CHAPPELL UNIVERSITY

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

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.

Lab 1: Wireshark Warm-Up

Objective: Get Comfortable with the Lab Process.

Completion of this lab requires many of the skills you will use throughout this lab book. If you are a bit shaky on any answer, take time when reviewing the answers on page 9 to ensure you have mastered the necessary skill(s).

Trace File: wwb001-http.pcapng

Skills Covered in this Lab

In this lab, you will have a chance to work with many key functions in Wireshark. The answers to this lab demonstrate how to use functions, including, but not limited to:

- Perform address detection
- Measure DNS response time
- Add and sort columns to detect the highest/lowest field values
- Determine the trace file capture location
- Analyze conversation statistics
- Apply display filters on request and response packet values
- Detect HTTP host names
- Filter on TCP flags with && (and) and == (eq)
- Use automatic display filter packet counts
- Compare TCP flag summaries in display filter results
- Detect HTTP request URI values with display filters
- Apply display filters to detect HTTP error responses
- Use the packet relationship indicator
- Evaluate port usage statistics in TCP conversations
- Identify TCP handshake option definitions
- Determine highest and lowest HTTP response times
- Change TCP preference settings
- Reassemble HTTP objects

Lab 2: Proxy Problem

Objective: Examine issues that relate to a web proxy connection problem.

Trace File: wwb001-pacing.pcapng

Skills Covered in this Lab

In this lab, you will have a chance to work with many key functions in Wireshark. The answers to this lab demonstrate how to use functions, including, but not limited to:

- Filter on SYN packets only
- Perform Maximum Segment Size (MSS) analysis
- Analyze TCP connection capabilities
- Detect HTTP *User-Agent* field values
- Detect HTTP request URI values
- Use the *Do not call subdissectors for error requests* TCP preference setting
- Analyze HTTP response time
- Detect HTTP server information
- Analyze HTTP response codes
- Create display filter buttons to detect HTTP errors
- Follow a TCP stream to identify an application
- Determine which host terminated a connection
- Use the Time Reference feature to measure delta times

Lab 3: HTTP vs. HTTPS

Objective: Analyze and compare HTTP and HTTPS communications and errors using inclusion and field existence filters.

Trace File: wwb001-httpvshttps.pcapng

Skills Covered in this Lab

In this lab, you will have a chance to work with many key functions in Wireshark. The answers to this lab demonstrate how to use functions, including, but not limited to:

- Use an inclusion filter
- Use the *Conversations* window to analyze application use
- Apply "field existence" filters
- Compare related packet filters (`http.response_in` and `http.request_in`)
- Obtain packet counts based on port number filters
- Use Wireshark's autocomplete feature to filter on the TLS/SSL handshake
- Graph and compare port usage in a trace file
- Graph and compare protocol errors in a trace file
- Filter on HTTP errors
- Determine which TCP conversation contains the highest number of Expert Information warnings
- Determine why the window size scaling factor is shown as *-1* (unknown) by Wireshark

Lab 4: TCP SYN Analysis

Objective: Filter on and analyze TCP SYN and SYN/ACK packets to determine the capabilities of TCP peers and their connections.

Trace File: wwb001-syns.pcapng

Skills Covered in this Lab

In this lab, you will have a chance to work with many key functions in Wireshark. The answers to this lab demonstrate how to use functions, including, but not limited to:

- Display TCP SYN packets only
- Analyze traffic coloring rules
- Analyze frame metadata
- Filter traffic based on the TCP Window Scaling option
- Filter traffic based on the TCP Selective Acknowledgment (SACK) option
- Filter traffic based on the TCP Timestamps option
- Filter traffic based on the TCP Maximum Segment Size value
- Limit conversation statistics to filters
- Analyze TCP SYN or SYN/ACK packets that are missing the MSS option
- Filter on TCP flags and network protocol
- Analyze true TCP sequence numbers
- Filter on data existence in frames
- Detect TCP Fast Open frames
- Determine the largest Window Scaling Shift Count
- Obtain the Initial Round-Trip Time (iRTT) between peers
- Determine whether TCP connections were terminated with FIN or RST

Lab 5: TCP SEQ/ACK Analysis

Objective: Examine and analyze TCP sequence and acknowledgment numbering and Wireshark's interpretation of non-sequential numbering patterns.

```
Trace Files:   wwb001-seqack1.pcapng
               wwb001-seqack2.pcapng
```

Skills Covered in this Lab

In this lab, you will have a chance to work with many key functions in Wireshark. The answers to this lab demonstrate how to use functions, including, but not limited to:

- Identify HTTP request packets
- Use DNS flags to filter on all DNS responses
- Count the TCP streams in a trace file
- Filter properly on DNS name values
- Learn how DNS name fields are constructed
- Learn when Wireshark's TCP relative sequence number starts at 0 vs. 1
- Follow TCP relative sequence numbers
- Learn how *Sequence/Acknowledgment number* fields increment during the TCP handshake
- Differentiate client services from server services
- Determine which direction data is flowing in a TCP conversation
- Learn to identify retransmissions based on sequence numbers
- Identify retransmissions triggered by the Retransmission Time Out (RTO) timer
- Learn how the RTO counter is calculated
- Learn how to locate a Wireshark bug in Bugzilla
- Examine frame size differences caused by Selective Acknowledgment (SACK) information
- Set and measure time using Time Reference frames

Lab 6: You're Out of Order!

Objective: Examine Wireshark's process of distinguishing between out-of-order packets and retransmissions and identify mis-identifications.

Trace Files: wwb001-ooo1.pcapng
wwb001-ooo2.pcapng

Skills Covered in this Lab

In this lab, you will have a chance to work with many key functions in Wireshark. The answers to this lab demonstrate how to use functions, including, but not limited to:

- Determine the conversation counts in a trace file
- Identify a client using a display filter
- Identify a server using a display filter
- Determine the out-of-order packet count using the *Expert Information* window
- Determine the out-of-order packet count using a display filter
- Determine TCP Initial Round-Trip Time (iRTT) values
- Learn how TCP adjusts the Retransmission Time Out (RTO) timer after TCP handshake issues
- Learn how Wireshark uses time to differentiate between an out-of-order packet and a retransmission
- Filter out a stream based on the *Stream index* field value
- Identify frames defined incorrectly as out-of-order packets
- Use `tcp.time_delta` to locate delays
- Configure your profile's *hosts* file for name resolution use
- Find initial conversation packets based on sequence and acknowledgment number values
- Identify where out-of-order and retransmission packets should have arrived in a trace file
- Identify out-of-order packets on the TCP *Time-Sequence* graph (*tcptrace*)

Lab 7: Sky High

Objective: Examine and analyze traffic captured as a host was redirected to a malicious site.

Trace File: wwb001-skyhigh.pcapng

Skills Covered in this Lab

In this lab, you will have a chance to work with many key functions in Wireshark. The answers to this lab demonstrate how to use functions, including, but not limited to:

- Examine arrival time metadata inside individual frames
- Use MAC address information to identify vendor products on the network
- Create a single filter to locate multiple MAC addresses
- Use DHCP to identify the name of network devices
- Locate the most active client for a specific protocol or application
- Identify IPv4/IPv6 addresses used by DNS servers
- Locate DNS errors
- Find the percentage of DNS error responses in a trace file
- Identify the client operating system based on HTTP User-Agent information
- Determine what search phrase is being used by a host
- Measure the time required to execute a search term
- Detect a malicious HTTP redirection process
- Identify when server responds to multiple host names
- Reassemble HTTP objects
- Determine what virus detection tool was running when a host was compromised

Lab 8: DNS Warm-Up

Objective: Examine and analyze DNS name resolution traffic that contains canonical name and multiple IP address responses.

Trace File: wwb001-dnswarmup.pcapng

Skills Covered in this Lab

In this lab, you will have a chance to work with many key functions in Wireshark. The answers to this lab demonstrate how to use functions, including, but not limited to:

- Determine the number of DNS packets in a trace file
- Measure DNS response time
- Identify multiple IP addresses associated with a single host name
- Detect DNS errors in a trace file
- Identify which IP address a client uses when multiple addresses are returned
- Determine how long a DNS client can cache DNS information
- Determine if recursive or iterative DNS queries are in use
- Identify the canonical name for a host

Lab 9: Hacker Watch

Objective: Analyze TCP connections and FTP command and data channels between hosts.

Trace File: wwb001-hackerwatch.pcapng

Skills Covered in this Lab

In this lab, you will work with many key functions in Wireshark including, but not limited to:

- Identify company relationships based on DNS traffic
- Determine the maximum Calculated Window Size offered by TCP peers
- Compare path latency using the Initial Round-Trip Time (iRTT) value
- Identify successful FTP logins (user names and passwords)
- Determine if Active, Passive, Extended Active, or Extended Passive mode connections are in use
- Filter on specific traffic types and list related stream index values
- Correlate packets with applications in a trace file

Lab 10: Timing is Everything

Objective: Analyze and compare path latency, name resolution, and server response times.

Trace File: wwb001-responsetime.pcapng

Skills Covered in this Lab

In this lab, you will have a chance to work with many key functions in Wireshark. The answers to this lab demonstrate how to use functions, including, but not limited to:

- Identify DNS retransmissions
- Analyze DNS query contents
- Detect DNS "naked domain name" resolutions[24]
- Measure DNS response time when retransmissions occurred
- Examine SOA information in unsuccessful IPv6 address resolution
- Identify the client browser in use
- Properly associate HTTP requests and responses
- Measure extreme HTTP response times
- Determine the best path latency time between a client and server
- Identify how caching can improve browsing performance

[24] A domain name without the *www* or other subdomain preceding it is considered a “naked domain name.” Wow – that could have been something skanky, eh?

Lab 11: The News

Objective: Analyze capture location, path latency, response times, and keep-alive intervals between an HTTP client and server.

Trace File: wwb001-thenews.pcapng

Skills Covered in this Lab

In this lab, you will have a chance to work with many key functions in Wireshark. The answers to this lab demonstrate how to use functions, including, but not limited to:

- Identify the name of an HTTP server without using DNS
- Determine if the capture was taken closer to a client or a server
- Count the connections required to load a web page
- Calculate the average Initial Round-Trip Time (iRTT) between hosts
- Calculate the average HTTP response time in a trace file
- Export filtered columns to *.csv* format
- Reassemble a stream to identify a client's interest
- Identify a content delivery service based on a server's response
- Compare the *Expert Information* results with TCP reassembly on and off
- Determine the interval of TCP keep-alives

Lab 12: Selective ACKs

Objective: Analyze the process of establishing Selective acknowledgment (SACK) and using SACK during packet loss recovery.

Trace File: wwb001-tcpslesre.pcapng

Skills Covered in this Lab

In this lab, you will have a chance to work with many key functions in Wireshark. The answers to this lab demonstrate how to use functions, including, but not limited to:

- Identify the target server name in TLS/SSL communications
- Determine which TCP streams have transmission issues
- Differentiate the use of TCP SEQ/ACK analysis display filters
- Extract a trace file subset based on a filtered *Conversations* window
- Identify the Initial Round-trip Time (iRTT) of a TCP conversation
- Compare client and server TCP capabilities (MSS, Window Scaling, SACK)
- Locate the point at which packet loss began in a trace file
- Determine the relative sequence number at the start of packet loss
- Estimate the number of packets lost in a set
- Interpret TCP Selective Acknowledgment Left Edge/Right Edge blocks
- Use a display filter to count Duplicate ACKs
- Identify a missing segment range based on SACK Left Edge/Right Edge values
- Locate the point of packet loss recovery
- Examine how the acknowledgment number changes as packet recovery progresses
- Determine why Wireshark defined retransmissions as out-of-orders

Lab 13: Just DNS

Objective: Analyze, compare, and contrast various DNS queries and responses to identify errors, cache times, and CNAME (alias) information.

Trace File: wwb001-justdns.pcapng

Skills Covered in this Lab

In this lab, you will work with many key functions in Wireshark including, but not limited to:

- Filter based on DNS type
- Filter based on DNS response code
- Locate packets based on keywords
- Use regular expression display filters
- Measure DNS response time
- Count DNS Resource Records (RRs) in responses
- Identify DNS error responses
- Combine display filters for more accurate packet detection
- Analyze DNS cache time
- Follow CNAME details
- Correlate DNS error responses with an application
- Determine why SOA information is included in DNS responses

Before we start this section, I want to call your attention to Statistics | DNS*. Many of the answers to this lab can be obtained using this Statistics window, but I would like you to also obtain the answers by working with display filters. It's great practice and gives you a chance to examine the contents of DNS query and response packets.*

Laura

Lab 14: Movie Time

Objective: Use various display filter types, including regular expressions (regex), to analyze HTTP redirections, end-of-field values, object download times, errors, response times and more.

Trace File: wwb001-movietime.pcapng

Skills Covered in this Lab

In this lab, you will work with many key functions in Wireshark including, but not limited to:

- Display filter on TCP SYN packets
- Correlate a host name with an IP address in a sanitized trace file
- Determine if a trace file has been altered with Tracewrangler
- Measure HTTP object download time
- Locate packets based on end-of-field values
- Quickly determine the size of HTTP downloaded objects
- Identify HTTP hosts that are redirecting clients
- Measure, extract, and average HTTP response time
- Reassemble and examine HTTP objects
- Locate "File Not Found" responses based on the HTTP response code
- Identify the fastest responding HTTP servers in a trace file
- Use an inclusion display filter to locate all traffic sent to a set of addresses

Lab 15: Crafty

Objective: Practice your display filter skills using "contains" operators, ASCII filters, and inclusion/exclusion filters, while analyzing TCP and HTTP performance parameters.

`Trace File: wwb001-crafty.pcapng`

Skills Covered in this Lab

In this lab, you will work with many key functions in Wireshark including, but not limited to:

- Create and apply display filters using the `contains` operator
- Perform HTTP response time analysis
- Build and apply ASCII display filters
- Create and apply inclusion display filters
- Create and apply exclusion display filters
- Perform TCP connection analysis to determine which TCP options are supported
- Create display filters based on TCP flag settings
- Differentiate between true out-of-order packets and retransmissions
- Perform TCP Maximum Segment Size (MSS) analysis
- Compare and contrast IPv4 and IPv6 MSS/Maximum Transmission Unit (MTU) values

Lab 16: Pattern Recognition

Objective: Focus on TCP conversations and endpoints while analyzing TCP sequence numbers, Window Scaling, keep-alive, and Selective Acknowledgment capabilities.

Trace File: wwb001-pattern.pcapng

Skills Covered in this Lab

In this lab, you will work with many key functions in Wireshark including, but not limited to:

- Identify conversation/host counts using the *Conversations* and *Endpoints* windows
- Identify Selective Acknowledgment capability when TCP handshake information is available
- Work with a trace file that does not contain the Window Scaling setup process
- Interpret *-1* in the TCP *Window Size Scaling Factor* field
- Analyze TCP keep-alive packets
- Analyze the TCP sequence numbers of TCP FIN packets