CHAPPELL UNIVERSITY

Wireshark® Protocols and Network Forensics

V1.2A

SAMPLE COURSE OUTLINE

STUDENT MANUAL

COURSE OUTLINE

Section 1: Course Introduction and Resources
Section 2: Wireshark Essential Features for Network Forensics (Course Profile)
Section 3: Capture Methods and Capture Filters
Section 4: Customization - Wireshark Preferences
Section 5: Navigation, Coloring, and Reassembly
Section 6: Extract and Interpret Essential Trace File Statistics
Section 7: Focus on Traffic Using Display Filters
Section 8: TCP/IP Communications Overview
Section 9: Analyze Domain Name System (DNS) Traffic
Section 10: Analyze Address Resolution Protocol (ARP) Traffic
Section 11: Analyze Internet Protocol (IPv4) Traffic
Section 12: Analyze Internet Control Message Protocol (ICMP) Traffic
Section 13: Analyze User Datagram Protocol (UDP) Traffic
Section 14: Analyze Transmission Control Protocol (TCP) Traffic
Section 15: Analyze Hypertext Transfer Protocol (HTTP) Traffic
Section 16: Command-Line and 3rd Party Tools
Appendix A: Advanced Display Filters
Appendix B: Analyze Dynamic Host Configuration Protocol (DHCP) Traffic
Appendix C: Analyze File Transfer Protocol (FTP) Traffic
Appendix D: Analyze VoIP Traffic

LAB LIST

Lab 1: Create Your Network Forensics Profile
Lab 2: Identify Site Dependencies
Lab 3: Detect Suspicious Endpoint Locations with GeoIP Mapping
Lab 4: Apply Names to Suspect Hosts
Lab 5: Build a Coloring Rule to Differentiate DNS Traffic
Lab 6: Detect and Identify Malicious Downloads
Lab 7: Annotate Your Findings
Lab 8: Locate a Compromised Host with Emerging Threats Signature
Lab 9: Interpret a Russian Connection
Lab 10: Detect a Bot-Infected Host and C2 Traffic
Lab 11: Identify Exfiltration and Poisoning
Lab 12: Identify Blacklisted IP Addresses
Lab 13: Detect and Annotate Suspicious ICMP Traffic
Lab 14: Locate UDP-Based Scans
Lab 15: Detect Suspicious TCP Traffic
Lab 16: Apply, Inject, and Export TLS Secrets
Lab 17: Identify Interesting and Suspicious TCP and HTTP Traffic
Lab 18: Merge, Split, and Extract Suspect Traffic

Chappell University® Course:
Wireshark® Protocols and Network Forensics
Student Manual

ISBN13: N/A

Student Manual Part Number: WNF6-A v1.2A

Distributed worldwide for Chappell University through Protocol Analysis Institute, Inc.

For general information on Chappell University or Protocol Analysis Institute, Inc, including information on corporate licenses, updates, future titles, or courses, contact 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.

Protocol Analysis Institute, Inc.
dba Chappell University
720 N. 10th Street, #A352
Renton, WA 98057 USA

Chappell University®
720 N. 10th Street, #A352
Renton, WA 98057 USA
info@chappellU.com
www.chappellU.com

ADD-ONS

All-Access Pass The All Access Pass provides full-time access to recorded courses focusing on Wireshark functionality, network troubleshooting, network forensics, and more. Visit https://chappellu.com for more information.

WCNA Exam Vouchers Purchase in bulk to take the WCNA Exam (online proctored or at a Kryterion testing center worldwide). The WCNA Certification Exam focuses on analyzing packets and protocols, for network troubleshooting, optimization, and security. For more information, visit https://wcnacertification.com.

For more information, email info@chappellU.com.

About the Course Author
Chappell University Founder

Laura Chappell
Founder, Chappell University
Sr. Protocol Analyst, Protocol Analysis Institute, Inc.
Creator, WCNA Certification Program

Ms. Chappell researches, documents, and presents information on network protocols, analysis, Wireshark, network forensics, and interplanetary communications. Ms. Chappell is the creator of the WCNA Certification program (formerly referred to as the Wireshark Certified Network Analyst Certification program). Ms. Chappell also founded the original Wireshark University, Wireshark University Instructor Program, and Wireshark University Training Partner Program.

Ms. Chappell is often called in to analyze more complex network problems that require visibility into the communications system. Her clients include the U.S. Navy, IBM Corporation, Apple, Cisco Systems, Disney, U.S. Court of Appeals, United Bank of Switzerland, Australian High Tech Crime Centre, Capital One Financial Services, U.S. Armory, Hong Kong Police Department, Symantec Corporation, McAfee Corporation, Microsoft, Bank of San Francisco, Beth Israel Medical Center (Harvard), U.S. Joint Warfare Analysis Center, and the Federal Aviation Administration (FAA).

Ms. Chappell mixes onsite analysis services with live analysis training to develop self-sufficient IT teams within her client organizations.

As a member of the High Technology Crime Investigation Association (HTCIA) and the FBI's Infragard, Ms. Chappell has trained local, regional, national, and international law enforcement officers, as well as corporate security professionals on the methods and tools used to attack and defend networks. Ms. Chappell is also a voting member of the Institute for Electrical and Electronics Engineers (IEEE) (member since 1990).

Ms. Chappell's enthusiasm for her topics, sense of humor, and preference for working "live" during sessions have consistently ranked her as a top presenter at numerous industry conferences including Microsoft TechEd North America, Microsoft TechEd Europe, HP Technical Forum, Cisco Live, HTCIA International Conference, SharkFest, and InterOp.

In 2020, Ms. Chappell joined the Deep Packet Inspection Consortium as a Board Advisor. The DPI Consortium focuses on historical patent protection for DPI-based products and the fight against patent trolls.

In addition, Ms. Chappell is an active member of the Interplanetary Networking Special Interest Group focused on the documentation of deep space networking communication protocols. Ms. Chappell regularly lectures on the Deep Space Network (DSN) and Delay and Disruption Tolerant Networking (DTN) and is the IPNSIG Academy Lead.

In 2024, Ms. Chappell became a Board Member of the Interplanetary Networking Special Interest Group.

Ms. Chappell can be reached via email at *laura@chappellu.com*.

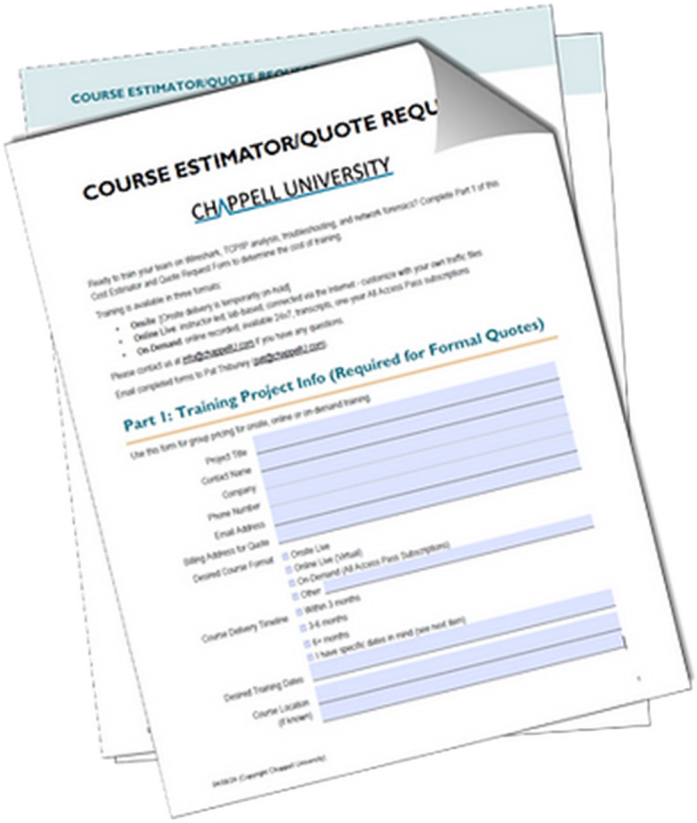
CHAPPELL UNIVERSITY

HOW TO PURCHASE A CUSTOM COURSE

Bring Laura Chappell online or onsite to speed up your team's trobleshooing and forensics processes.

Complete and submit the Course Estimator/Quote Request (Course Designer) document or simply let us know the following:

1. **Course Focus**: Do you want the course to focus on troubleshooting, network forensics, and/or general Wireshark functionality
2. **Course Length**: Minimum course length is 2 days. Laura's maximum course length is 10 days.
3. **Date Range**: Let us know which in which months you'd like the course delivered. We typically need at least 2 months advance preparation time.



Questions? Contact us at info@chappellu.com.

Table of Contents

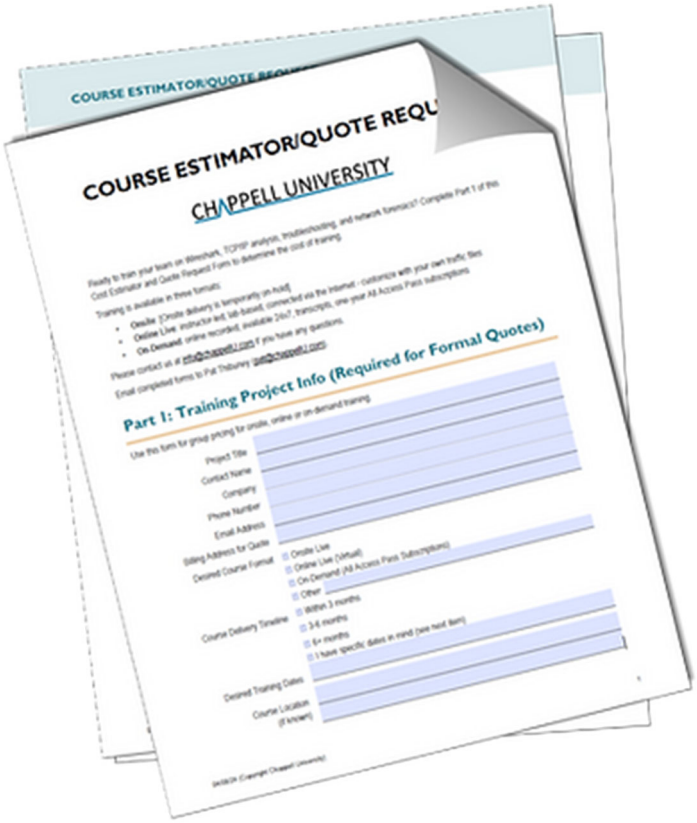
Section 14: Analyze Transmission Control Protocol (TCP) Traffic 217

HOW TO PURCHASE A CUSTOM COURSE

Bring Laura Chappell online or onsite to speed up your team's trobleshooing and forensics processes.

Complete and submit the Course Estimator/Quote Request (Course Designer) document or simply let us know the following:

1. **Course Focus**: Do you want the course to focus on troubleshooting, network forensics, and/or general Wireshark functionality
2. **Course Length**: Minimum course length is 2 days. Laura's maximum course length is 10 days.
3. **Date Range**: Let us know which in which months you'd like the course delivered. We typically need at least 2 months advance preparation time.



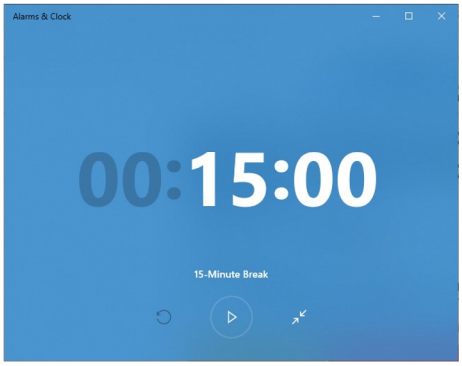
Questions? Contact us at info@chappellu.com.

Section 1: Course Introduction and Resources

[This page intentionally left blank.]

Course Introductions

- About this training format
- Course starting, ending and break times
- Student Guide format
- Course trace files
- Testing your systems
- Questions along the way



CHAPPELLUNIVERSITY®

Course Logistics

At this time, your instructor will provide detail about the training facility, course times and course format. In addition, you review the Student Manual format and items contained in your Student Kit.

This is the ideal time to test your system to ensure Wireshark can open trace files and to just scroll through the packets. Also examine the Interface List on the Wireshark Start Screen to be certain that Wireshark can see at least one interface on your system. Wireshark relies on packet capture drivers (such as Npcap and libpcap to capture traffic).

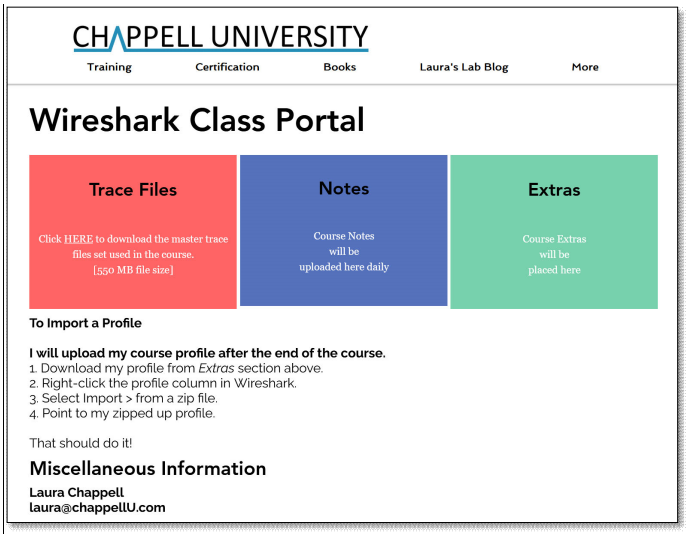
Course Content

This course focuses heavily on Wireshark functionality and TCP/IP communications – both normal and abnormal. This course provides the foundation knowledge required to capture, analyze, and identify suspicious network traffic with Wireshark.

Course Supplements

Supplements

Trace Files and other resources can be downloaded from your course portal page.



CHAPPELLUNIVERSITY®

Course Portal and Supplements

Information regarding your Course Portal Page was sent prior to the course start date.

If you haven't already downloaded the trace files for class, now is the time!

The Course Portal Page will be updated throughout the course. Check each day after class for the latest updates.

Note: Your Course Portal Page will remain open for 2 months after the course.

Prepare for the Course
The latest version of Wireshark should be loaded on your computer.

Do not use a Development Version unless your instructor explicitly requests that you do so.

O.S.C.A.R. Methodology

Obtain Information: Who, what, when, where…

Strategize: What evidence may exist; how to collect it (legally)

Collect Evidence: Capture location, capture integrity, Capinfos, copy

Analyze: Filter, extract, annotate

Report: Clear and concise, easy to understand (if possible)

Capture **Analyze**

The OSCAR methodology is defined in "Network Forensics, Tracking Hackers through Cyberspace", by Sherri Davidoff and Jonathan Ham

CHAPPELLUNIVERSITY®

OSCAR Methodology

Many organizations have adopted the OSCAR methodology as a guideline for network forensics procedures. The following is an example of using OSCAR from the network analysts' perspective.

O – Numerous hosts on the network are unable to communicate with internal or external resources. More and more users start to complain over time. The problem is getting worse.

S – Examining a network client system, you notice how slowly the machine is performing, often appearing to lock up completely. Attempting to connect to any device leaves you waiting with the feeling that the remote machine is unresponsive or you just can't get a connection for some reason.

C – Since Wireshark is loaded on the local host, you start capturing. You don't apply a filter – at this point, you want to capture all traffic. You try connecting to a few servers and have the same problems connecting. After stopping capture, you save the trace file and run Capinfos on it to obtain hash values for the file. Now you make a copy of the file and begin analyzing the copy.

A – The problem is immediately visible as broadcasts and multicasts flood the network. You note the source IP address of those broadcasts and multicasts and add comments in the trace when each new "storm" appears. You research the source of the traffic and find that disconnecting that host from the network alleviates the problem.

R – Your report details the broadcast/multicast-per second rate of traffic and provides details on the sender.

When to Use Wireshark

You suspect a system may be breached and you want to see:

- With whom does it communicate?
- Does it "phone home" to a suspicious host/location?
- What is currently being sent to/from the target system?
- Does a host send malformed protocol or application traffic?

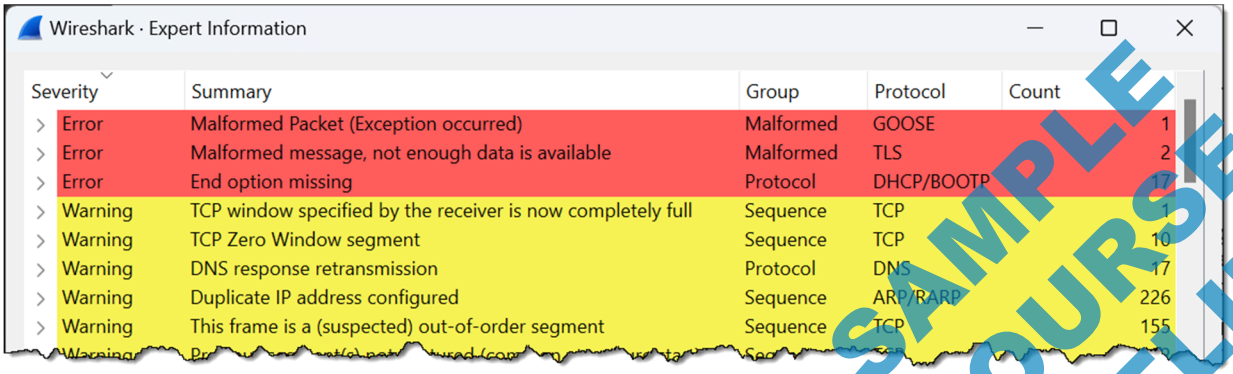
CHAPPELLUNIVERSITY®

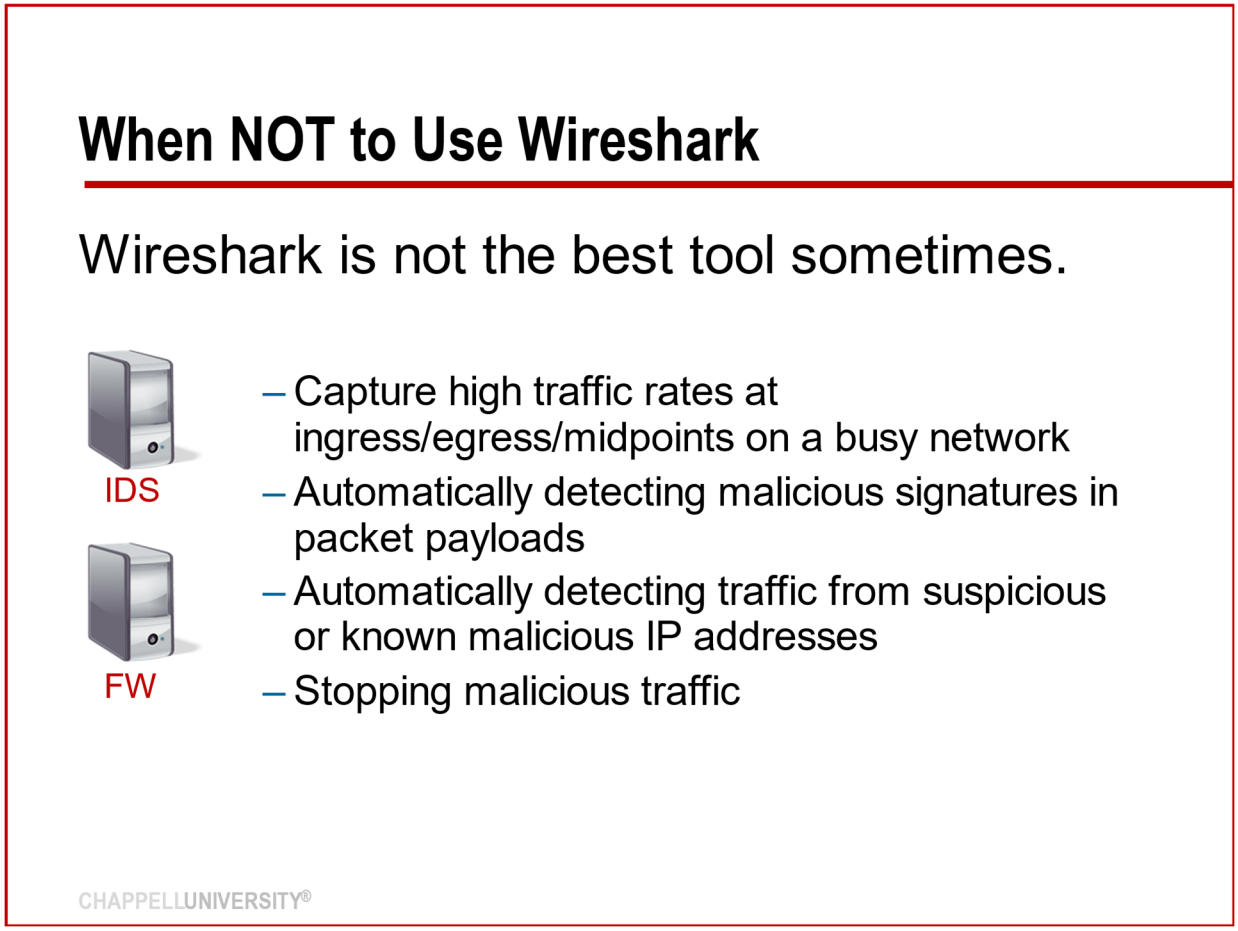
When to Use Wireshark

Wireshark is a graphical, passive packet analysis tool. It contains a packet capture tool called Dumpcap. Wireshark itself cannot capture traffic. It calls on Dumpcap to capture network traffic.

Wireshark is an ideal tool when you want to view the deciphered network traffic. Using Wireshark, you can identify nework conversations, protocols, and applications used by network devices.

You can also detect malformed frames, unusual protocols/applications, service refusals – possible indications of malicious traffic.





When NOT to Use Wireshark

Wireshark is not an intrustion detection tool.

Wireshark is not a firewall.

Wireshark may not keep up (via Dumpcap) when capturing a high-rate of traffic.

If you need to identify a variety of suspicious/malicious traffic on your network, use an IDS.

If you need to stop potentially suspicious/malicious traffic, use a firewall.

If you need to capture traffic at a very busy network point, consider using a tap orother capture tool.

Wireshark Network Forensics Examples

A user complains about strange pop-ups appearing on their system. You capture the traffic to/from that host to identify malicious connections.

You receive an alert from your IDS indicating that clear text personal information is seen leaving the network and a possible PayPal phish has been successfully executed. Your IDS has captured the traffic. You use Wireshark to analyze the traffic to/from the host as indicated in the IDS logs.

CHAPPELLUNIVERSITY®

Wireshark Network Forensics Examples

Wireshark is usually brought into the network forensics process by symptoms on the network or by detection via another tool, such as an IDS.

If users complain about performance or suspicious activity on a system, you need to decide how to capture the traffic.

If an IDS has indicated that there is a problem, it may have captured the traffic for you – this is an ideal situation.

What About Cloud-Based Capture?

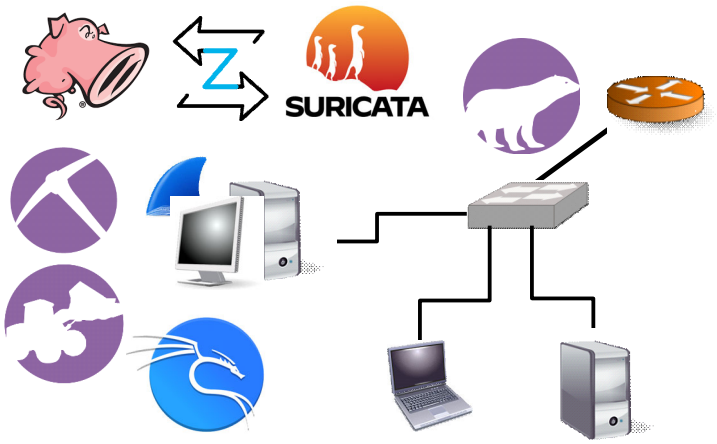
If you need to capture traffic to/from your cloud-based server, find out what solution is available from your cloud service provider. For example, Google offers "packet mirroring with IDS"[1] for Google Cloud customers.

If you are interested in traffic flowing between cloud servers and your network, consider capturing inside your network.

[1] https://cloud.google.com/blog/products/networking/using-packet-mirroring-with-ids

Tools for Network Forensics

- Wireshark
- Dumpcap/Tcpdump/Tshark
- Kali Linux
- Snort/Suricata
- Zeek
- NetworkMiner
- CAPloader
- PolarProxy



CHAPPELLUNIVERSITY®

Tools for Network Forensics

Obviously, Wireshark isn't the only tool to use for capturing and analyzing suspect traffic.

Note that Wireshark cannot capture traffic. When you run a capture from within the Wireshark graphical interface, Wireshark calls Dumpcap. We use Wireshark to do the analysis of the traffic.

Kali Linux is a penetration testing tool – one that is worth your time to learn.

Snort and Suricata are Intrusion Detection Systems (IDS). They use a set of rules to match traffic with known malicious/suspicious patterns (including blacklisted IP addresses).

Zeek is a network security monitor that can capture and interpret traffic, building incredibly detailed transaction logs.

We will look at NetworkMiner and CAPloader, both from Netresec, in the final section of the course.

PolarProxy is a transparent SSL/TLS proxy used to intercept and decrypt traffic. If you run a network forensics research lab, PolarProxy is a must-have tool.

emerging-all.rules

Reference: Emerging Threats Rules

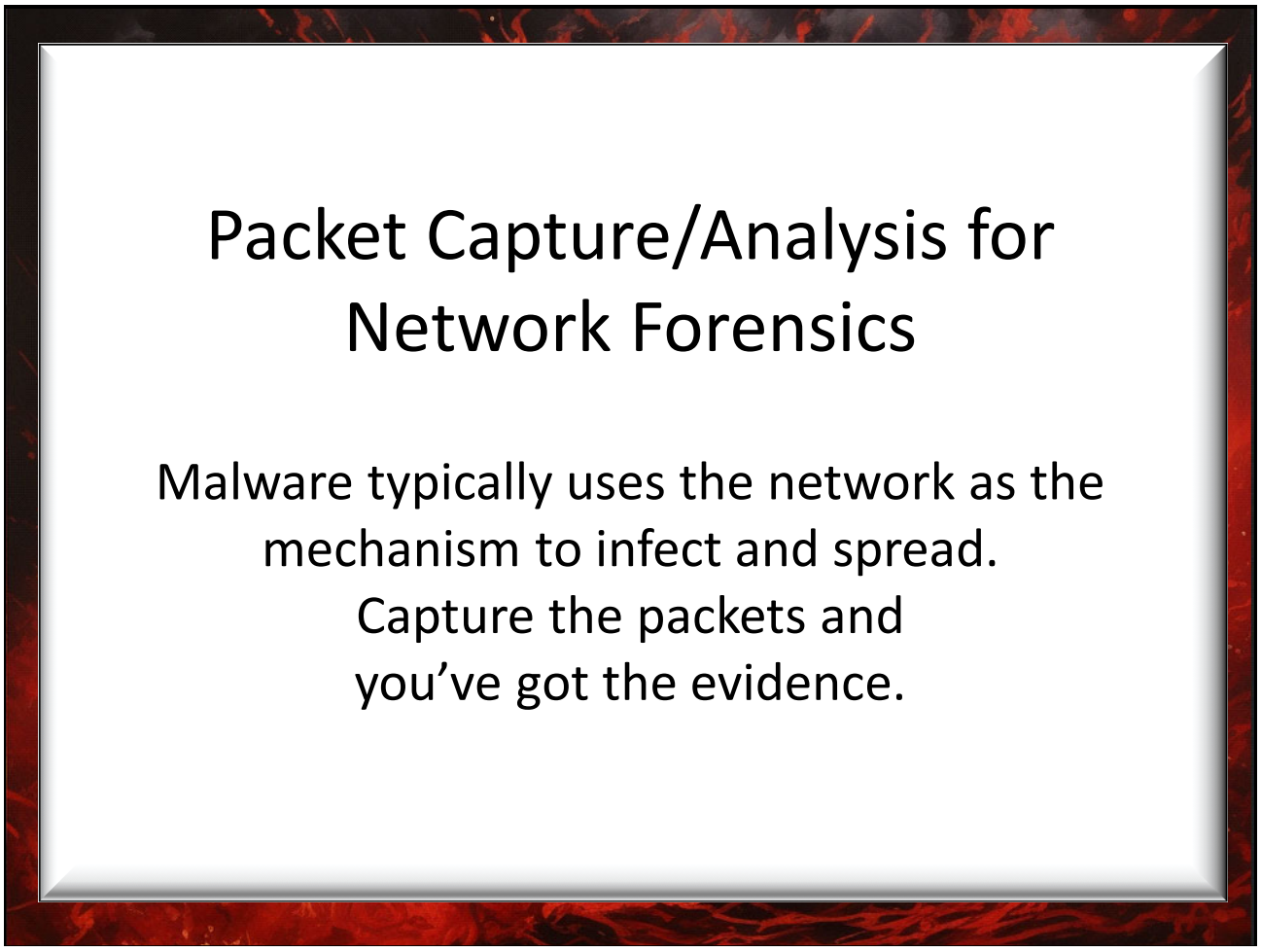
```
alert http $HOME_NET any -> $EXTERNAL_NET any (msg:"ET MALWARE Win32/Backdoor.Graphon Checkin Activity (GET)";
flow:established,to_server; http.method; content:"GET"; http.uri;
content:"/api/Values_V1/AuthAsyncComplete_V1?Identity="; fast_pattern; startswith; http.uri.raw; content:"=%3E";
http.header_names; content:"|0d 0a|Accept|0d 0a|Host|0d 0a|Connection|0d 0a 0d 0a|"; bsize:30;
reference:url,symantec-enterprise-blogs.security.com/blogs/threat-intelligence/harvester-new-apt-attacks-asia;
reference:md5,ff81a65150e318c1ffbeaba7a56bb09f; classtype:command-and-control; sid:2034224; rev:1;
metadata:attack_target Client_Endpoint, created_at 2021_10_18, deployment Perimeter, deployment SSLDecrypt,
former_category MALWARE, signature_severity Major, updated_at 2021_10_18;)

alert tcp any any -> [$HOME_NET,$HTTP_SERVERS] 445 (msg:"ET MALWARE [CISA AA21-291A] Possible BlackMatter Ransomware
Lateral Movement"; content:"|01 00 00 00 00 00 05 00 01 00|"; content:"|2e 00 52 00 45 00 41 00 44 00 4d 00 45 00 2e
00 74 00|"; distance:100; fast_pattern; detection_filter:track by_src, count 4, seconds 1;
classtype:command-and-control; sid:2034225; rev:2; metadata:attack_target Client_Endpoint, created_at 2021_10_19,
deployment Perimeter, deployment Internal, former_category MALWARE, malware_family BlackMatter, signature_severity
Major, tag Ransomware, updated_at 2021_10_19, mitre_tactic_id TA0040, mitre_tactic_name Impact, mitre_technique_id
T1486, mitre_technique_name Data_Encrypted_for_Impact;)

#alert http $EXTERNAL_NET any -> $HOME_NET any (msg:"ET DELETED Possible BlackByte Ransomware Encryption Key Inbound
(fake .png)"; flow:established,from_server; http.stat_code; content:"200"; file.data; content:!"|89 50 4E 47 0D 0A
1A 0A|"; startswith; pcre:"/^[\x20-\x7e\r\n]{0,13}[^\x20-\x7e\r\n]/si"; flowbits:isset,ET.httpget.png;
classtype:command-and-control; sid:2034213; rev:2; metadata:attack_target Client_Endpoint, created_at 2021_10_18,
deployment Perimeter, former_category MALWARE, malware_family BlackByte, signature_severity Major, tag Ransomware,
updated_at 2021_10_19;)

alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"ET MALWARE Observed Malicious SSL/TLS Certificate (MagnitudeEK
Associated)"; flow:from_server,established; tls.cert_subject; content:"CN=swissarny.store"; fast_pattern;
classtype:domain-c2; sid:2034226; rev:1; metadata:attack_target Client_and_Server, created_at 2021_10_19, deployment
Perimeter, former_category MALWARE, performance_impact Low, signature_severity Major, tag SSL_Malicious_Cert,
updated_at 2021_10_19, mitre_tactic_id TA0042, mitre_tactic_name Resource_Development, mitre_technique_id T1587,
mitre_technique_name Develop_Capabilities;)

alert tls $EXTERNAL_NET any -> $HOME_NET any (msg:"ET MALWARE Observed Malicious SSL/TLS Certificate (MagnitudeEK
Associated)"; flow:from_server,established; tls.cert_subject; content:"CN=rtpdn14.com"; fast_pattern;
classtype:domain-c2; sid:2034227; rev:1; metadata:attack_target Client_and_Server, created_at 2021_10_19, deployment
Perimeter, former_category MALWARE, performance_impact Low, signature_severity Major, tag SSL_Malicious_Cert,
updated_at 2021_10_19, mitre_tactic_id TA0042, mitre_tactic_name Resource_Development, mitre_technique_id T1587,
mitre_technique_name Develop_Capabilities;)
```

CHAPPELLUNIVERSITY®

Using Emerging Threats Rules with Wireshark

Emerging Threats (now ProofPoint) maintains a set of commercial and open rules for Snort and Suricata – two popular Intrusion Detection Systems.

If you are not familiar with reading Snort or Suricata rules, we will help you use an AI tool to interpret rules! This will be very interesting – there are times when AI really doesn't do a job well, but in this case, it's great!



The Network is Common Ground

Malware may infiltrate hosts, but eventually it will communicate on the network.

Ransomware infects and may periodically 'recheck' targets across the network.

Bots talk across the network to Command and Control (C2) servers.

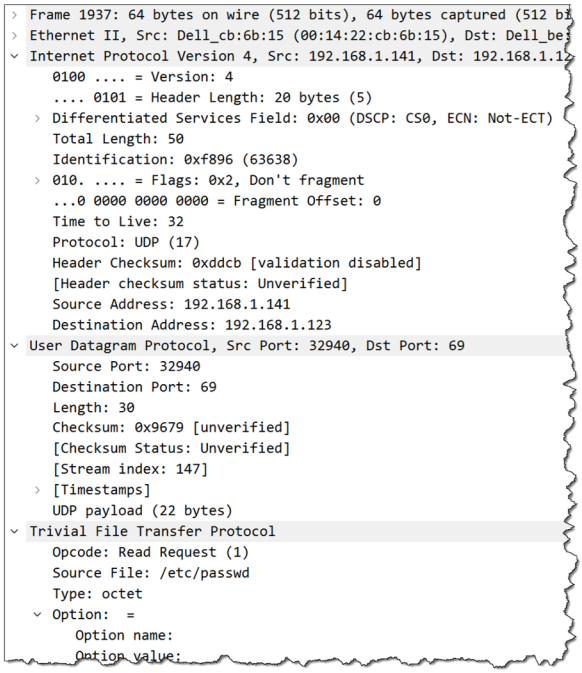
Scans run across the network.

The network is the common ground – except for rare instances, hosts get infected via traffic across the network.

Capturing that traffic is key.

What Can we Detect with Wireshark?

1. Unusual protocols or applications
2. Suspicious DNS requests/responses
3. Suspect IP addresses
4. Malformed frames
5. Download of suspect files
6. Remote execution commands
7. Suspicious tunneling
8. Non-standard port usage
9. TCP Resets with no data exchange
10. Large data flows outbound
11. Questionable endpoints
12. Unusual ICMP traffic
13. HTTP POST/PUT commands
14. Redirections
15. Persistent connections/heartbeats
16. Unencrypted sensitive information
17. Other known signatures



CHAPPELLUNIVERSITY®

Suspect and Malicious Traffic Detection with Wireshark?

The slide lists many individual items that we can detect with Wireshark.

It is important, however, to know what is normal on the network.

This includes:

- Protocols typically seen on the network
- Applications typically seen on the network
- Normal behavior of those protocol and applications
- IP addresses on the local network
- Flow of data on the network

About the Wireshark Protocols and Troubleshooting Course

Course starts with Wireshark essential features and details protocol behavior. A 3-day "Labs Only" version of the course exists to avoid overlapping of topics.

COURSE OUTLINE

Section 1: Course Introduction and Resources
Section 2: Wireshark Essential Features for Troubleshooting (Course Profile)
Section 3: Capture Methods and Capture Filters
Section 4: Customization - Wireshark Preferences
Section 5: Navigation, Coloring, and Reassembly
Section 6: Detect Application and Path Delays (Working with Time)
Section 7: Extract and Interpret Essential Trace File Statistics
Section 8: Focus on Traffic Using Display Filters
Section 8: TCP/IP Communications Overview
Section 10: Analyze Domain Name System (DNS) Traffic
Section 11: Analyze Address Resolution Protocol (ARP) Traffic
Section 12: Analyze Internet Protocol (IPv4) Traffic
Section 13: Analyze Internet Control Message Protocol (ICMP) Traffic
Section 14: Analyze User Datagram Protocol (UDP) Traffic
Section 15: Analyze Transmission Control Protocol (TCP) Traffic
Section 16: Analyze Hypertext Transfer Protocol (HTTP) Traffic
Section 17: Decrypting Traffic
Section 18: Command-Line and 3rd-Party Tools
Appendix A: Advanced Display Filters
Appendix B: Analyze VoIP Traffic
Appendix C: Analyze Dynamic Host Configuration Protocol (DHCP) Traffic
Appendix D: Analyze File Transfer Protocol (FTP) Traffic

LAB LIST

Lab 1: Create Your Troubleshooting Profile
Lab 2: Use GeoIP Mapping to Find an Issue
Lab 3: Build a Coloring Rule to Differentiate DNS Traffic
Lab 4: Detect and Differentiate Delays
Lab 5: Find Top Talkers/Protocols/Applications on a Network
Lab 6: Create and Use an I/O Graph to Spot Performance Issues
Lab 7: Practice Display Filtering
Lab 8: Catch DNS Errors and Slow DNS Responses
Lab 9: Find the Fault – Network Disconnects
Lab 10: Filter on Problem Addresses
Lab 11: Analyze and Color ICMP Traffic
Lab 12: Analyze UDP-based Multicast Streams and Queuing Delays
Lab 13: Use an IO Graph to Locate TCP Performance Issues
Lab 14: Determine the Cause of Slow Page Loading
Lab 15: Create a Button to Detect HTTP Error Responses
Lab 16: Export an HTTP Object (Carving)
Lab 17: Decrypt HTTPS Communications
Lab 18: Evaluate, Extract, and Capture with CLI Tools

CHAPPELLUNIVERSITY®

About the Troubleshooting 3-Day Labs-Only Course

The course you are currently in focuses on network forensics.

Both this network forensics course and the troubleshooting course begin with an overview of Wireshark functionality.

During the protocol deep dive, however, you are going to be looking for evidence of malicious applications, protocols, and bad actors on the network. In the *Wireshark Protocols and Troubleshooting Course*, the protocol deep dive focuses on network performance and identification of network problems.

If you are interested in troubleshooting, we recommend you consider taking the 3-day Labs-Only version of the *Wireshark Protocols and Troubleshooting Course*.

[This page intentionally left blank.]

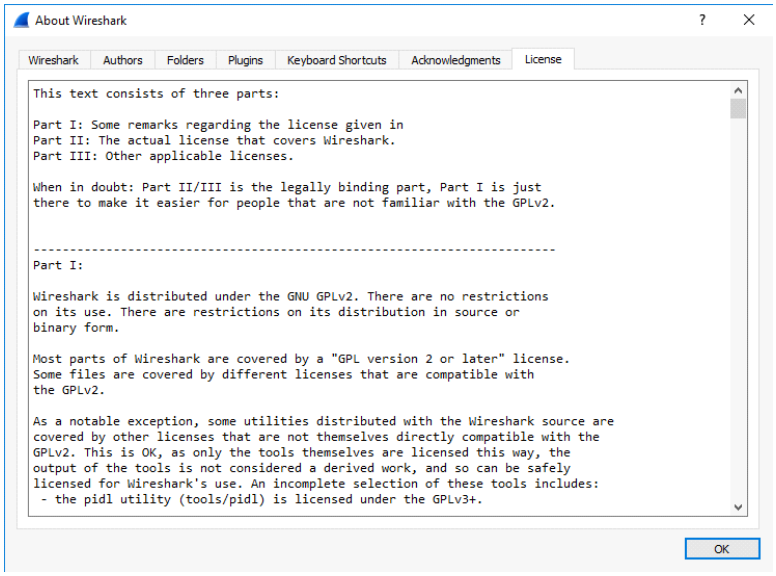
Section 2: Wireshark Essential Features for Network Forensics (Course Profile)

[This page intentionally left blank.]

The Wireshark License

GNU General Public License

Help | About Wireshark | License



CHAPPELLUNIVERSITY®

The Wireshark License

Wireshark is licensed under the GNU General Public License (referred to as the GNU GPL).

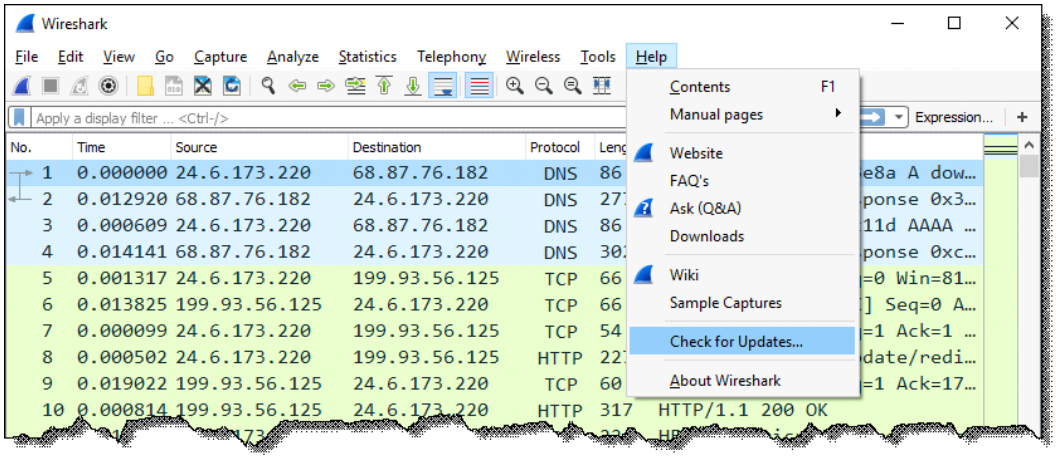
The GNU GPL is a widely accepted free software license originally defined by Richard Stallman for the GNU Project which created the GNU Operating System.

GNU is pronounced guh-*new*.

Information about the GNU GPL can be found at *www.gnu.org/licenses/licenses.html#GPL*.

Get the Latest Version of Wireshark

www.wireshark.org/download.html
Help | About Wireshark | Check for Updates



CHAPPELLUNIVERSITY®

Keep Wireshark Updated

Wireshark is updated often. Updates address bugs, functional enhancements, and security flaws. It is highly recommended that you stay up to date with the latest version of Wireshark.

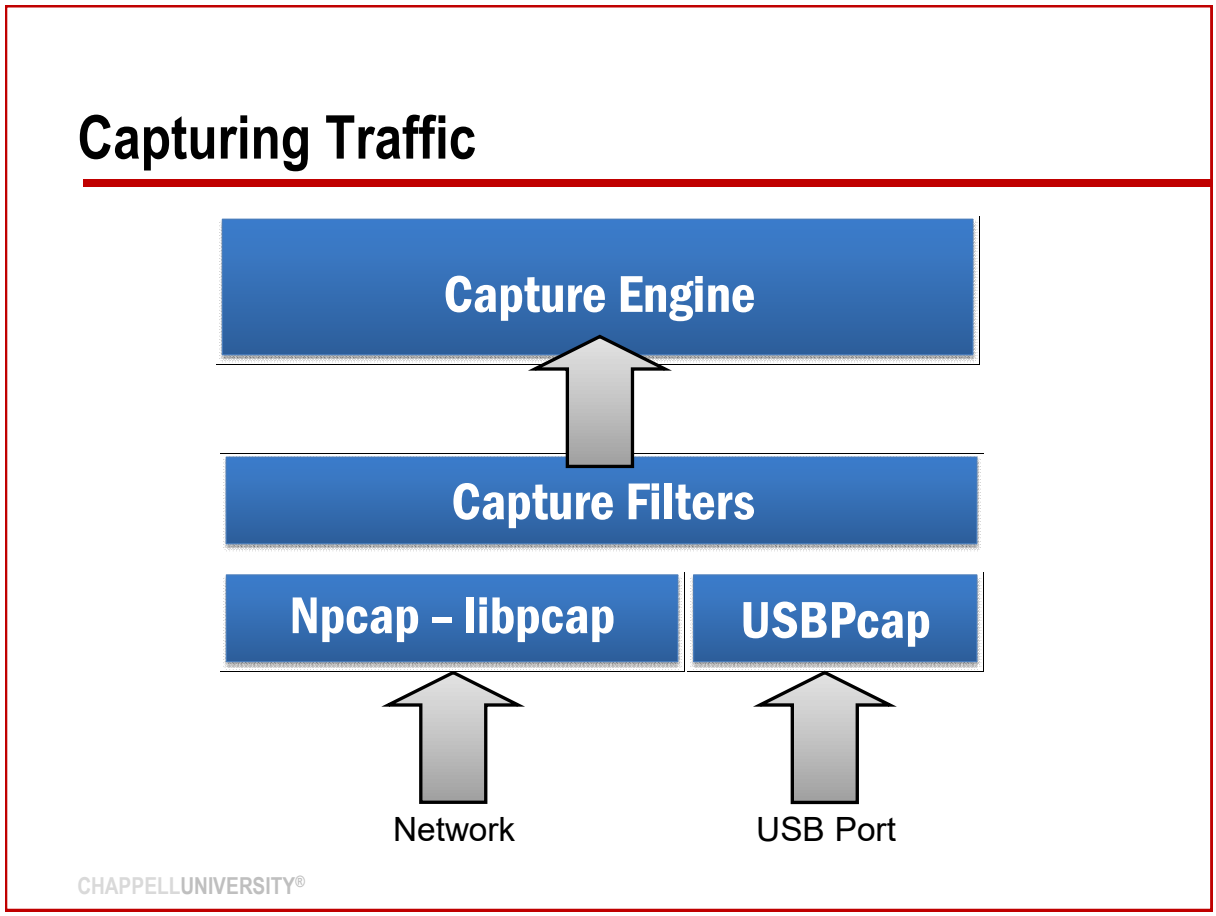
Wireshark automatically checks for updates. If you must manually check for updates[2], select **Help | Check for Updates** to determine if there is a newer version and launch the update process.

For a new installation, visit *www.wireshark.org/download.html*. As of Wireshark version 2, the Windows and Apple OS X installation processes are quite simple since these versions of Wireshark are available with an installer program.

Binary packages are available for most *NIX distributions. If a binary package is not available for your platform you can download the source and build it yourself. Refer to the Wireshark documentation (*www.wireshark.org/docs/wsug_html/*).

Wireshark also comes preinstalled on a number of forensic tool distributions, such as Kali Linux (*www.kali.org*), although it may not be the latest Wireshark version.

[2] If you are using earlier versions of Wireshark (such as early 2.x versions) or operating system other than Windows, you must manually check for and install updates.



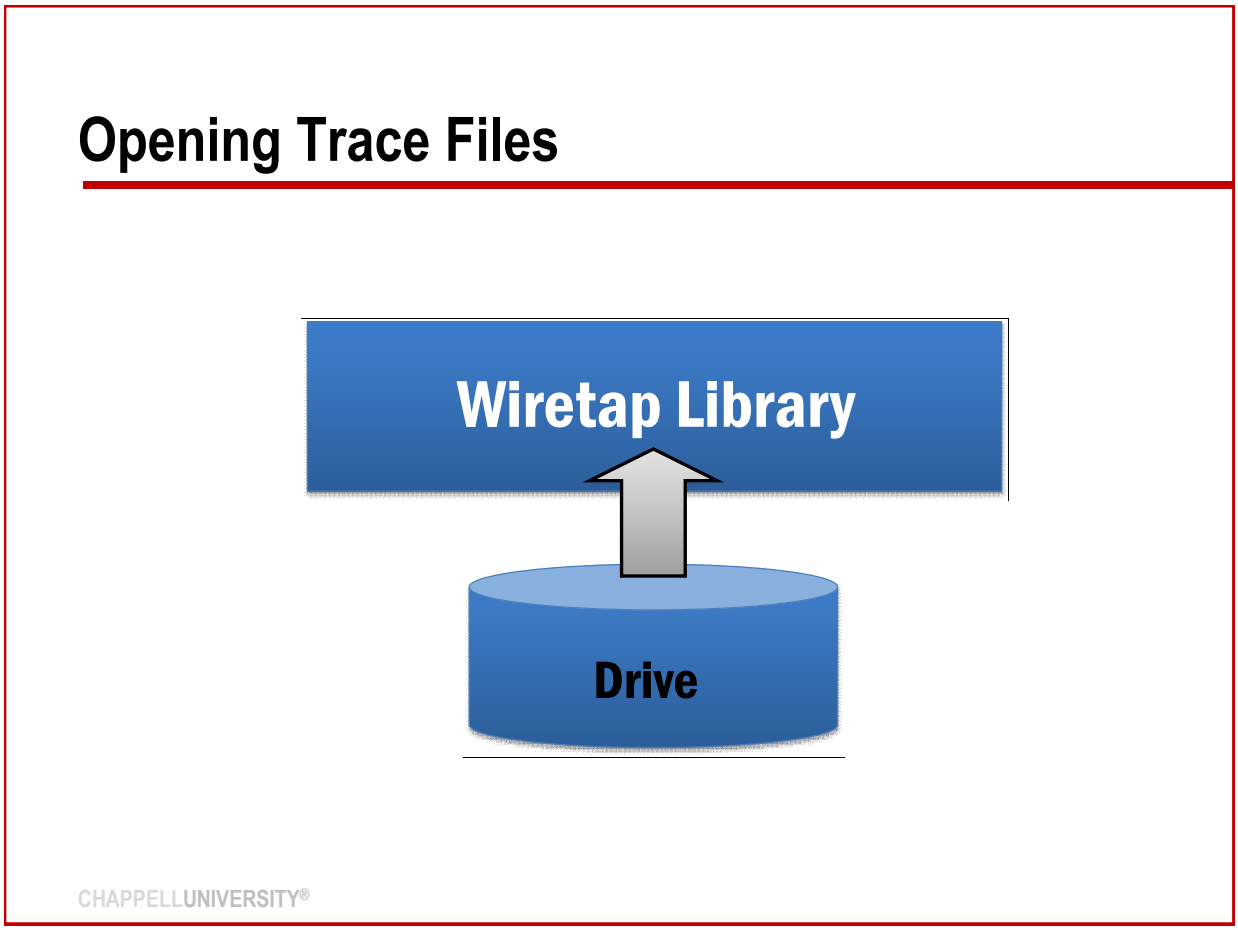
Capturing Traffic: Link-Layer Interfaces

When Wireshark is connected to a network, traffic is processed by either the Npcap or libpcap link-layer interface. In addition, the local adapter must support *promiscuous mode* operation to capture traffic that is destined to all hardware addresses, not just the local hardware address and the broadcast address. There is also an option to capture USB port communications.

Libpcap: The libpcap library is the industry standard link-layer interface for capturing traffic on *NIX hosts. For more information on libpcap, visit *www.tcpdump.org*.

Npcap: Npcap is the Nmap project's packet sniffing library for Windows. As of Wireshark v3, Npcap replaced WinPcap as a port of the libpcap link-layer interface. Npcap works on Windows 7 and later using the NDIS 6 Light-Weight Filter (LWF) API. Npcap can be restricted so only Administrators can sniff packets, offers a loopback packet capture, and can be installed in WinPcap Compatibility Mode, if desired. Visit *nmap.org/npcap/* for more information on Npcap.

USBPcap: This driver enables Wireshark to capture communications to and from local USB ports. For more information on USBPcap, see *desowin.org/usbpcap/* .



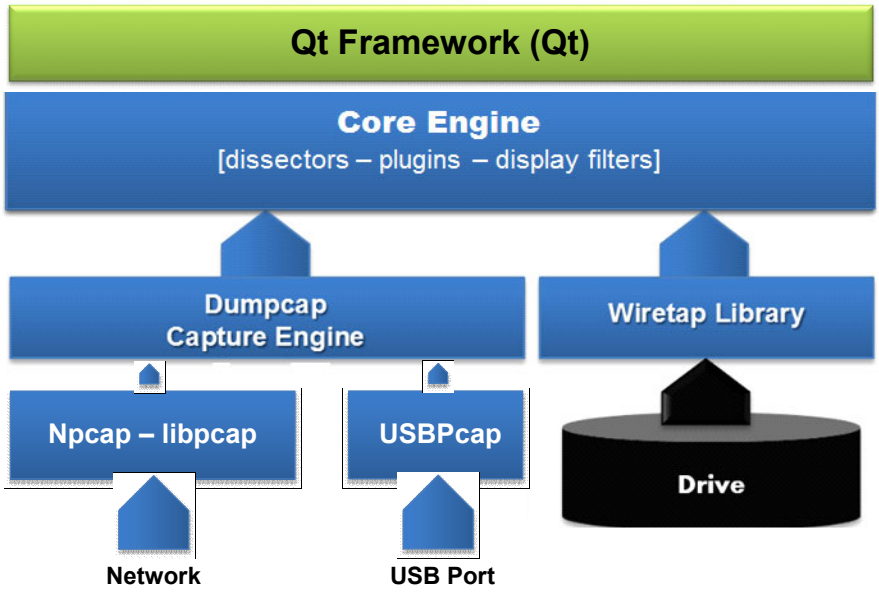
Opening Trace Files: the Wiretap Library

Npcap and libpcap interfaces are not used when saved trace files are opened. Opened trace files are processed through the Wireshark wiretap library, which enables Wireshark to read a variety of trace file formats including the following:

- Wireshark/tcpdump-libpcap
- Network Monitor, Surveyor, NetScaler
- Colasoft Capsa
- Novell LANalyzer
- Endace ERF capture
- TamoSoft CommView
- Ixia IxVeriWave .vwr Raw 802.11 capture
- Network Instruments Observer

To view the entire list of trace file formats in the Wireshark wiretap library, launch Wireshark and select **File | Open** and open the **Files of Type** drop down list.

Processing Packets



Note: The local adapter must support promiscuous mode to capture traffic addressed to other hosts' hardware addresses.

CHAPPELLUNIVERSITY®

Processing and Dissection of Packets

Whether packets are obtained from the network using Npcap or libpcap or from a saved trace file, they are processed in the core engine.

Core Engine

The core engine is described as the "glue code that holds the other blocks together."

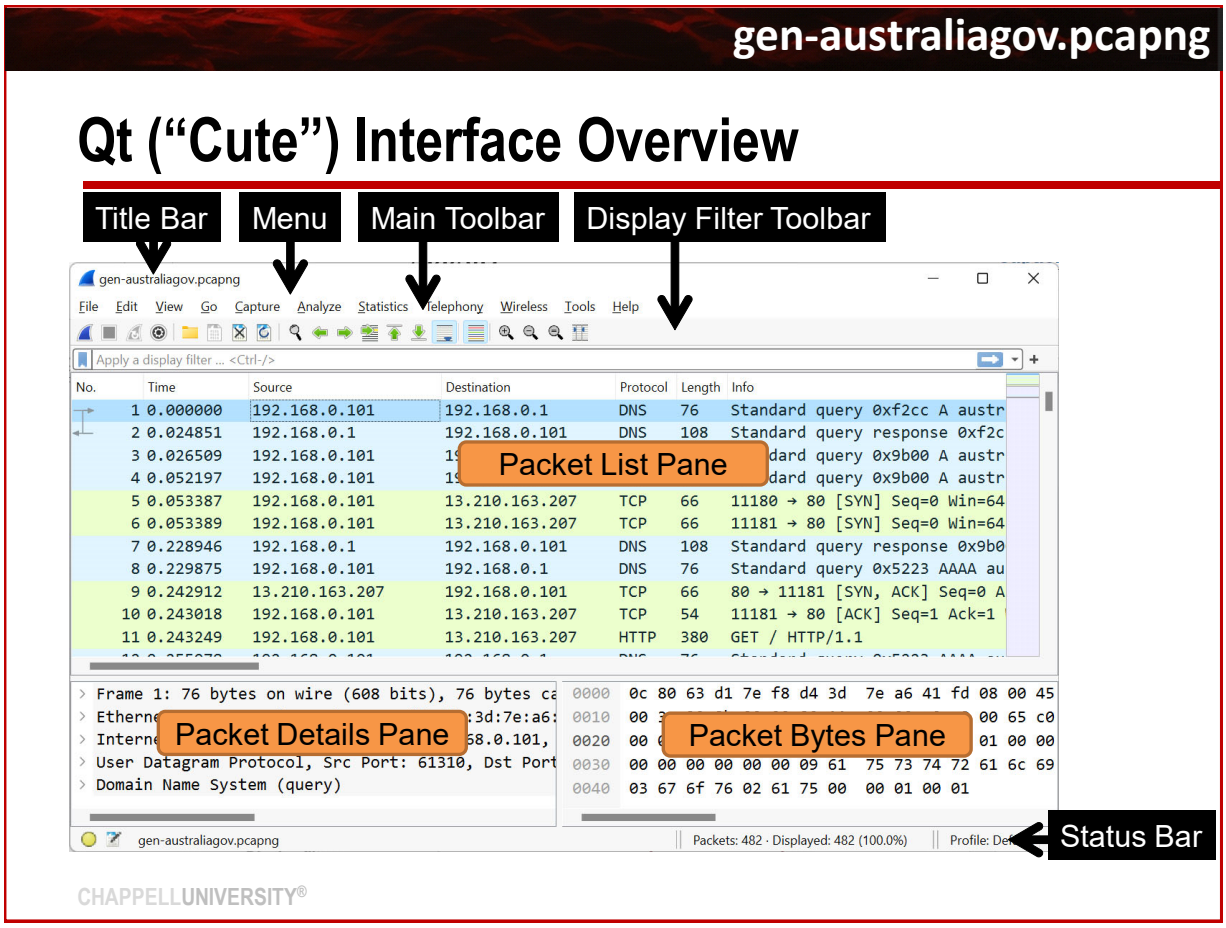
Dissectors, Plugins and Display Filters

Dissectors (decodes), plugins (special routines for dissection), display filters (used to define which packets should be displayed) are applied to the incoming traffic at this time.

The Qt Framework Provides the User Interface

The Qt (pronounced "cute") framework is the preferred option to provide the cross-platform graphical interface for Wireshark. With very few exceptions, you can move seamlessly from a Wireshark system running on one platform to a Wireshark system running on another platform with no problems. The basic interface elements are essentially the same.

Prior to Wireshark v2, GTK (Graphical Tool Kit) was used as the only graphical interface for Wireshark. In Wireshark v2, GTK was still available as an option. Support for GTK was removed in Wireshark v3.



The Qt Interface Overview

There are nine distinct sections in the default Wireshark look.

- Title
- Menu (text)
- Icon Toolbar (also referred to as the Main Toolbar)
- Display Filter Toolbar
- Packet List Pane
- Packet Details Pane
- Packet Bytes Pane
- Status Bar

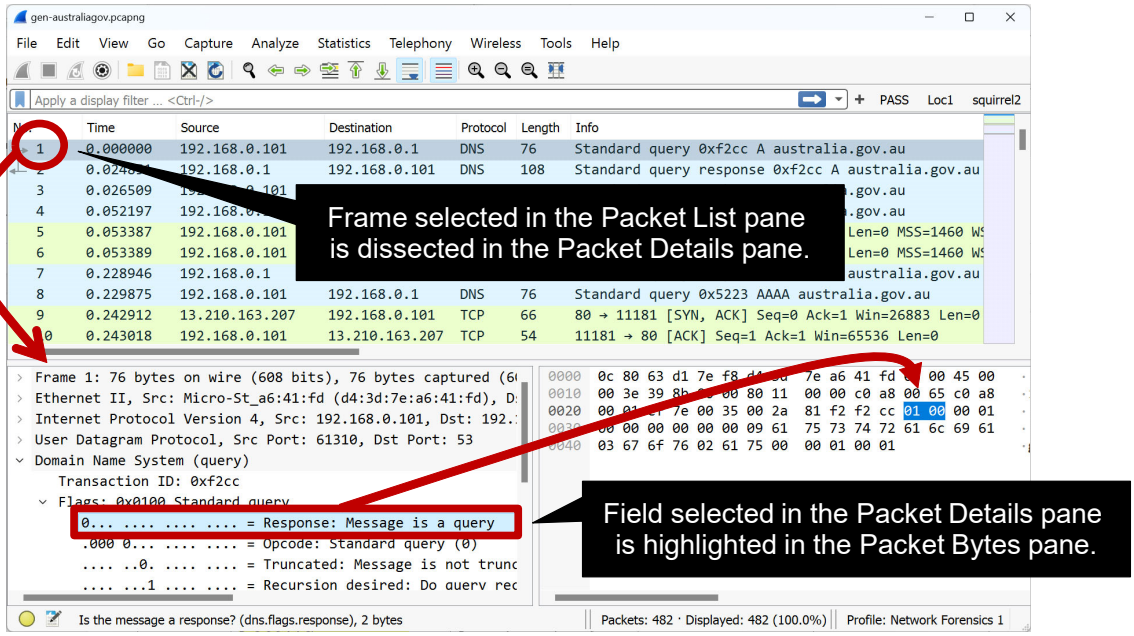
NOTE:

The new default layout places the Packet Details pane side-by-side with the Packet Bytes pane. This manual will show the layouts using the previous setting and the new setting. (To change the layout, select **Edit | Preferences | Layout**.)

gen-australiagov.pcapng

Linked Panes

The Packet List, Packet Detail, and Packet Bytes panes are linked.



CHAPPELLUNIVERSITY®

Using Linked Panes

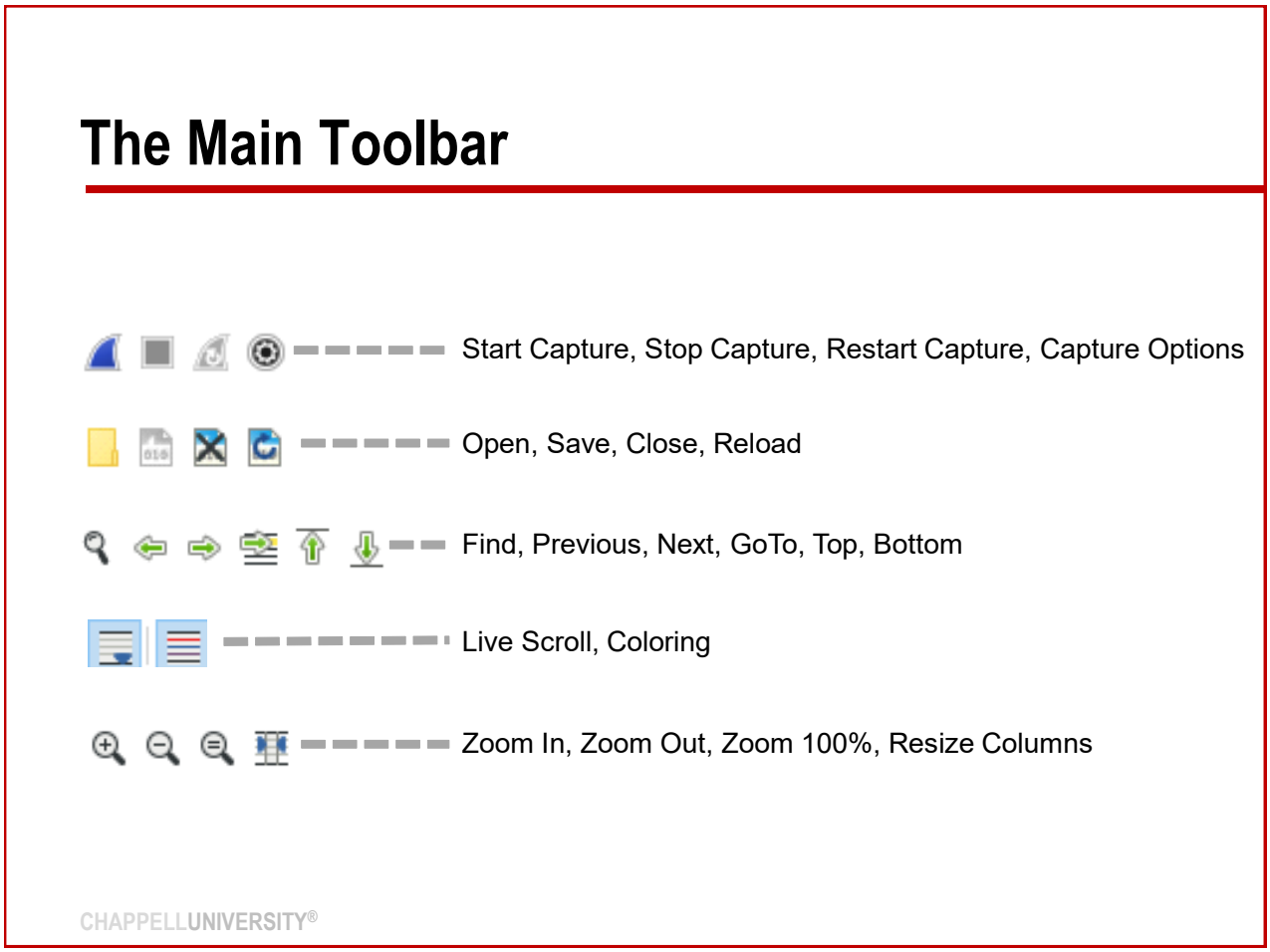
One of the most helpful features in Wireshark is the linking of the three panes – the Packet List pane, Packet Details pane, and Packet Bytes pane.

When you select a frame listed in the Packet List pane, the Packet Details pane displays the dissected frame.

When you select a field or summary line in the Packet Details pane, that field or summary section is highlighted in the Packet Bytes pane. Wireshark also displays field description, display filter syntax and field length information in the Status Bar.

HOT TIP

If you ever want to know the display filter syntax for a field, simply select that field in the Packet Details pane and look for the filter syntax in the Status Bar. In the example above, we selected the **DNS Response Flag** field in the Packet Details pane. The Status Bar indicates the field name is **`dns.flags.response`**.



The Main Toolbar

The Main Toolbar (also referred to as the “Icon Toolbar) provides access to many key Wireshark functions.

The instructor will go through a quick demonstration of using the icon toolbar to identify a capture interface, apply a display filter, and stop the capture process.

Tooltips provide a short definition of each icon toolbar button when you hover your mouse over an icon.

The Related Packets Indicator

	First packet in a conversation
	Part of the selected conversation
	Not part of the selected conversation
	Last packet in a conversation
	Request
	Response
	The selected packet acknowledges this packet.
	The selected packet is a duplicate acknowledgement of this packet.
	The selected packet is related to this packet in some other way, such as part of reassembly.

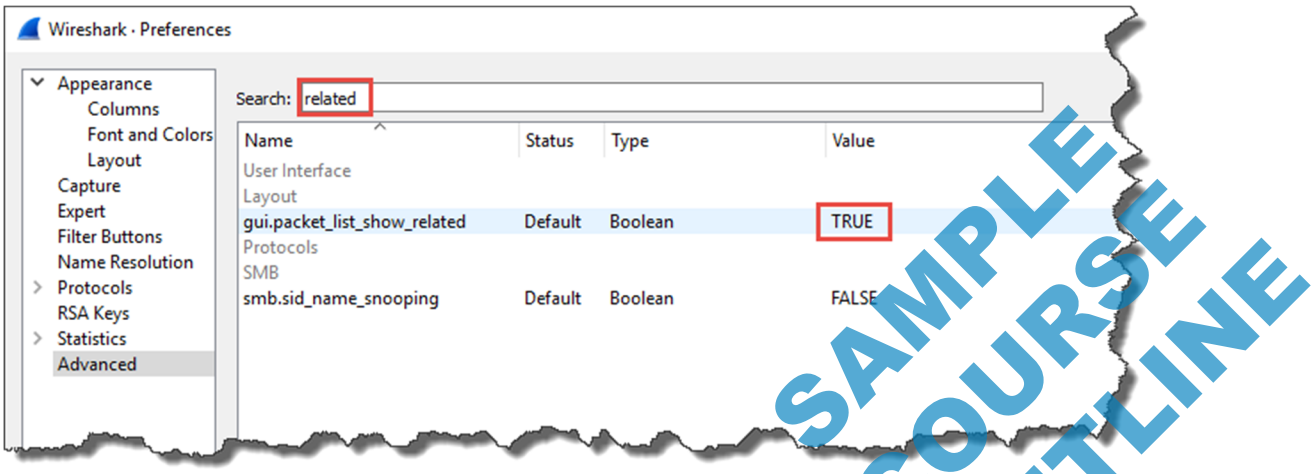
CHAPPELLUNIVERSITY®

The Related Packets Indicator

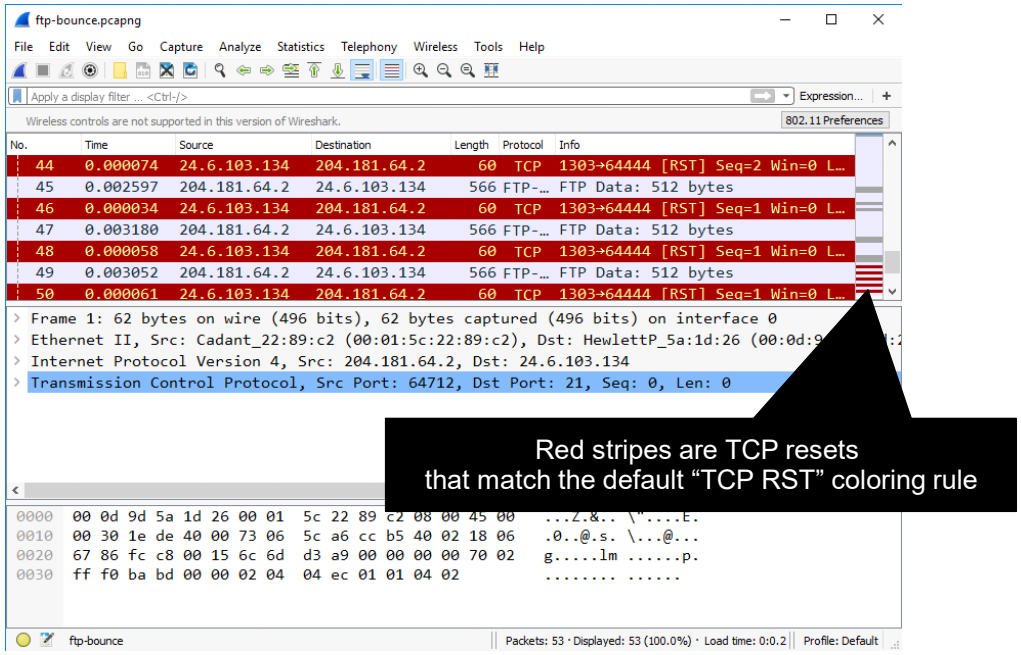
The Related Packets Indicator is part of the Number column.

When you select a packet, Wireshark adjusts the Related Packets Indicator to show other packets that are part of the conversation, related requests/responses, related acknowledgments, and more.

To disable the Related Packets Indicator, select **Edit | Preferences | Advanced** and type in "**related**" to locate the *gui.packet_list_show_related* setting. Change the value to **False**.



The Intelligent Scroll Bar



CHAPPELLUNIVERSITY®

Master the Intelligent Scroll Bar

The Intelligent Scroll Bar gives you a very tall, skinny view of the Packet List pane. The Intelligent Scroll Bar depicts the coloring seen in the Packet List pane so you can quickly locate areas of interest in your trace file.

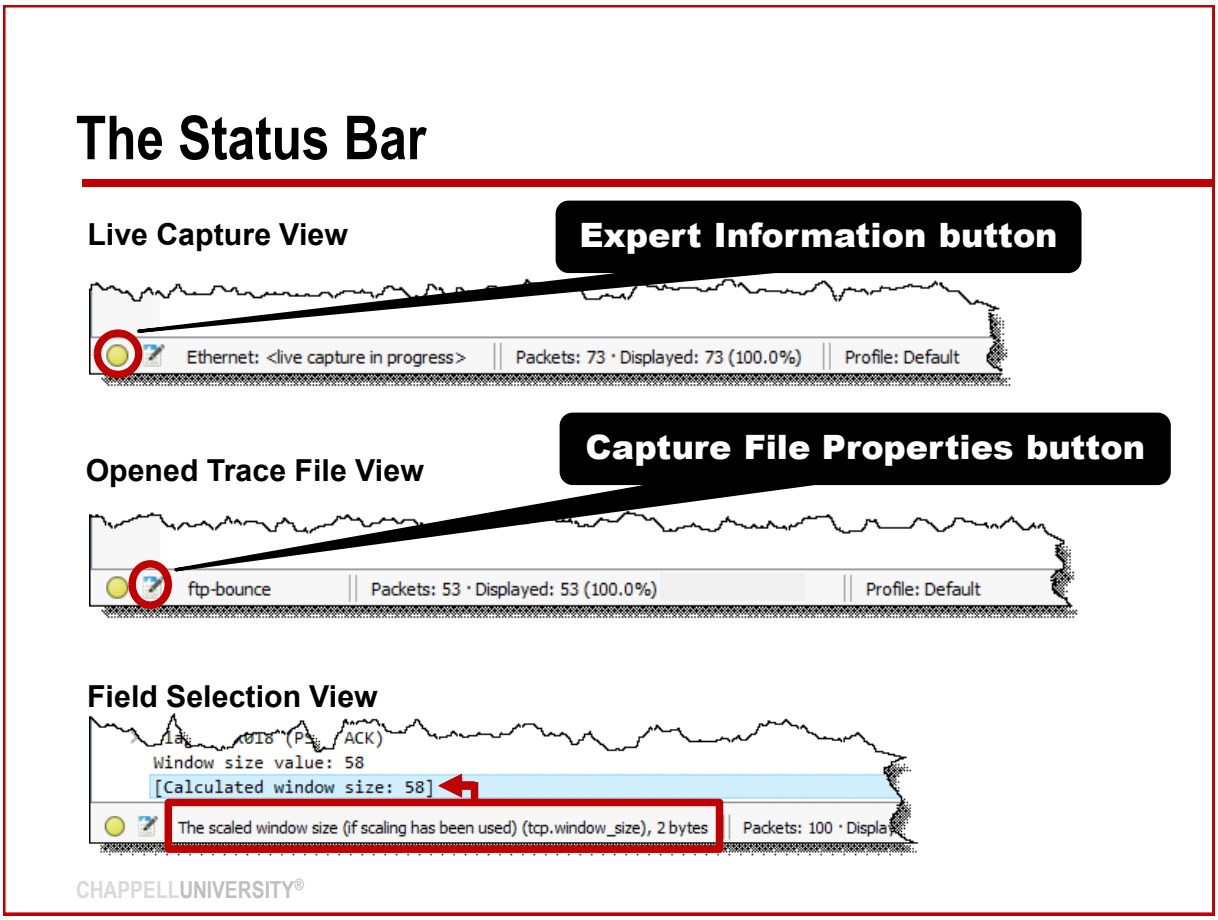
The Intelligent Scroll Bar does not appear during a live capture process. It will appear once the capture process is stopped.

There is limited space on the Intelligent Scroll Bar. When working with larger trace files, the Intelligent Scroll Bar will not display the coloring of the entire trace file. You may need to drag the thumb down to view the Intelligent Scroll Bar information for the entire file.

In the image above, we have opened *ftp-bounce.pcapng* and moved the thumb of the scrollbar down to a point where we can see red stripes in the Intelligent Scroll Bar. Those represent the TCP Resets seen in the trace file.

Disabling Wireshark’s coloring feature disables the Intelligent Scroll Bar.

The most efficient way to use the Intelligent Scroll Bar is to enhance your coloring rules so the spots of interest stand out on the Intelligent Scroll Bar.



The Changing Status Bar

The Expert Information button resides on the left side of the Status Bar—the coloring corresponds to the highest level of Expert Information message seen in the trace file. Click on this button to quickly identify numerous performance issues.

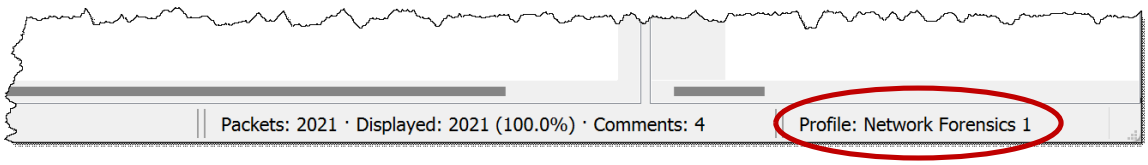
The content in two columns of the Status Bar changes to reflect details about the current operation.

- During capture the Status Bar displays the interface being used, the temporary file location and name and the file length as well as the total number of packets and the displayed and marked packet count.
- After opening a trace file, the Status Bar provides file information such as the total number of packets, displayed packet count, marked packet count (if applicable), and ignored packets (if applicable).
- When you select a field in the Packet Details pane, the Status Bar displays the field description, field name (used in display filtering, graphing and coloring rules), the length of the field and notes, if available.

Step 1: Create a Network Forensics Profile

Profiles contain:

- Preferences
- Capture Filters
- Display Filters
- Coloring Rules
- and more…



CHAPPELLUNIVERSITY®

First Step: Create Your Network Forensics Profile

Wireshark can be customized for specific analysis tasks by creating separate profiles. Profiles can contain their own capture filters, display filters, preference settings, coloring rules, etc.

Right-click on the profile column in the Status Bar to create a new profile or use **Edit | Configuration Profiles**.

As you customize your profile, Wireshark adds configuration files to your profile directory. Some of the files you may see include:

cfilters	saved capture filters used with this profile
dfilters	saved display filters used with this profile
dfilter_buttons	saved display filter buttons used with this profile
preferences	protocol and interface settings used with this profile
recent	toolbar and last directory settings used with this profile

Wireshark includes a default profile and some predefined global profiles. Global profiles (Bluetooth, Classic, and No Reassembly) offer customized settings for specific purposes.

We begin this section by creating a Network Forensics profile. We will work within this profile for the remainder of the course.

Hands-On Lab

Lab 1: Create Your Network Forensics Profile

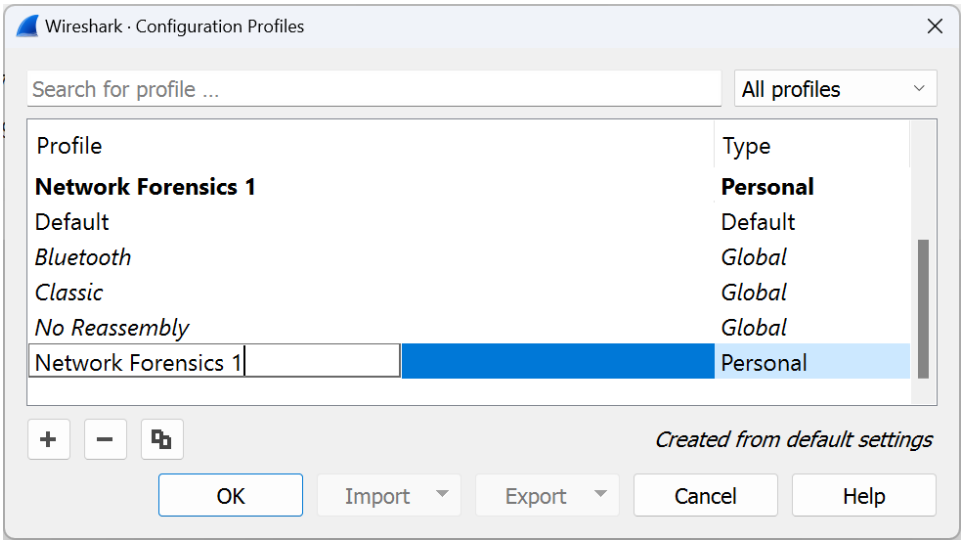
Overview You will create a profile and use this profile through the remainder of this course. The profile will eventually contain numerous filters, coloring rules and columns to speed up the forensic investigation process.

Lab Steps

Step 1 Open **Wireshark**.

Step 2 Right-click on the Status Bar **Profile** column and select **New**.

Rename your new profile **Network Forensics 1**.



Step 3 Click **OK**.

Your new profile name is shown in the right column of the Status Bar.

Step 4 Right click on the Profile column and select **Manage Profiles**.

Step 5 Your Network Forensics profile is selected. Profiles are contained in the personal configuration directory. This directory location is dependent on the operating system on which Wireshark was installed.

Click the **hyperlink** listed for your Network Forensics profile directory.

As you customize Wireshark (while working in the Network Forensics profile) you will see numerous configuration files appear in this directory.

Close the profile directory and return to Wireshark.

We will use this profile throughout the course.