



Laura Shupe

Script Kitties

Wireshark Runbook

14 February 2024

## Wireshark

### 1. Installation:

Windows:

- Download the installer from the official website: Wireshark Downloads
- Run the installer and follow the installation wizard.

Linux:

- Install Wireshark from the package manager:

```
sudo apt-get install wireshark
```

- Ensure that you have proper permissions to capture packets by adding your user to the wireshark group:

```
sudo usermod -aG wireshark <username>
```

macOS:

- Download the installer from the official website and follow the installation instructions.

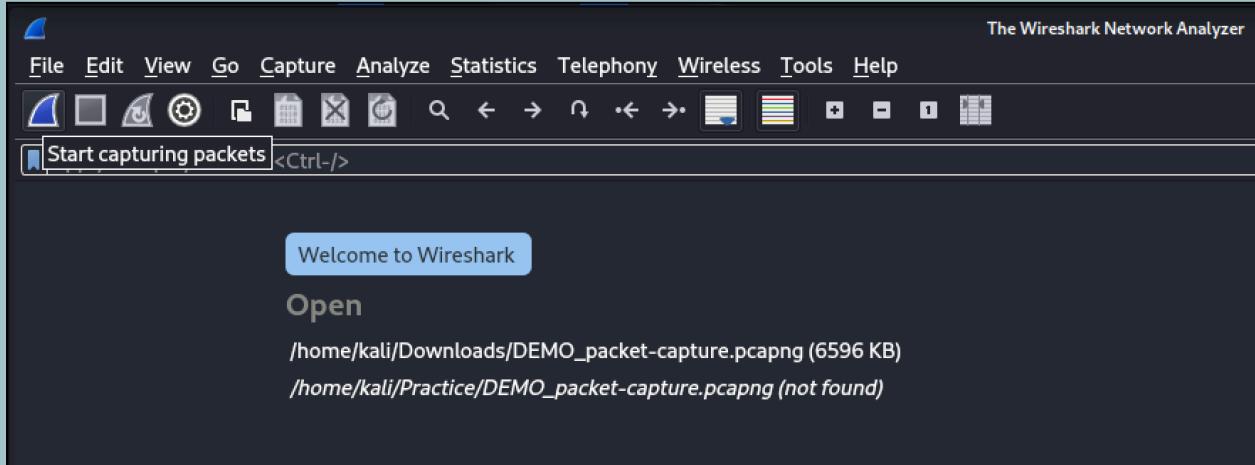
### 2. Basic Usage:

Start Wireshark:

- Open the Wireshark application from the desktop shortcut or terminal.

Capture Traffic:

- Click on the interface name to start capturing traffic.
- Select the appropriate network interface.
- Click "Start" to begin capturing packets.



### Stop Capture:

- Click "Stop" or use the keyboard shortcut (Ctrl + E) to stop capturing packets.

No.	Time	Source	Destination	Protocol	Length	Info
2410	4.818552340	172.31.8.66	172.31.13.81	TCP	66	50168 - 8443 [ACK] Seq=5859 Ack=1821328 Win=5421 Len=0 TSval=3633175670 TSecr=874477071
2411	4.818675527	172.31.8.66	172.31.13.81	TCP	66	50168 - 8443 [ACK] Seq=5859 Ack=1826278 Win=5421 Len=0 TSval=3633175670 TSecr=874477071
2412	4.855557944	172.31.8.66	172.31.13.81	TCP	66	50168 - 8443 [ACK] Seq=5859 Ack=1826341 Win=5421 Len=0 TSval=3633175680 TSecr=874477072
2413	4.876064828	172.31.8.66	172.31.13.81	TLSv1.2	117	Application Data
2414	4.876658424	172.31.13.81	172.31.8.66	TLSv1.2	169	Application Data
2415	4.876911972	172.31.8.66	172.31.13.81	TCP	66	50168 - 8443 [ACK] Seq=5910 Ack=1826444 Win=5421 Len=0 TSval=3633175685 TSecr=874477130
2416	4.877013916	172.31.13.81	172.31.8.66	TCP	9015	8443 - 50168 [ACK] Seq=1826444 Ack=5910 Win=472 Len=8949 TSval=874477130 TSecr=3633175685 [TCP s...
2417	4.877017551	172.31.13.81	172.31.8.66	TLSv1.2	547	Application Data
2418	4.877119263	172.31.13.81	172.31.8.66	TLSv1.2	129	Application Data
2419	4.877273346	172.31.8.66	172.31.13.81	TCP	66	50168 - 8443 [ACK] Seq=5910 Ack=1835874 Win=5353 Len=0 TSval=3633175685 TSecr=874477130
2420	4.877349636	172.31.8.66	172.31.13.81	TCP	66	50168 - 8443 [ACK] Seq=5910 Ack=1835937 Win=5421 Len=0 TSval=3633175685 TSecr=874477130
2421	4.880150892	172.31.8.66	172.31.13.81	TLSv1.2	117	Application Data
2422	4.880150892	172.31.13.81	172.31.8.66	TLSv1.2	129	Application Data
2423	4.880222055	172.31.8.66	172.31.13.81	TLSv1.2	3255	Application Data
2424	4.880340448	172.31.13.81	172.31.8.66	TLSv1.2	129	Application Data
2425	4.880387411	172.31.13.81	172.31.8.66	TLSv1.2	2567	Application Data
2426	4.880549352	172.31.8.66	172.31.13.81	TCP	66	50168 - 8443 [ACK] Seq=5961 Ack=1839189 Win=5421 Len=0 TSval=3633175688 TSecr=874477143
2427	4.88060702	172.31.8.66	172.31.13.81	TCP	66	50168 - 8443 [ACK] Seq=5961 Ack=1841753 Win=5421 Len=0 TSval=3633175688 TSecr=874477143
2428	4.880764370	172.31.13.81	172.31.8.66	TLSv1.2	129	Application Data
2429	4.880818154	172.31.13.81	172.31.8.66	TLSv1.2	1119	Application Data
2430	4.880998797	172.31.13.81	172.31.8.66	TLSv1.2	129	Application Data
2431	4.891042975	172.31.8.66	172.31.13.81	TCP	66	50168 - 8443 [ACK] Seq=5961 Ack=1842860 Win=5421 Len=0 TSval=3633175688 TSecr=874477144
2432	4.891062025	172.31.13.81	172.31.8.66	TCP	9015	8443 - 50168 [ACK] Seq=1842923 Ack=5961 Win=472 Len=8949 TSval=874477144 TSecr=3633175688 [TCP s...
2433	4.891060818	172.31.13.81	172.31.8.66	TLSv1.2	388	Application Data
2434	4.891207671	172.31.13.81	172.31.8.66	TLSv1.2	129	Application Data
2435	4.891318332	172.31.8.66	172.31.13.81	TCP	66	50168 - 8443 [ACK] Seq=5961 Ack=1852192 Win=5353 Len=0 TSval=3633175688 TSecr=874477144
2436	4.931563169	172.31.8.66	172.31.13.81	TCP	66	50168 - 8443 [ACK] Seq=5961 Ack=1852255 Win=5421 Len=0 TSval=3633175699 TSecr=874477144

### Analyze Packets:

- Select a captured packet to view detailed information in the packet list and packet details panes.

9 3.379219	10.40.1.115	172.237.7.131	TCP	55	58093 - 443 [ACK] Seq=1 Ack=1 Win=509 Len=1 [TCP segment of a reassembled PDU]
10 3.401956	172.237.7.131	10.40.1.115	TCP	66	443 - 58093 [ACK] Seq=1 Ack=2 Win=246 Len=0 SLE=1 SRE=2
11 6.539753	10.40.1.115	10.40.0.1	DNS	74	Standard query 0xa005 A www.google.com
12 6.542786	10.40.0.1	10.40.1.115	DNS	98	Standard query response 0xa005 A www.google.com A 172.217.5.228
Frame 12: 90 bytes on wire (720 bits), 90 bytes captured (720 bits) on interface Intel PRO/100 MT Desktop, Src: Google [172.217.5.228], Dst: Kali Linux [10.40.1.115]					
Ethernet II, Src: Intel PRO/100 MT Desktop [172.217.5.228], Dst: Kali Linux [10.40.1.115]					
Internet Protocol Version 4, Src: 10.40.1.115, Dst: 10.40.0.1					
User Datagram Protocol, Src Port: 53, Dst Port: 64136					
Source Port: 53 Destination Port: 64136 Length: 56 Checksum: 0xcx76 [unverified] [Checksum Status: Unverified] [Stream index: 0] [Timestamps] UDP payload (48 bytes) Domain Name System (response) Transaction ID: 0xa005 Flags: 0x8180 Standard query response, No error					
0000 08 71 9d f4 1c 5e 0c 64 e6 88 08 00 45 00 q 4 ^ d - E 0010 00 45 c9 00 00 40 11 df 14 0a 28 00 01 0a 28 L - @ . ( - ( . 0020 01 73 00 35 fa 88 09 38 cc 7c a9 09 81 80 00 01 s - b - V 0030 00 01 00 00 00 00 03 77 77 77 06 67 0f 6f 67 6c ..... w ww googl 0040 65 03 63 6f 6d 00 00 01 00 01 c0 0c 00 01 00 01 e com ..... 0050 00 00 00 87 00 04 ac d9 05 e4 ..... .....					

### 3. Command-Line Interface (CLI):

#### Capture Packets:

```
dumpcap -i <interface> -w <output_file>
```

- Example:

```
dumpcap -i eth0 -w capture.pcap
```

Read Captured Files:

- Read a captured file using the tshark command:

```
tshark -r <input_file>
```

- Example:

```
tshark -r capture.pcap
```

Filtering Packets:

- Use display filters to filter packets based on specific criteria:

```
tshark -r <input_file> -Y <display_filter>
```

- Example:

```
tshark -r capture.pcap -Y "ip.src == 192.168.1.1"
```

Statistics:

- Generate statistics on captured packets:

```
tshark -r <input_file> -z <statistics_type>
```

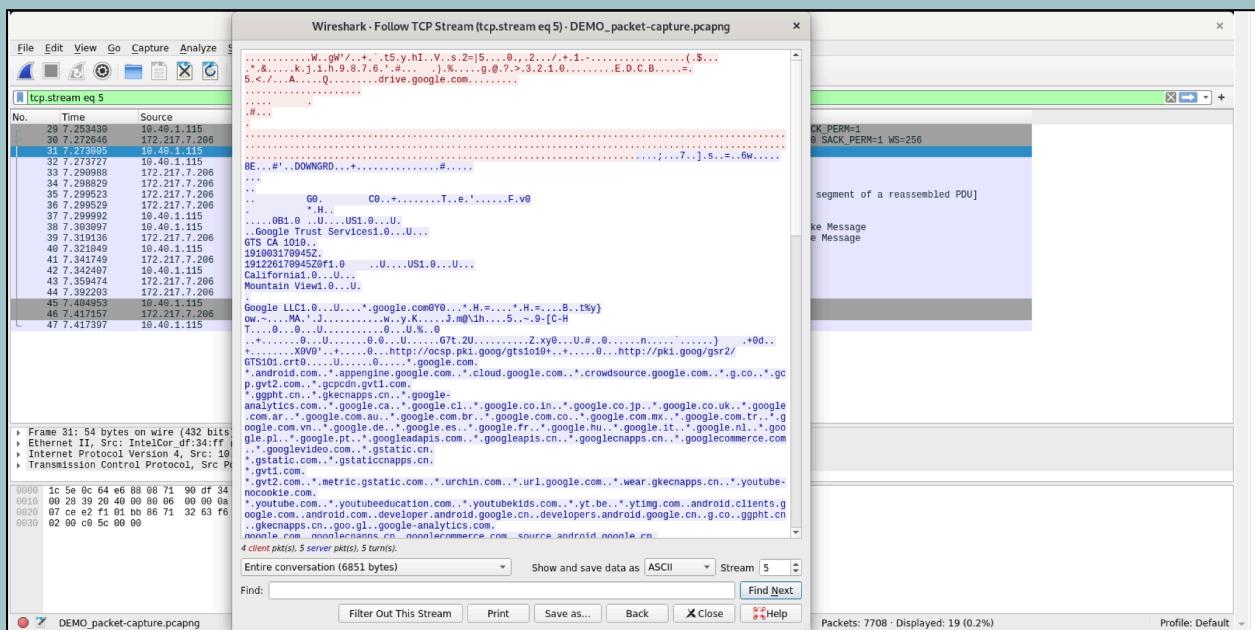
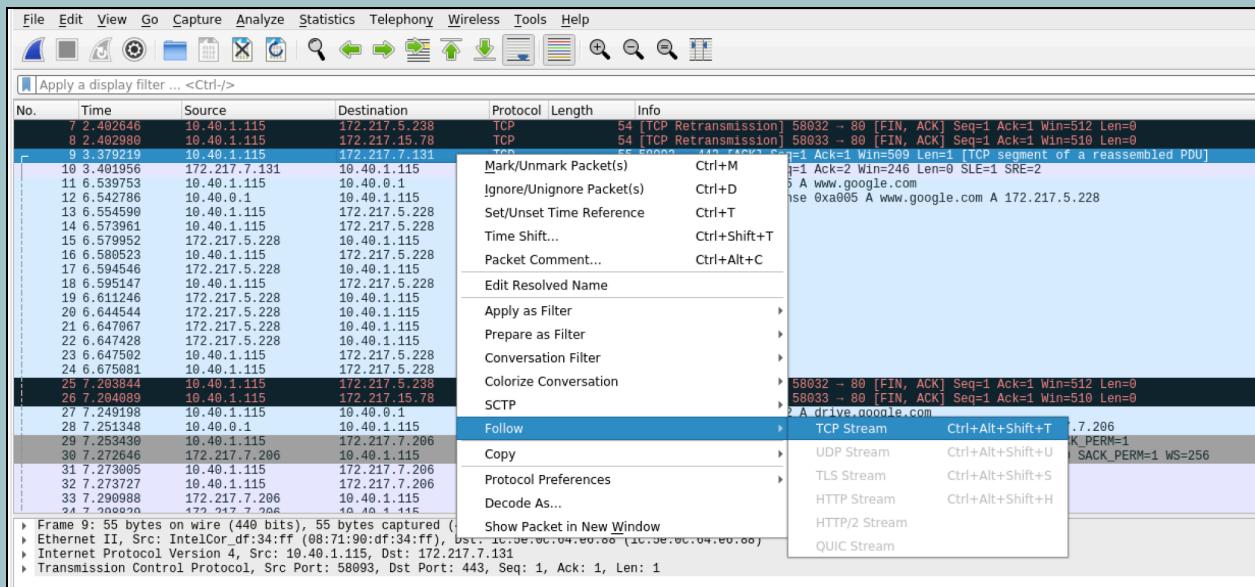
- Example:

```
tshark -r capture.pcap -z io,phs
```

#### 4. Advanced Features:

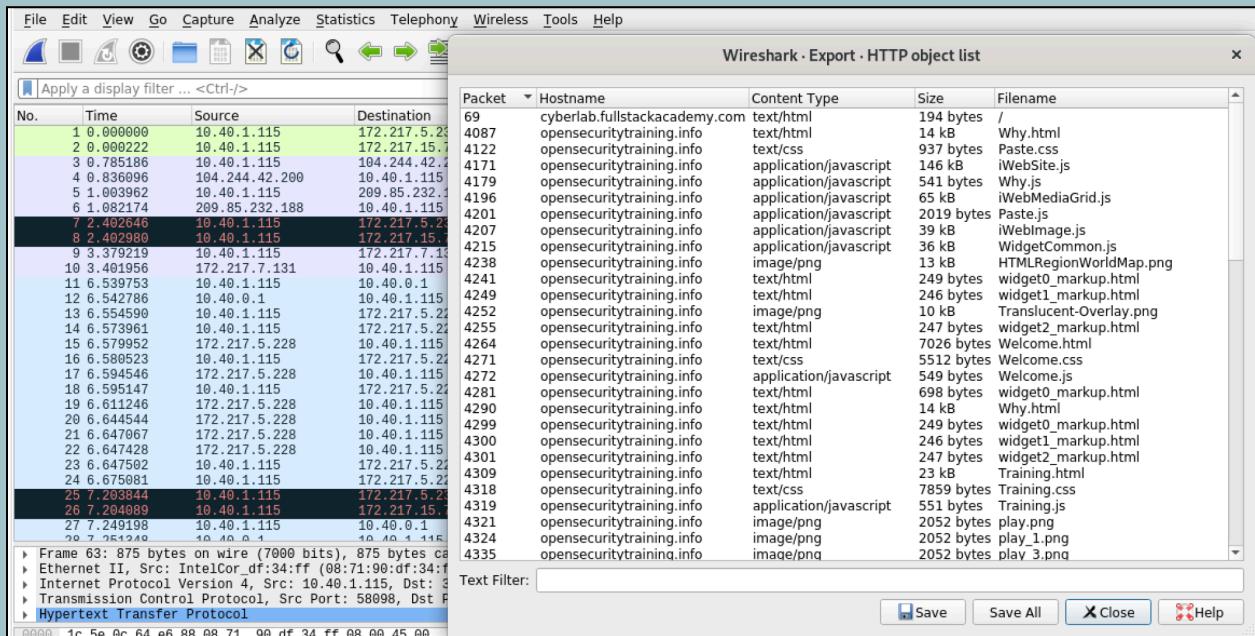
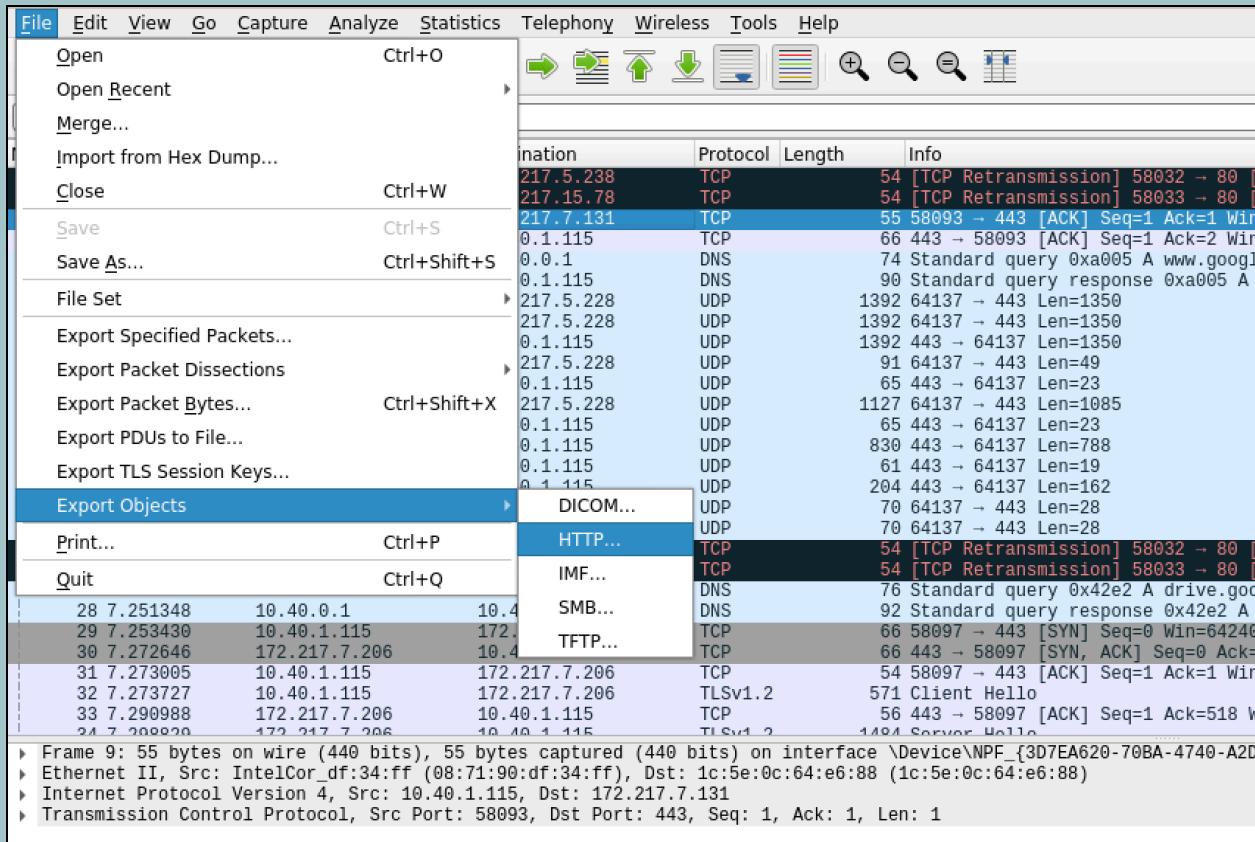
Follow TCP Stream:

- Follow a TCP stream to view the entire conversation:
  - Right-click on a TCP packet and select "Follow > TCP Stream."

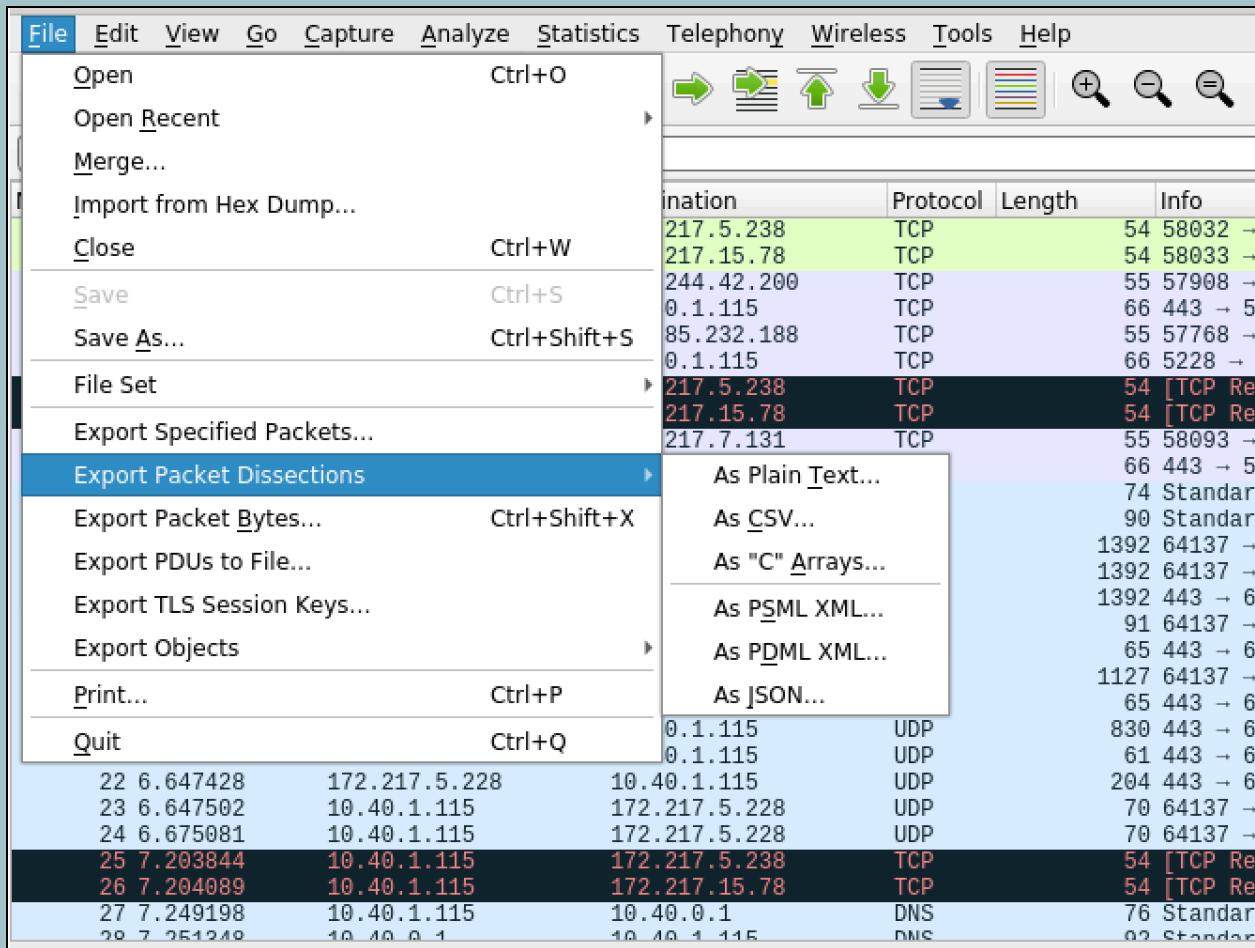


## Export Packets:

- Export packets to various formats:
  - Choose "File > Export Objects" to export objects like HTTP files.



- Choose "File > Export Packet Dissections" to export packet details.



#### Command-Line Capture Filters:

- Use capture filters to specify which packets to capture:

```
dumpcap -i <interface> -w <output_file> <capture_filter>
```

- Example:

```
dumpcap -i eth0 -w capture.pcap host 192.168.1.1
```

## 5. Tips and Best Practices:

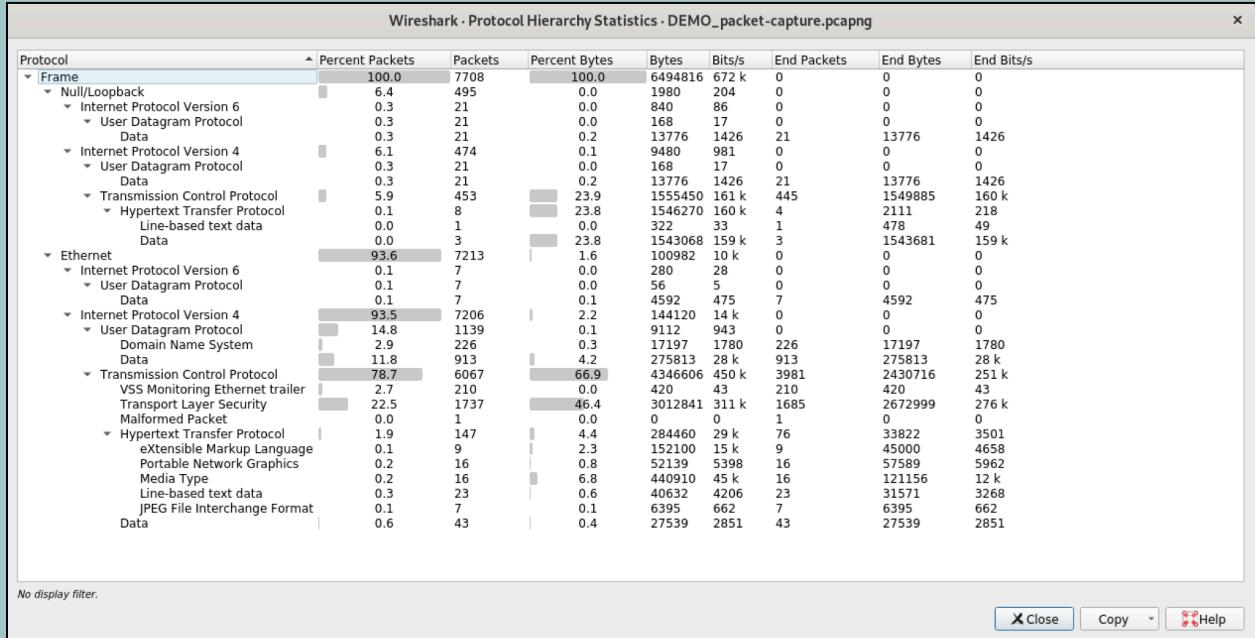
### Use Display Filters Wisely:

- Utilize display filters to narrow down the packet list to relevant packets.

No.	Time	Source	Destination	Protocol	Length	Info
63	7.704933	19.49.1.115	3.218.245.26	HTTP	875	GET / HTTP/1.1
4079	38.719988	19.49.1.115	59.62.109.1	HTTP	497	GET /why.html HTTP/1.1
4111	38.892931	19.49.1.115	59.62.109.1	HTTP	395	GET /Scripts/Widgets/HTMLRegion/Paste.css HTTP/1.1
4112	38.892503	19.49.1.115	59.62.109.1	HTTP	415	GET /Scripts/Widgets/HTMLRegion/Paste.js HTTP/1.1
4125	38.963214	19.49.1.115	59.62.109.1	HTTP	416	GET /Scripts/Widgets/HTMLRegion/WidgetImage.js HTTP/1.1
4126	38.963316	19.49.1.115	59.62.109.1	HTTP	420	GET /Scripts/Widgets/HTMLRegion/WidgetMediaGrid.js HTTP/1.1
4129	38.964440	19.49.1.115	59.62.109.1	HTTP	443	GET /Scripts/Widgets/SharedResources/WidgetCommon.js HTTP/1.1
4132	38.965576	19.49.1.115	59.62.109.1	HTTP	431	GET /Scripts/Widgets/HTMLRegion/Paste.js HTTP/1.1
4135	38.967556	19.49.1.115	59.62.109.1	HTTP	412	GET /Why_files/Why.js HTTP/1.1
4221	31.318923	19.49.1.115	59.62.109.1	HTTP	481	GET /Scripts/Widgets/HTMLRegion/HTMLRegionWorldMap.png HTTP/1.1
4222	31.311417	19.49.1.115	59.62.109.1	HTTP	571	GET //Why_files/widget0_markup.html HTTP/1.1
4223	31.311454	19.49.1.115	59.62.109.1	HTTP	487	GET /Scripts/Widgets/SharedResources/Translucent-Overlay.png HTTP/1.1
4225	31.314499	19.49.1.115	59.62.109.1	HTTP	571	GET //Why_files/widget1_markup.html HTTP/1.1
4226	31.316634	19.49.1.115	59.62.109.1	HTTP	571	GET //Why_files/widget2_markup.html HTTP/1.1
4260	32.600389	19.49.1.115	59.62.109.1	HTTP	553	GET /Welcome.html HTTP/1.1
4269	32.842869	19.49.1.115	59.62.109.1	HTTP	440	GET /Welcome_files/Welcome.css HTTP/1.1
4270	32.851547	19.49.1.115	59.62.109.1	HTTP	424	GET /Welcome_files/Welcome.js HTTP/1.1
4275	33.138672	19.49.1.115	59.62.109.1	HTTP	579	GET //Welcome_files/widget0_markup.html HTTP/1.1
4285	33.896812	19.49.1.115	59.62.109.1	HTTP	553	GET /Why.html HTTP/1.1
4296	34.174961	19.49.1.115	59.62.109.1	HTTP	571	GET //Why_files/widget0_markup.html HTTP/1.1
4297	34.178149	19.49.1.115	59.62.109.1	HTTP	571	GET //Why_files/widget1_markup.html HTTP/1.1
4298	34.180036	19.49.1.115	59.62.109.1	HTTP	571	GET //Why_files/widget2_markup.html HTTP/1.1
4305	34.889394	19.49.1.115	59.62.109.1	HTTP	554	GET /Training.html HTTP/1.1
4313	35.021917	19.49.1.115	59.62.109.1	HTTP	443	GET /Training_files/Training.css HTTP/1.1
4314	35.024881	19.49.1.115	59.62.109.1	HTTP	427	GET /Training_files/Training.js HTTP/1.1
4315	35.027792	19.49.1.115	59.62.109.1	HTTP	460	GET /Training_files/play.png HTTP/1.1
4316	35.028872	19.49.1.115	59.62.109.1	HTTP	462	GET /Training_files/play_1.png HTTP/1.1
4319	35.100704	19.49.1.115	59.62.109.1	HTTP	460	GET /Training_files/play_2.png HTTP/1.1

### Analyze Protocol Hierarchy:

- Check the protocol hierarchy statistics to understand the distribution of protocols in the capture.



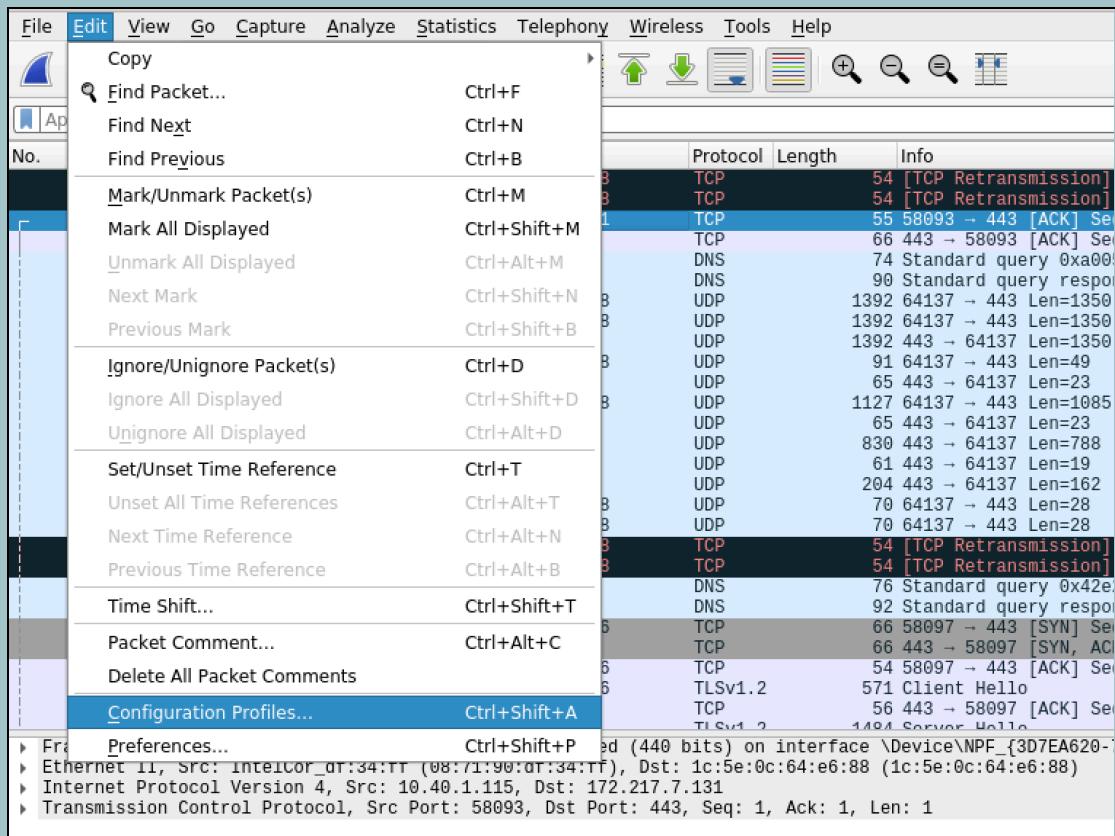
### Document Findings:

- Keep detailed notes on packet captures for future reference and analysis.

## 6. Profiles and Configuration:

### Profiles:

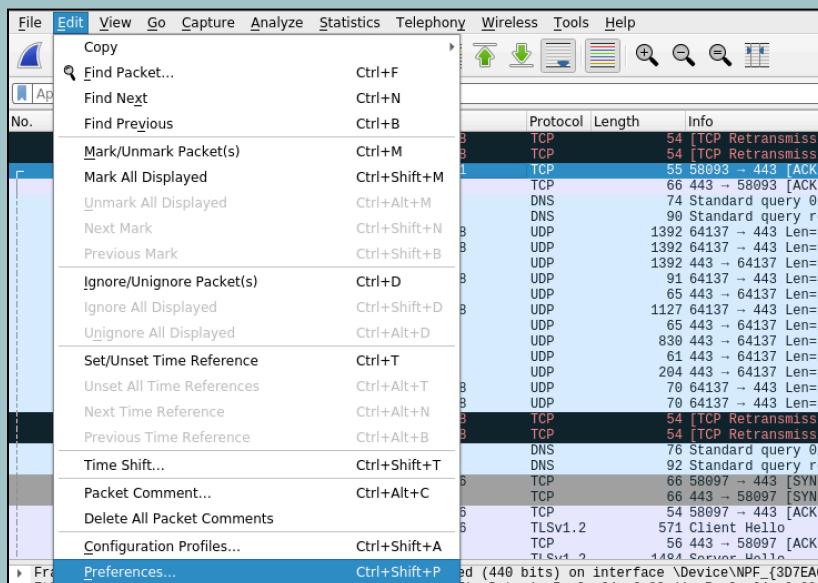
- Create and manage profiles for different capture scenarios:
  - Use "Edit > Configuration Profiles" to create, edit, and switch between profiles.



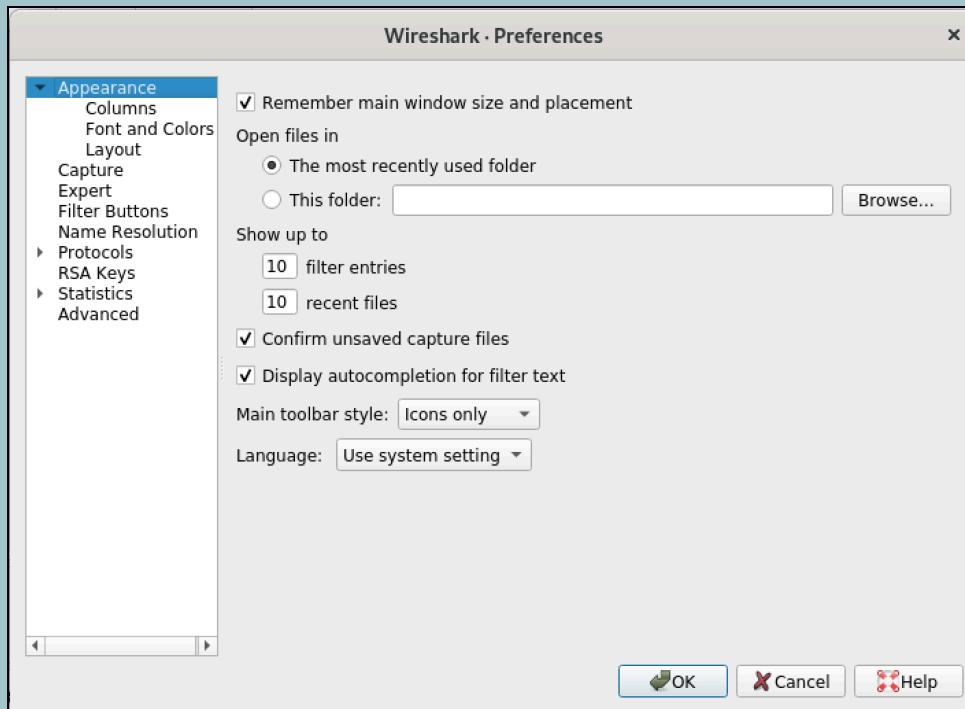
- Customize settings such as capture filters, display filters, and column preferences for each profile.

#### Configuration:

- Adjust Wireshark settings to suit your preferences and requirements:
  - Navigate to "Edit > Preferences" to configure options such as appearance, protocols, and capture settings.



- Customize colorization rules, protocol preferences, and interface settings.



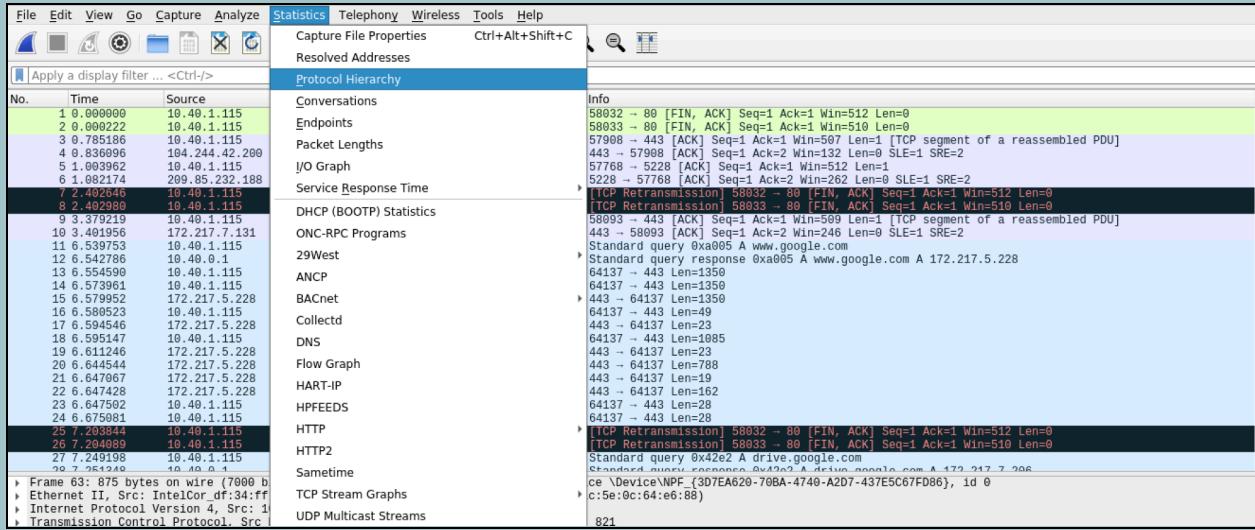
## 7. Expert Information and Analysis:

### Expert Information:

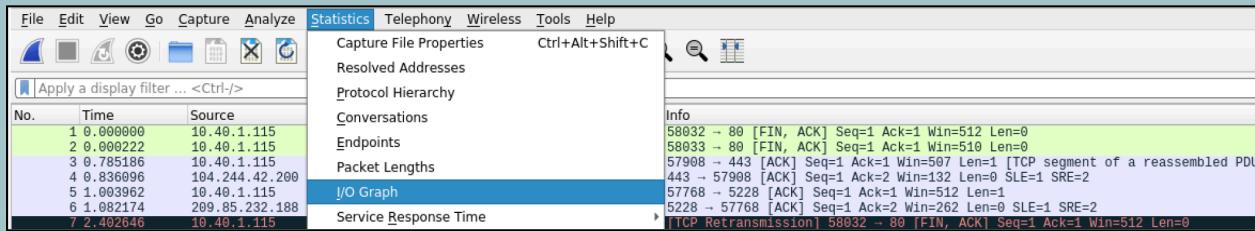
- Use Wireshark's expert system to identify potential issues and anomalies in the capture:
  - The expert information column provides alerts and warnings about packet characteristics.
  - Investigate expert warnings to troubleshoot network problems and diagnose issues.

### Graphs and Charts:

- Visualize packet data using built-in graphs and charts:
  - Navigate to "Statistics > Protocol Hierarchy" to view a hierarchical breakdown of protocols.



- Use "Statistics > IO Graphs" to plot various statistics over time.



## 8. Remote Capture:

### Remote Packet Capture:

- Capture packets from remote interfaces using the dumpcap utility:

```
fstack@ip-172-31-13-81:~$ # dumpcap -i <interface> -w <output_file> -i <remote_interface>
-P -w - | wireshark -k -i -
```

- Example:

```
fstack@ip-172-31-13-81:~$ dumpcap -i eth0 -w capture.pcap -i ssh://user@remote_host/eth0
-P -w - | wireshark -k -i -
```

### SSH Remote Capture:

- Use SSH to securely capture packets from remote interfaces:
  - Ensure that SSH is configured on the remote host and accessible from the local machine.
  - Replace <remote\_host> with the hostname or IP address of the remote host.

## 9. Scripting and Automation:

### Lua Scripting:

- Extend Wireshark's functionality using Lua scripting:
  - Write Lua scripts to automate tasks, extract data, and analyze packets.
  - Use Wireshark's Lua API to access packet data and interact with the user interface.

### Automated Packet Processing:

- Integrate Wireshark with other tools and scripts for automated packet processing:
  - Use command-line tools like tshark to process captured packets in batch mode.
  - Develop custom scripts to analyze specific protocols or perform complex analysis tasks.

## **10. Collaboration and Sharing:**

- Packet Sharing:
  - Share packet captures with colleagues and collaborators:
    - Export captures to common file formats like PCAP or CSV for sharing and analysis.
    - Use file sharing platforms or collaboration tools to exchange captures and findings.
- Wireshark Profiles:
  - Share Wireshark profiles and configurations with team members:
    - Export and distribute configuration profiles to ensure consistency across team members.
    - Store profiles in version control systems for easy access and versioning.

## **11. Advanced Protocol Analysis:**

### Protocol Dissection:

- Analyze protocol details and dissect packets for in-depth inspection:
  - Wireshark provides detailed information about each protocol layer, including headers, fields, and payloads.
  - Use protocol preferences to customize protocol decoding and display options.

### Reassembly and Reconstruction:

- Reassemble fragmented packets and reconstruct higher-layer protocols:
  - Wireshark automatically reassembles fragmented packets and presents them as complete units for analysis.
  - Use follow stream functionality to reconstruct higher-layer protocols like HTTP, FTP, and SMTP.

## **12. Network Troubleshooting:**

### Identifying Performance Issues:

- Use Wireshark to diagnose and troubleshoot network performance problems:
  - Analyze packet capture to identify latency issues, packet loss, and network congestion.
  - Investigate TCP retransmissions, out-of-order packets, and other anomalies affecting network performance.

### Detecting Security Threats:

- Use Wireshark for security analysis and threat detection:
  - Identify malicious traffic, suspicious behavior, and unauthorized access attempts.

- Analyze packet payloads, protocol anomalies, and network patterns to detect security threats.

### **13. Continuous Monitoring and Analysis:**

Packet Capture Automation:

- Automate packet capture and analysis for continuous monitoring:
  - Use scheduled tasks or cron jobs to periodically capture packets and analyze network traffic.
  - Implement alerts and notifications based on predefined criteria to monitor network health and security.

Log File Integration:

- Integrate Wireshark with log management systems for centralized analysis:
  - Export packet capture data to log files compatible with SIEM (Security Information and Event Management) solutions.
  - Analyze network events alongside system logs for comprehensive security monitoring.

### **14. Training and Education:**

Training Resources:

- Utilize Wireshark training materials and resources for skill development:
  - Access online tutorials, documentation, and video courses to learn Wireshark fundamentals and advanced techniques.
  - Participate in Wireshark community forums and user groups for knowledge sharing and support.

Certification:

- Obtain Wireshark certification for formal recognition of expertise:
  - Prepare for Wireshark Certified Network Analyst (WCNA) certification exams to validate proficiency in network analysis and troubleshooting.
  - Attend training courses and study relevant materials to prepare for certification exams.

In conclusion, Wireshark stands as a comprehensive and powerful network protocol analyzer, offering a wide array of features and functionalities tailored for network administrators, security professionals, and analysts alike. Throughout this runbook, we've explored the depth and breadth of Wireshark's capabilities, from its basic usage to advanced protocol analysis, troubleshooting, and beyond.

Wireshark enables users to:

- *Capture and Analyze Packets:* Wireshark provides a robust platform for capturing and analyzing network packets, allowing users to gain insights into network traffic, troubleshoot issues, and identify potential security threats.

- *Customize and Extend Functionality:* With its extensive customization options, including profiles, configuration settings, and scripting capabilities, Wireshark can be tailored to suit specific use cases and requirements. Users can create custom profiles, apply display filters, and develop Lua scripts to automate tasks and enhance analysis.
- *Collaborate and Share Findings:* Wireshark facilitates collaboration and knowledge sharing among team members by allowing users to share packet captures, profiles, and configurations. Whether it's exporting packet captures to common file formats or distributing configuration profiles, Wireshark supports seamless collaboration and communication.
- *Continuous Monitoring and Education:* Wireshark serves as a valuable tool for continuous network monitoring and analysis, enabling users to automate packet capture, analyze network traffic trends, and detect anomalies. Additionally, Wireshark offers a wealth of training resources and educational materials, including tutorials, documentation, and certification programs, to support ongoing skill development and proficiency in network analysis.

Overall, Wireshark empowers users to gain deep insights into network behavior, troubleshoot issues efficiently, enhance network security, and foster continuous improvement in network performance. With its user-friendly interface, powerful features, and active community support, Wireshark remains an indispensable tool for network professionals seeking to understand, optimize, and secure their networks effectively.