, 162
- tcp.seq==x*, 113, 115, 140, 274
- tcp.stream==x*, 32, 75, 109, 240
- tcp.time_delta*, 134, 204
- tcp.window_size_scalefactor*, 341
- tls vs. ssl*, 59
- tls.handshake*, 60
- tls.handshake.type==2*, 264
- to detect FTP modes, 210
- udp.stream==x*, 31, 75
- using \$ in regex, 306
- using == or *contains*, 42
- dissectors applied to frames, 77
- DNS
 - "No such name" response, 287
 - "Refused" response, 287
 - A record filter, 101
 - A record filters, 220, 288
 - AAAA record filter, 101
 - AAAA record query filter, 225
 - Answer RRs section, 286
 - Authority RRs section, 291
 - caching, 195, 226, 290
 - canonical name (CNAME), 191, 197, 198, 291
 - dns display filter*, 9
 - dns matches "x" filter*, 191
 - dns.a filter*, 174
 - dns.flags.rcode > 0 filter*, 165
 - dns.flags.response* field, 99, 284
 - dns.flags.response==1 filter*, 163, 281
 - dns.qry.name filter*, 103, 288
 - dns.qry.type==1 filter*, 221
 - dns.resp.ttl* field, 290
 - dns.resp.type==5*, 198
 - error responses, 287
 - filter for web address, 101
 - identify the client, 9
 - identify the server, 9, 281
 - iterative query, 196
 - measure response time, 10, 187, 224, 284, 286
 - multiple addresses offered, 194
 - obtain the IP address of a host, 302
 - percentage of packets, 187
 - problem with *dns contains "webaddress"* filter, 102, 103
 - recursive query, 196
 - reply codes, 193
 - retransmissions, 224, 285
 - Start of a Zone of Authority (SOA), 225, 296
 - Time* field in responses, 10
 - Time to Live* field (caching), 292
 - Type* field values, 294
- dumpcap, 27
- encapsulation type, 74
- Enhanced Packet Block (EPB) container, 74
- Excel, averaging with, 241
- Expert Information
 - ACKed lost segment, 11
 - Allow subdissector to reassemble TCP streams effect upon*, 248
 - Fast Retransmissions, 132
 - incorrect out-of-order designation, 133

348 Index

- launching, **128**
- out-of-order* packets, **142**
- Previous segment(s) not captured*, **248, 266, 271**
- retransmissions, **132**
- TCP error listing, **248**
- TCP Zero Window segment, **205**
- This frame is a (suspected) out-of-order segment*, **248**
- exporting
 - displayed packets, **262**
 - except ignored frames, **77**
 - HTTP objects, **307**
 - specified packets, **241, 244, 261, 310, 329**
- favicon.ico* file, **20, 32, 326**
- frame comments, **157**
- Frame* section, **49, 73**
 - arrival time, **74**
 - capture length, **76**
 - coloring rule name, **77**
 - coloring rule string, **77**
 - encapsulation type, **74**
 - epoch time, **75**
 - Frame is ignored*, **77**
 - Frame is marked*, **76**
 - frame length, **76**
 - frame number, **76**
 - interface ID, **74**
 - protocols in frame, **77**
 - summary lines, **74**
 - time delta from previous captured frame, **75**
 - time delta from previous displayed frame, **75**
 - time shift, **75**
- FTP
 - EPRT Extended Active mode, **210**
 - EPSV Extended Passive Mode, **210**
 - frame contains "x"* filter, **207**
 - ftp* filter, **205**
 - ftp.request.command=="x"*, **208**
 - PASV Passive Mode, **210**
 - PORT Active Mode, **210**
- Gabon, .ga ccTLD, **172**
- GitHub repository, **74**
- graphs
 - detecting Selective ACKs, **150**
 - IO Graph, **33**
 - Time-Sequence (tcptrace) graph, **150**
- gzipped traffic, reassembling, **246**
- hackerwatch.org, **203**
- hex editor, sanitizing with a, **304**
- HTTP
 - 200 OK response, **24**
 - 302 Moved Temporarily response, **175**
 - 404 Not Found response, **19, 46, 49, 315, 325**
 - 413 Request Entity Too Large response, **63**
 - client errors, **19**
 - Content-Encoding field, **247**
 - detect browser in use, **323**
 - detect clients, **237**
 - detect malicious redirection, **175**
 - detect related packets, **325**
 - detect web client, **9, 99**
 - detect web server, **9, 237**
 - determine web server software in use, **45, 324**
 - dissector not applied, **43**
 - end-of-field marker, **175**
 - error response codes, **19, 63**
 - exporting objects, **26, 176, 179**
 - Host* field, **14, 15**
 - http contains "GET"*, **176**
 - http.host* display filter, **14**
 - http.request* display filter, **14**
 - http.request* display filter, **170**
 - http.request.uri* field, **41**
 - http.request.uri* display filter, **17**
 - http.response* display filter, **13**

- http.response* field, **57**
- http.response.code* field, **46**
- http.response.code* filter, **19, 247**
- http.time* column, **243**
- http.time* field, **25**
- locate error responses, **19**
- locate *GET* requests, **167**
- locate slowest server, **328**
- match requests to responses, **328**
- measure object download time, **305, 312**
- measure response time, **23, 244, 305, 309, 317, 329**
- object not located on server, **20**
- proxy autoconfiguration file, **41**
- reassemble objects, **25, 314**
- reassembly issues, **23**
- Referer* field, **246, 308**
- Request URI* field, **17**
- response codes, **19, 63**
- response time, **25, 44, 228**
- response time, ridiculously large, **230**
- Server* column, **247**
- server errors, **19**
- time since request* field, **24**
- Uniform Resource Identifier (URI)
 - requested, **17**
- User-Agent* field analysis, **40, 167**
- User-Agent* identifies browser, **227**
- web browser identification*, **229**
- HTTPS. *See also* TLS/SSL analysis
 - display filtering, **59**
 - graph bits/s rate, **61**
 - graph retransmission rates, **62**
- ignored frames, **77**
- Info* column, TCP handshake analysis, **21**
- Initial Round Trip Time (iRTT). *See* TCP
- interconnecting devices, **85**
- interface ID, **74**
- IO Graph
 - default items, **34**
 - graphing port usage, **61**
 - linked to Packet List pane, **34**
 - port-specific retransmissions, **62**
- IP
 - client IP address detection, **9, 39, 55, 219, 255, 281, 301, 323**
 - dual-stack clients, **282**
 - fragmentation filter, **33**
 - IPv6 address filters, **204**
 - server IP address detection, **9, 39, 57, 219, 281**
 - Total Length* field, **116**
- keyboard shortcuts, **49**
- load time, **129**
- manuf* file, **136**
- marked frames, **76, 208**
- Maximum Segment Size (MSS). *See* TCP
- Maximum Transmission Unit (MTU), **335**
- metadata in trace files, **73**
- Microsoft BITS, **168**
- Microsoft Live Tiles fetch, **167**
- name resolution
 - enabling temporarily, **303**
 - HTTP *Host* name field, **239**
 - in the *Conversations* window, **125**
 - Only use the profile *hosts* file, **135**
- network forensics
 - detect non-standard port usage, **326**
 - exporting suspicious file, **177**
 - fake alert image, **177**
 - locate Command and Control (C2)
 - server, **47**
 - User-Agent* field, **40**
- NEWS.txt* file, **60**

350 Index

packet structures

IPv4, **335**

IPv6, **335**

packet-tcp.c file, **111, 131**

point the finger, **21**

port spanning concerns, **11**

preferences

Allow subdissector to reassemble TCP streams, **23, 176, 228, 230, 243, 304, 309, 312, 317, 328, 329**

name resolution, **125, 135, 303**

Only use the profile hosts file, **135**

protocol settings, **24, 25, 87, 228, 307**

Resolve MAC addresses, **136**

Resolve transport name, **136**

profiles

copying, **312**

creating a "Wireshark Workbook", **2**

managing, **135**

sharing buttons from, **289**

Profitap (TAP manufacturer), **11**

proxy.pac file, **42**

queuing device along a path, **92**

reassembly

Follow | HTTP Stream, **247**

Follow | TCP Stream, **247**

HTTP objects, **25, 314**

TCP streams, **25, 26, 47, 146, 176, 229, 230, 246, 304, 305, 307, 312, 314**

regular expressions (regex), **191, 284, 304, 306**

related packets indicator, **20**

relative sequence number. *See* TCP

RFC 6298, *Computing TCP's Retransmission Timer*, **114, 130**

RFC 7323, *TCP Extensions for High Performance*, **90**

RFC 7413, *TCP Fast Open*, **88**

RFC 793, *Transmission Control Protocol*, **88**

right-click

Apply as Column, **87, 196, 204**

Apply as Filter, **163, 169, 221, 249, 260, 261**

conversation filtering, **333**

copy field values, **163, 231**

manage profiles, **135**

Prepare a Filter, **159, 220, 221, 292, 293, 294**

protocol preferences, **87, 228, 309**

set/unset *Time References*, **117, 171**

scrolling

how to avoid, **14, 81, 207, 208**

when to use, **15**

Selective acknowledgment. *See* TCP *services* file, **136**

SMB/SMB2, response times, **10**

sorting

Answer RRs column, **286**

Bytes column, **162**

Calculated window size column, **28, 204, 205**

Content Type column, **176**

Conversation window, **13, 64**

DNS Time column, **10, 285, 286**

Filename column, **314**

HTTP Host column, **15, 302, 318**

HTTP Object List window, **179**

HTTP Server column, **324**

HTTP Time since request column, **25, 328**

Info column, **331**

iRTT column, **29, 91**

No. column (original sort order), **32**

Packets column, **260**

Port column, **100**

Shift count column, **89**

Size column, **307**

Source column, **55, 301**

Stream index column, **92, 109, 249**

- TCP *Stream index* column, **32**
- TCP *Time since request* column, **24**
- tcp.time_delta* column, **134, 204**
- User-Agent* column, **167**
- Spamhaus, **173**
- SSL and TLS analysis. *See also* TLS/SSL analysis
 - "ssl.*" display filter fields, **60**
- statistics
 - absolute start time, **125**
 - average bits/s rate, **27**
 - Capture File Properties* window, **77**
 - conversations, **13, 20, 30, 56, 80, 101, 109, 125, 141, 240, 257, 339, 341**
 - filtering, **64, 80, 125, 340**
 - graphing TCP, **81**
 - setting types, **126**
 - type button, **110**
 - using Find, **64**
 - data flow, **110**
 - endpoints, **56, 100, 158**
 - Ethernet addresses, **158**
 - following streams, **126**
 - graph port usage bits/s, **61**
 - graphing, **126**
 - IO Graph, **34, 61, 62**
 - name resolution in, **125**
 - OS on capture device, **77**
 - packets-per-second rate, **33**
 - port-specific retransmissions, **62**
 - UDP ports in use, **100**
- streams, counting UDP and TCP, **30**
- SYN packets. *See* TCP
- SYN/ACK packets. *See* TCP
- TAP (Test Access Port), need for a, **11**
- TCP
 - "phantom byte", **107, 108**
 - [SEQ/ACK analysis] section, **29, 91, 265**
 - Acknowledgment number* field, **269, 274**
 - Acknowledgment number* in SYN, **105**
 - Allow subdissector to reassemble TCP streams*, **23, 25, 44, 228**
 - analyzing without the handshake, **65, 341**
 - Calculated window size* field, **28, 90, 204, 341**
 - client and server different MSS values, **84**
 - connection lifetime measurement, **49**
 - connection order, **212**
 - connection termination, **48**
 - conversations, **20, 56**
 - detect highest sequence number values, **88**
 - detect missing TCP options, **83**
 - detect options supported, **21, 22, 40**
 - detect SACK support, **82**
 - detect TCP Timestamps support, **83**
 - detect Window Scaling support, **77**
 - detecting handshakes, **86**
 - detecting Maximum Segment Size (MSS) values, **83**
 - determine calculated window sizes advertised, **28**
 - determine missing bytes, **267, 272**
 - determine missing packets, **267**
 - determine missing sequence number values, **149**
 - different MSS sizes at client and server, **40**
 - dissector code, **111**
 - Duplicate ACKs, **64, 115, 258, 260, 269, 273, 274**
 - estimate number of missing packets, **272**
 - Fast Open (RFC 7413), **88**
 - Fast Retransmissions, **132, 138, 274, 275**
 - filter on *Flags* summary line, **39**
 - filter on SYN/ACKs, **79, 82, 83, 241, 330**
 - filter on SYNs, **71, 72, 86, 126, 211, 301, 330**

352 Index

- filter on SYN-SYN/ACKs, **83, 88, 212**
- filter on the *Flags* summary line, **16**
- FIN process, **48**
- Flags* summary line, **16, 71**
- follow streams, **47**
- graphing error rates, **34**
- graphing retransmissions, **62**
- half-open state, **48**
- handshake calculations, **29**
- handshake to determine capture location, **11**
- highest Duplicate ACKs in conversations, **64**
- identify if Selective ACK is in use, **265**
- identify window size multiplier, **265**
- iRTT*, arbitrary 3 ms value, **131**
- iRTT*, calculation, **29, 130**
- iRTT*, column use, **133**
- iRTT*, effect and out-of-orders, **130**
- iRTT*, field value, **92**
- iRTT*, inflated values, **130**
- iRTT*, measuring fastest, **30**
- keep-alives, **248, 249, 250, 342**
- low window sizes advertised, **205**
- Maximum Segment Size (MSS), **265, 335**
- missing SACK support, **330**
- missing the handshake, **72, 104, 105**
- MSS missing from SYN or SYN/ACK, **83**
- MSS size supported, **40**
- Next sequence number* field, **108, 119, 148, 266, 342**
- options listed at IANA, **83**
- out-of-order incorrect designation, **138**
- out-of-order packets, **128, 132, 139, 140**
- phantom byte, **343**
- port numbers in use, **20**
- preference settings, **23, 24, 43**
- Previous segment not captured*, **272**
- reassembling streams, **246**
- relative acknowledgment numbers, **107**
- relative sequence numbers, **87, 104, 106, 111, 266**
- Retransmission Time Out (RTO) timer, **42, 113**
- retransmissions, **42, 132, 275**
- retransmissions of SYN or SYN/ACK, **72**
- RFC 7413, *TCP Fast Open*, **88**
- Round Trip Time (RTT), **72**
- RST/ACK packets, **48**
- SACK permitted, **330**
- Segment Len(gh)* field, **85**
- Selective ACK Left Edge/Right Edge, **268, 270, 272**
- sequence numbering, **108**
- shift count, **90**
- starting window size, **330**
- stream count, **101**
- stream.index* field, **30, 64**
- SYN or SYN/ACK with data, **88**
- SYN packet display filter, **16, 21, 39**
- SYN/ACK packet display filter, **22**
- TCP Zero Window segment, **205**
- tcp.analysis.initial_rtt* field, **29**
- tcp.analysis.keep_alive* filter, **245, 248**
- tcp.dstport* field, **55**
- tcp.flags.fin==1 || tcp.flags.reset==1* filter, **92**
- tcp.window_size_scalefactor*, **341**
- termination with FIN, **92**
- termination with Reset, **92**
- true sequence numbers, **104**
- Window Scaling option detection, **79**
- window size multiplier, **65, 89, 330**
- time
 - arrival time in *Frame* section, **74**
 - auto-changing column setting, **118**
 - calculate average HTTP response time, **244**
 - calculate the average *iRTT*, **241**
 - detect slowest server, **328**
 - DNS resolution, **187**
 - dns.time* measurement, **313**
 - Epoch Time* in *Frame* section, **75**
 - http.time* field, **309**
 - http.time* measurement, **313**

- in *Frame* section, **73**
- iRTT measurement, **231**
- lousy HTTP response time, **230**
- measure DNS response time, **285**
- measure keep-alive intervals, **248**
- measure the *iRTT*, **265**
- set/unset* Time References, **171**
- smb.time* measurement, **313**
- smb2.time* measurement, **313**
- TCP delta, **249**
- time delta from previous captured frame* in *Frame* section, **75**
- time delta from previous displayed frame* in *Frame* section, **75**
- Time Reference* frames, **76, 117, 171**
- time shift in *Frame* section, **75**
- time since reference or first frame in *Frame* section, **76**
- Time since reference or first frame* in *Frame* section, **49**
- timestamping frames, **75**
- using to determine capture location, **239**
- Time Reference frame. *See* time
- tips
 - !=* and "combo fields", **326**
 - "most active" designations, **162**
 - .pcapng* for commenting, **76**
 - brackets around fields, **30, 65, 75, 213**
 - copying field values*, **163**
 - display multiple fields in one column, **75**
 - disregard *favicon.ico* issues, **326**
 - examine iRTT when troubleshooting, **92**
 - http* filter, **57**
 - http.request_in* filter, **57**
 - http.response_in* filter, **57**
 - jumping to packets, **145**
 - make buttons, buttons, buttons, **81**
 - multiple monitor use, **21, 290**
 - sanitize trace files before sharing, **304**
 - sending traces across time zones, **75**
 - separation of User-Agent values, **168**
 - sequence numbering formula, **109**
 - share your *dfilter_buttons* file, **289**
 - TCP preferences for HTTP response time analysis, **25, 26**
 - TCP preferences for TLS/SSL analysis, **25, 26**
 - use parentheses in display filters, **86**
 - viewing and sorting field values, **87**
 - when dissectors are not applied, **47, 50**
 - when you need a stream index column, **85**
- TLS/SSL analysis
 - Client Hello packets, **255**
 - handshake packets, **59**
 - HTTP/HTTPS IO Graph, **61**
 - Packet Bytes* window use, **256**
 - Server Hello packets, **264**
 - server_name* area, **255**
 - settings for, **25, 26**
 - ssl.handshake* display filter, **59**
 - TCP preference setting for, **230**
 - TCP statistics, **257**
 - tls.handshake* display filter, **60**
 - working in TCP header, **109**
- trace file capture date, **157**
- Tracewrangler, **303**
- typewriters, gray hairs and, **175**
- UDP
 - stream list, **30**
 - stream.index* field, **30, 31**
- User-Agent* field. *See* HTTP
- virus detection tool, identify the, **180**
- VSS-Monitoring Ethernet trailer bug, **116**
- white papers to read, **11**
- Window Scaling. *See* TCP