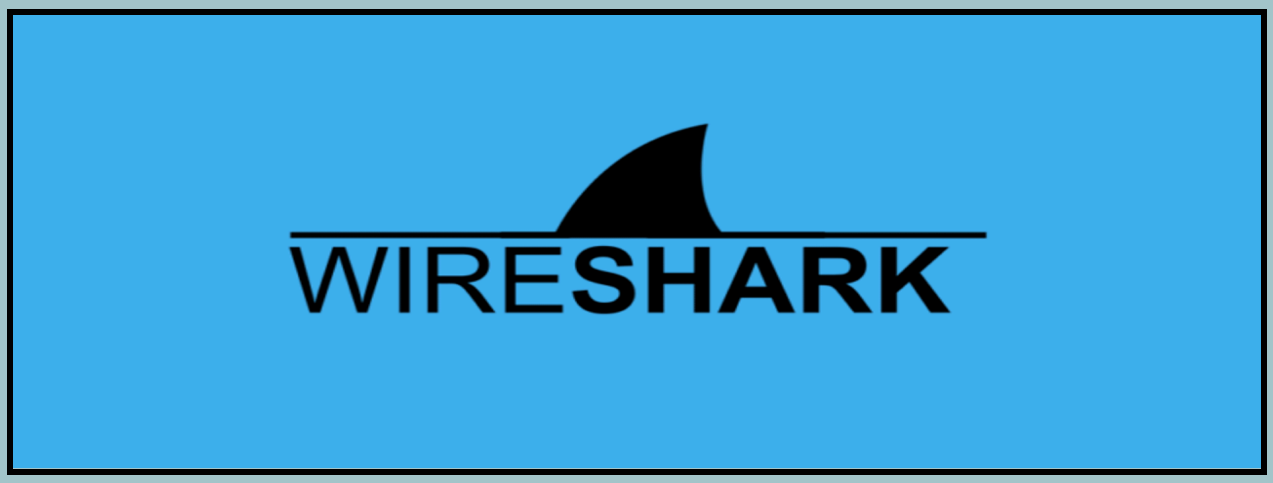



Laura Shupe

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>
```

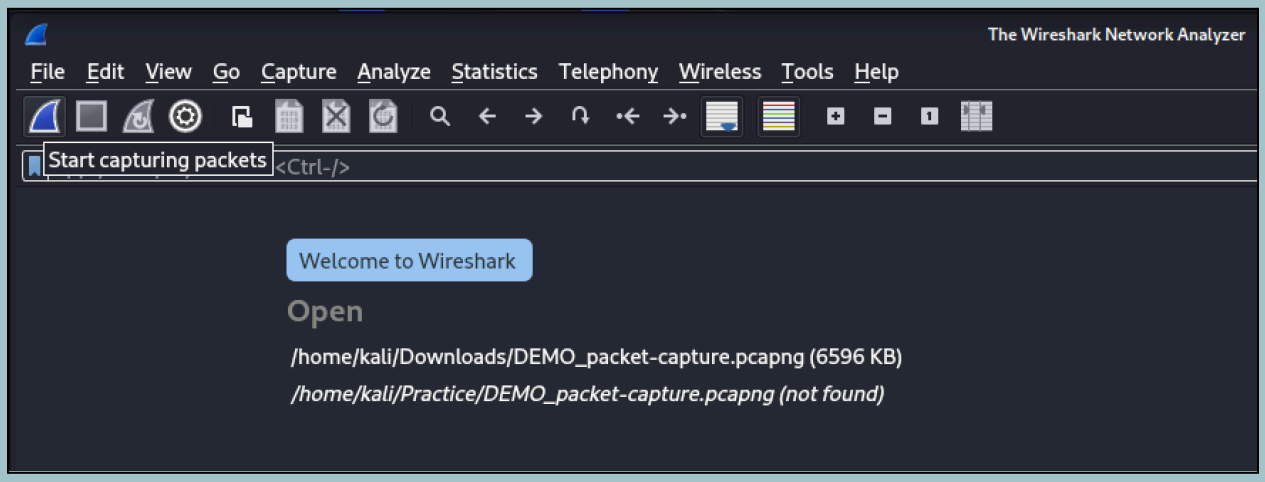
   macOS:
   - Download the installer from the official website and follow the installation instructions.

2. **Basic Usage:**

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

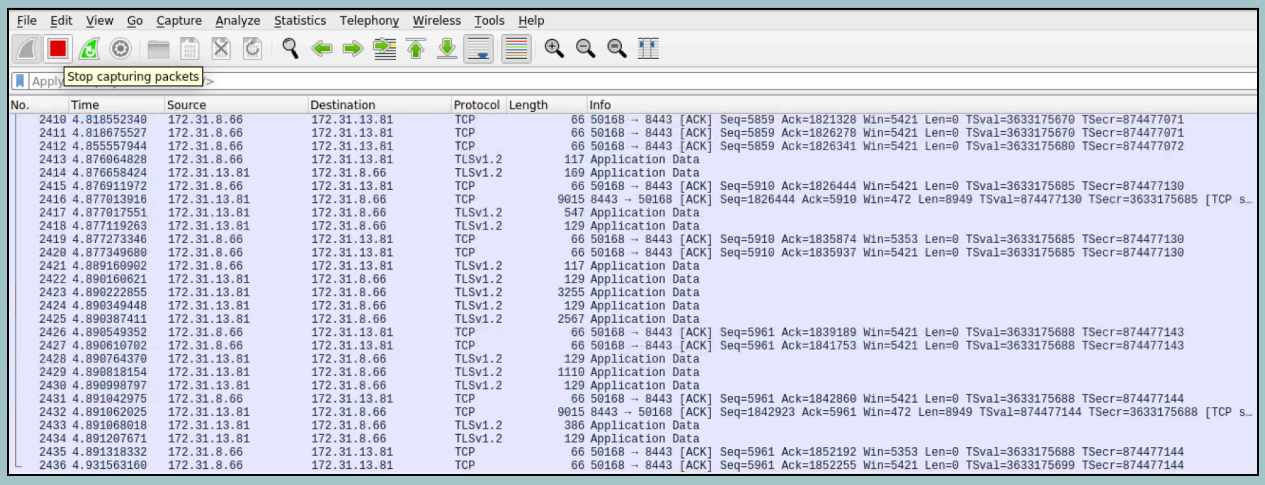
   Capture Traffic:

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



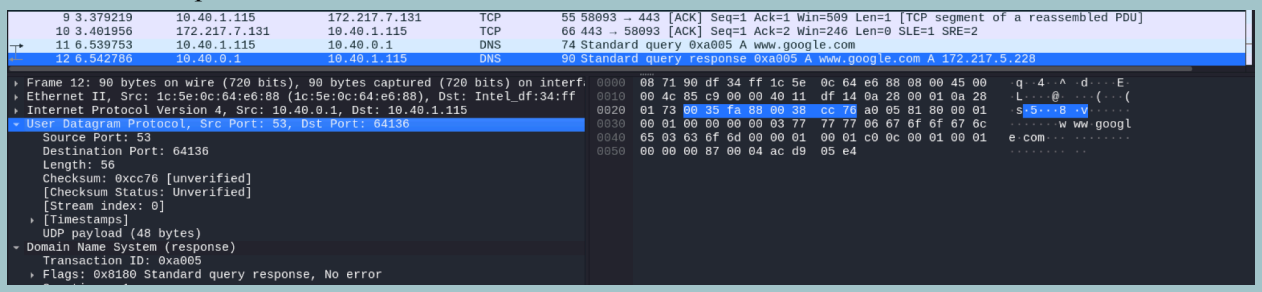

Stop Capture:

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




Analyze Packets:

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




### 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>
```

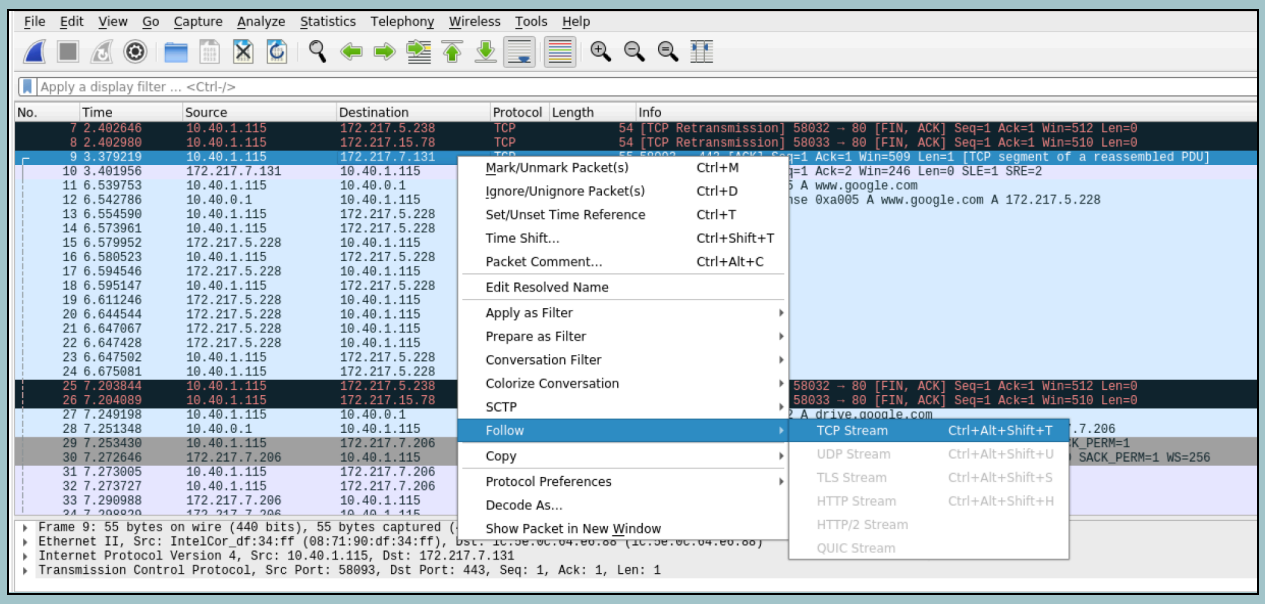
- Example:

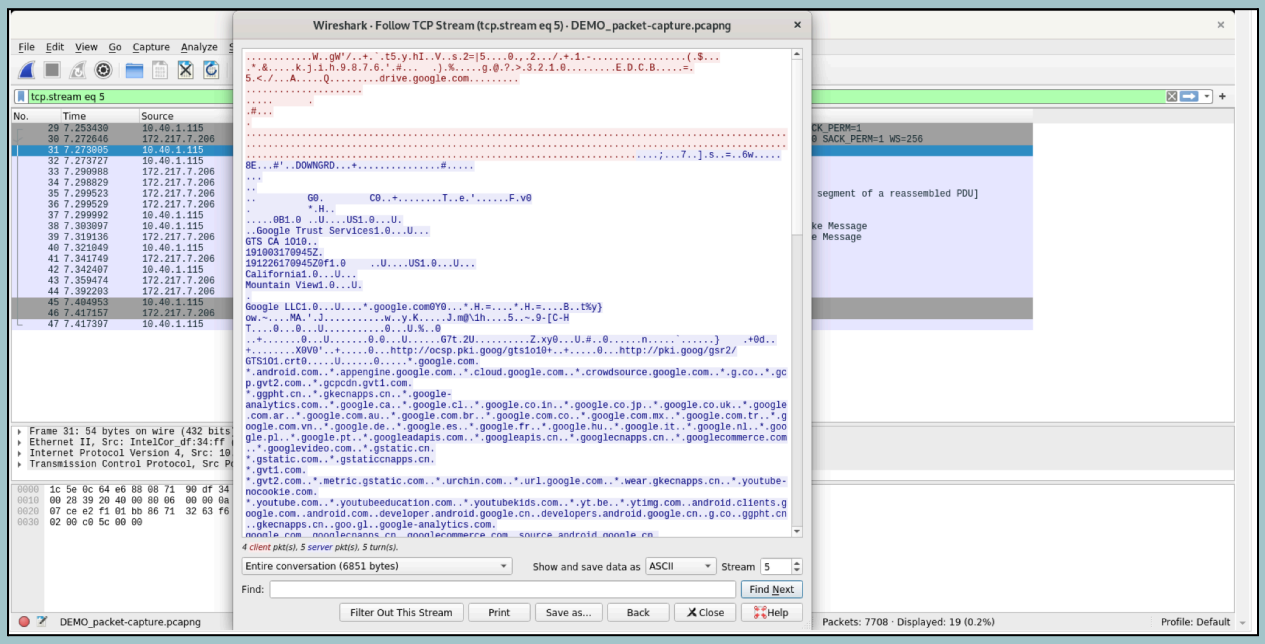
```
tshark -r capture.pcap -z io,phs
```

4. **Advanced Features:**

Follow TCP Stream:

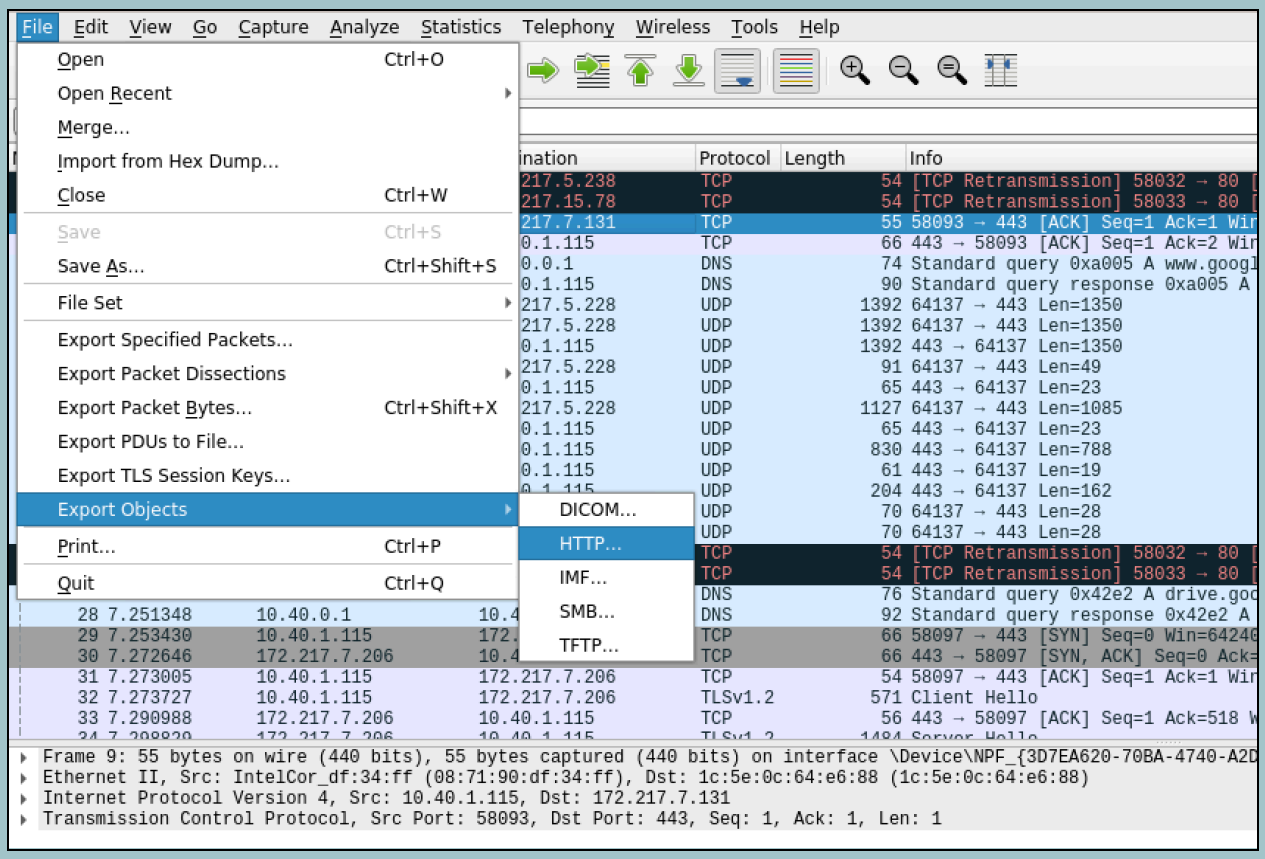
- Follow a TCP stream to view the entire conversation:
    - Right-click on a TCP packet and select "Follow > TCP Stream."

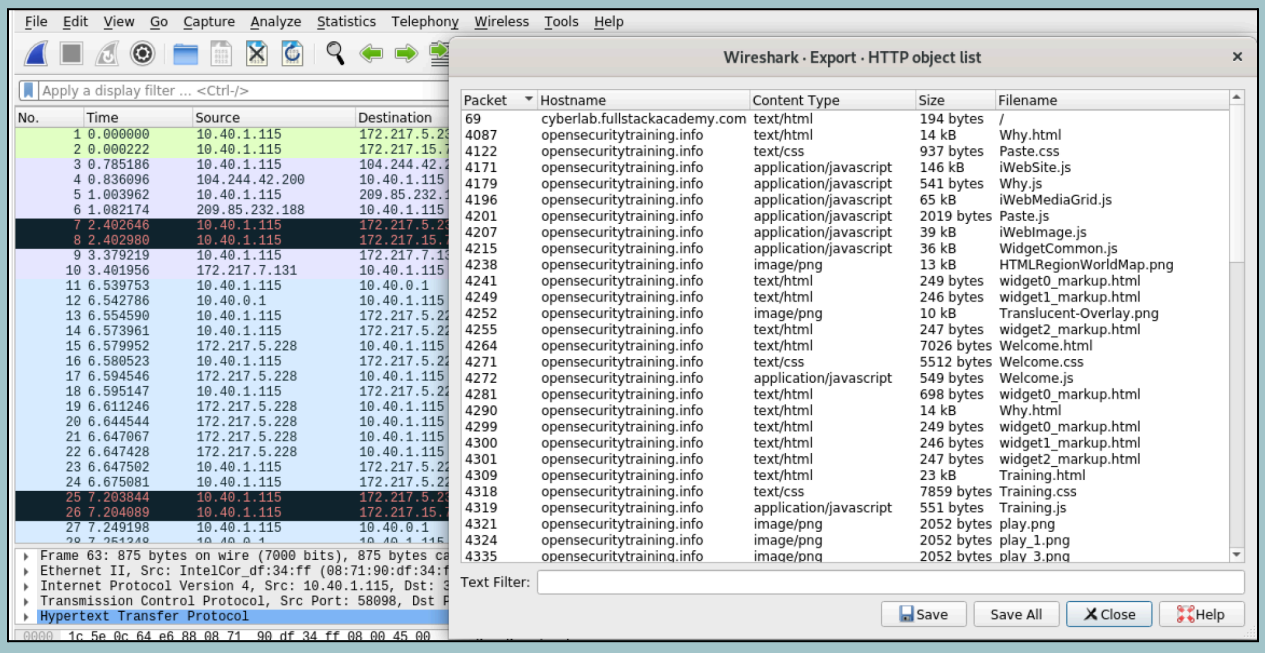


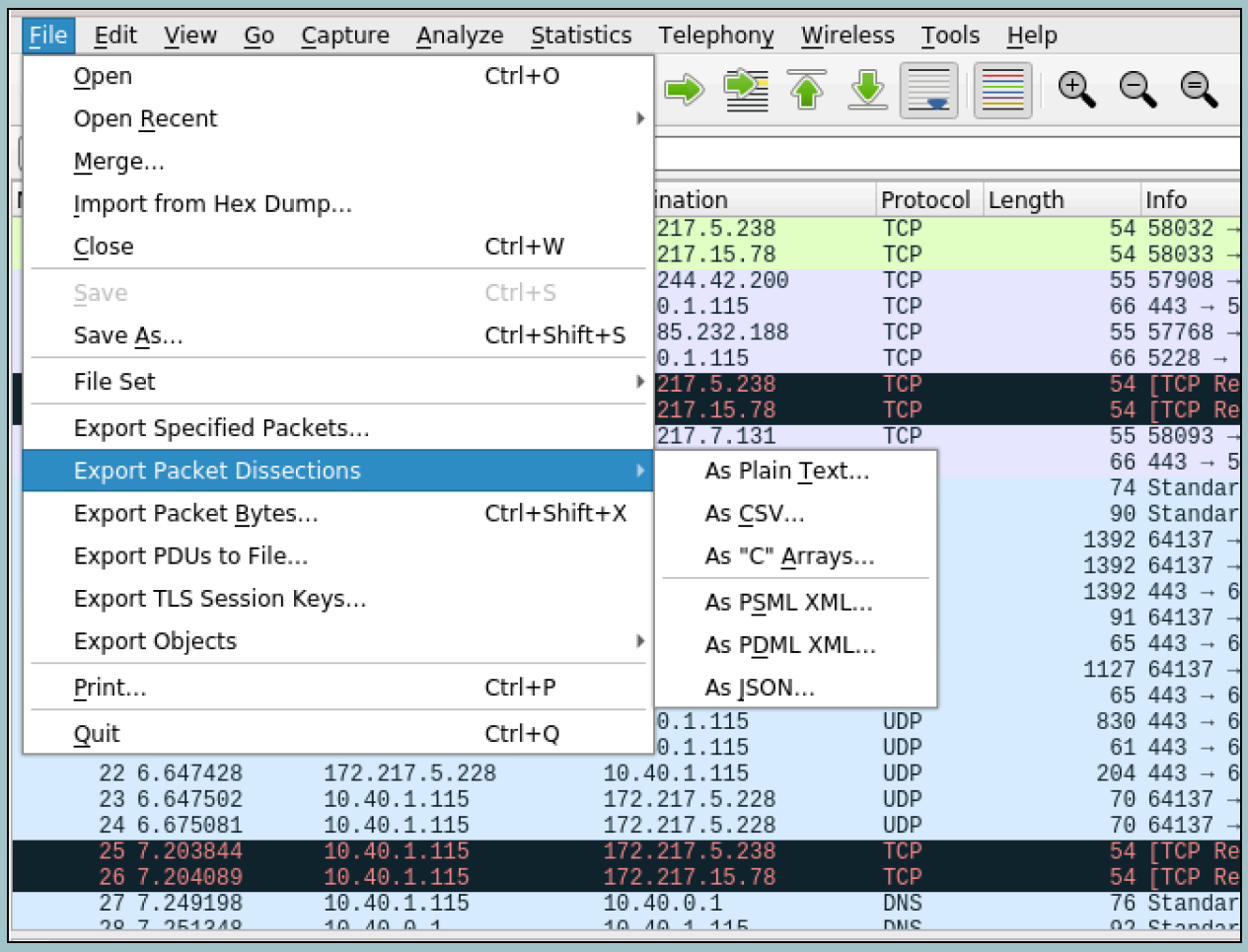

Export Packets:

- 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>
```

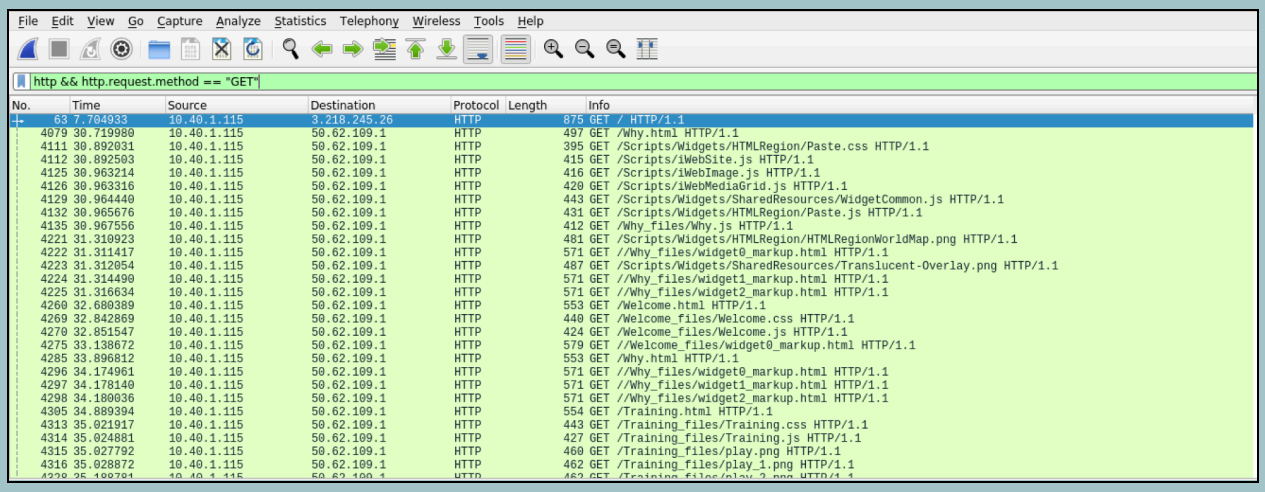
- Example:

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

5. **Tips and Best Practices:**

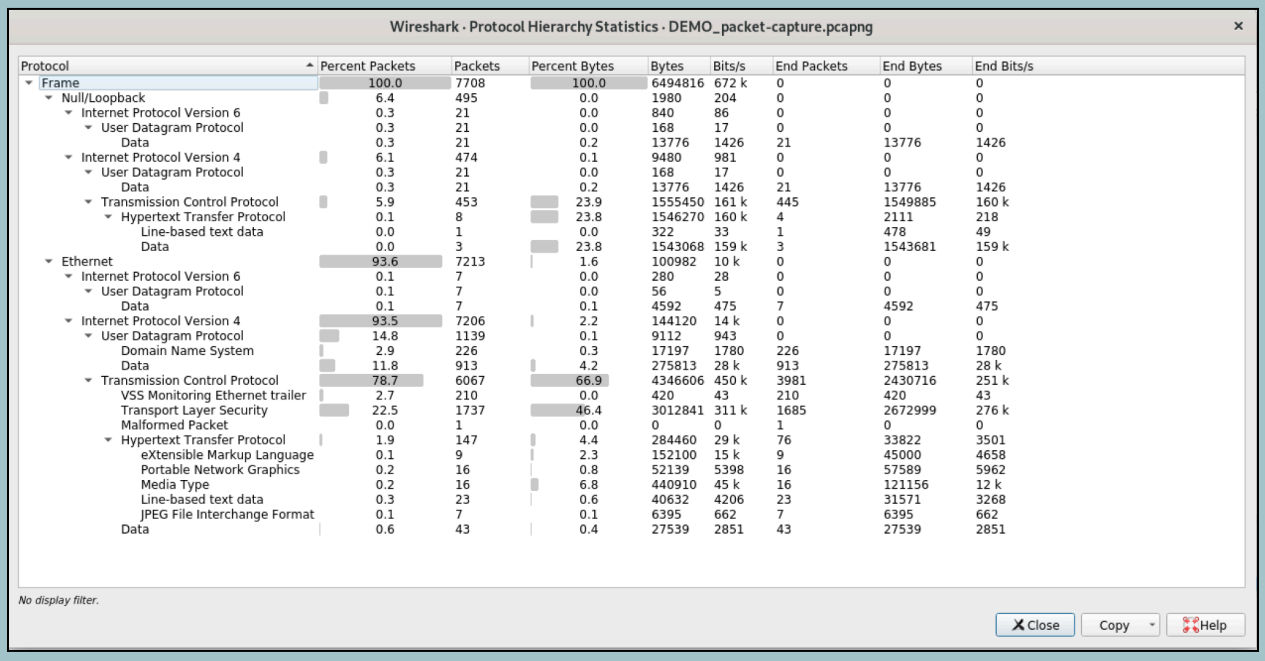
Use Display Filters Wisely:

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




Analyze Protocol Hierarchy:

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



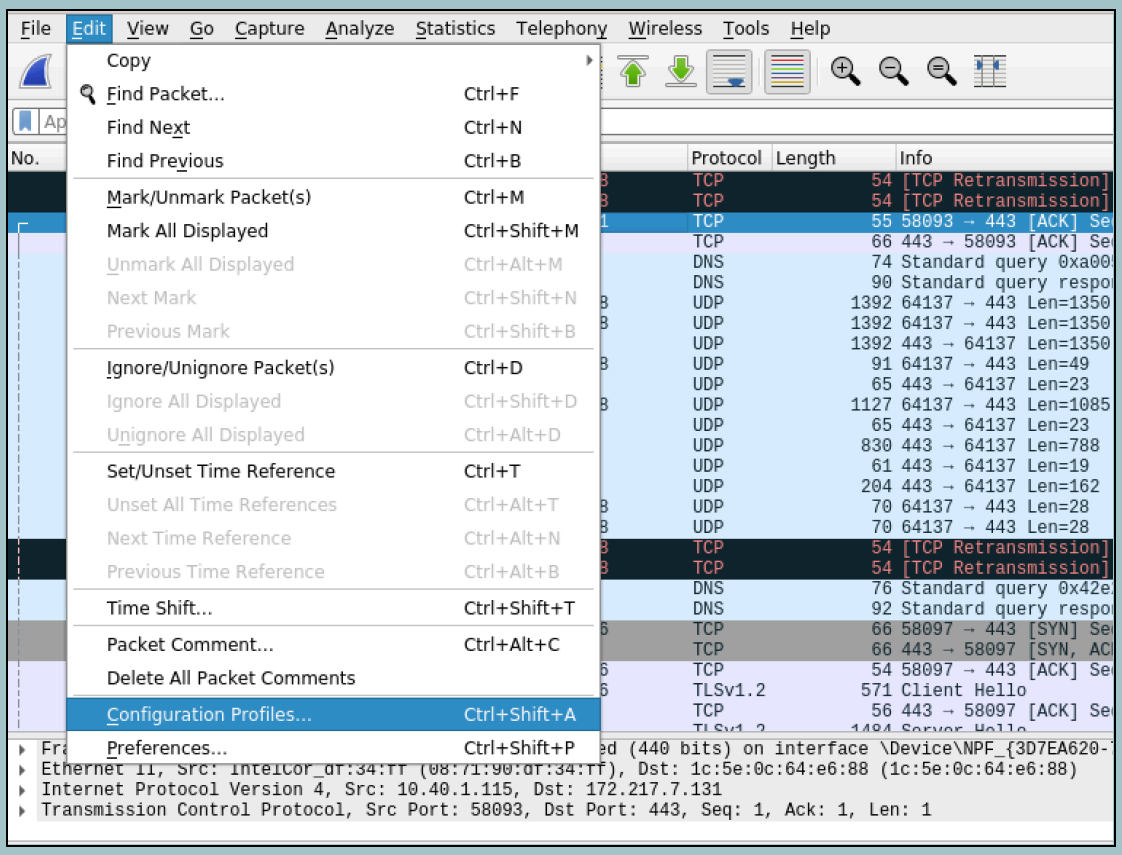

Document Findings:

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

6. **Profiles and Configuration:**

Profiles:

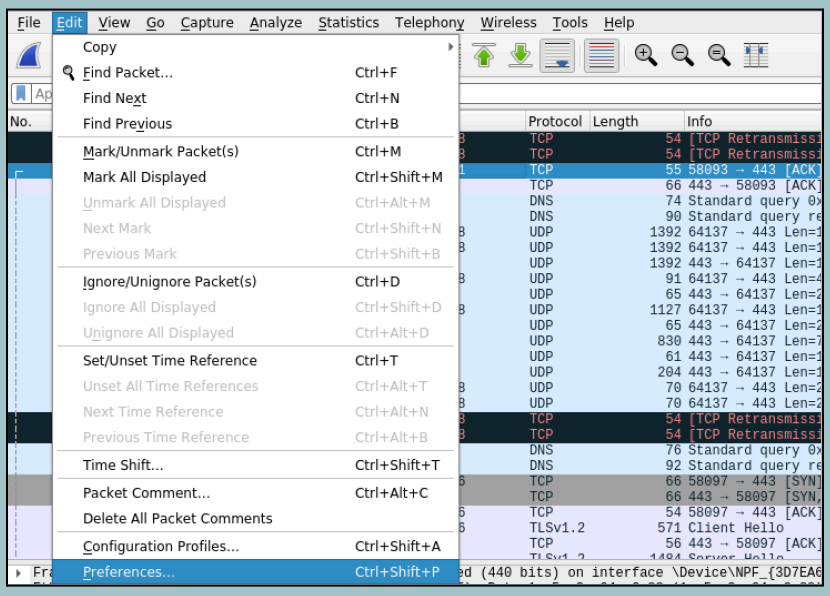
- Create and manage profiles for different capture scenarios:
  - Use "Edit > Configuration Profiles" to create, edit, and switch between profiles.



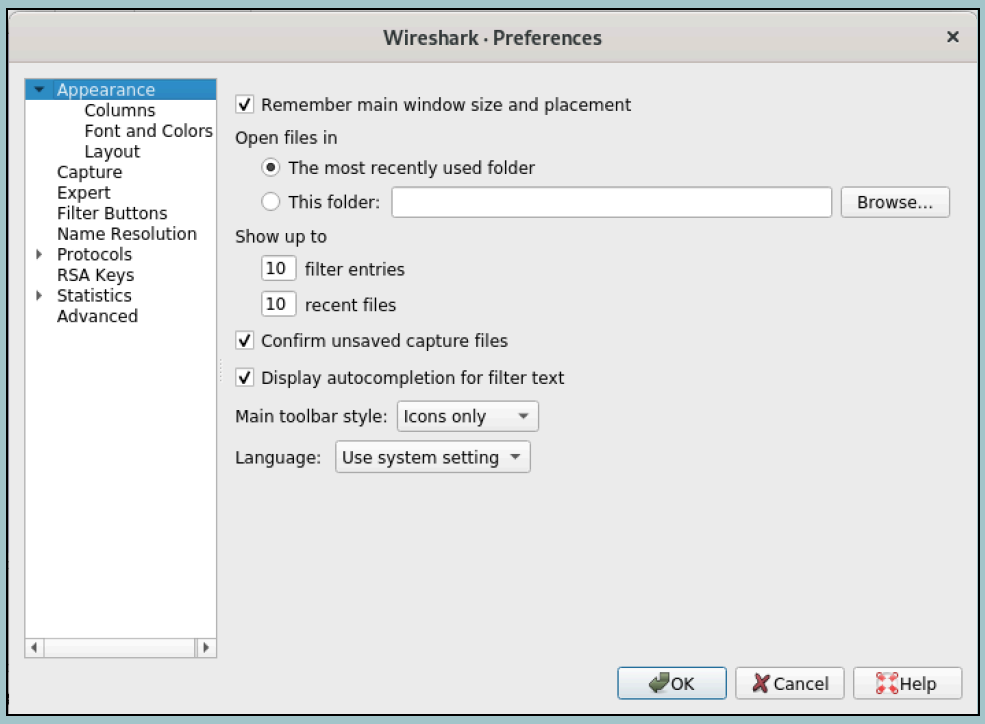

- Customize settings such as capture filters, display filters, and column preferences for each profile.

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.



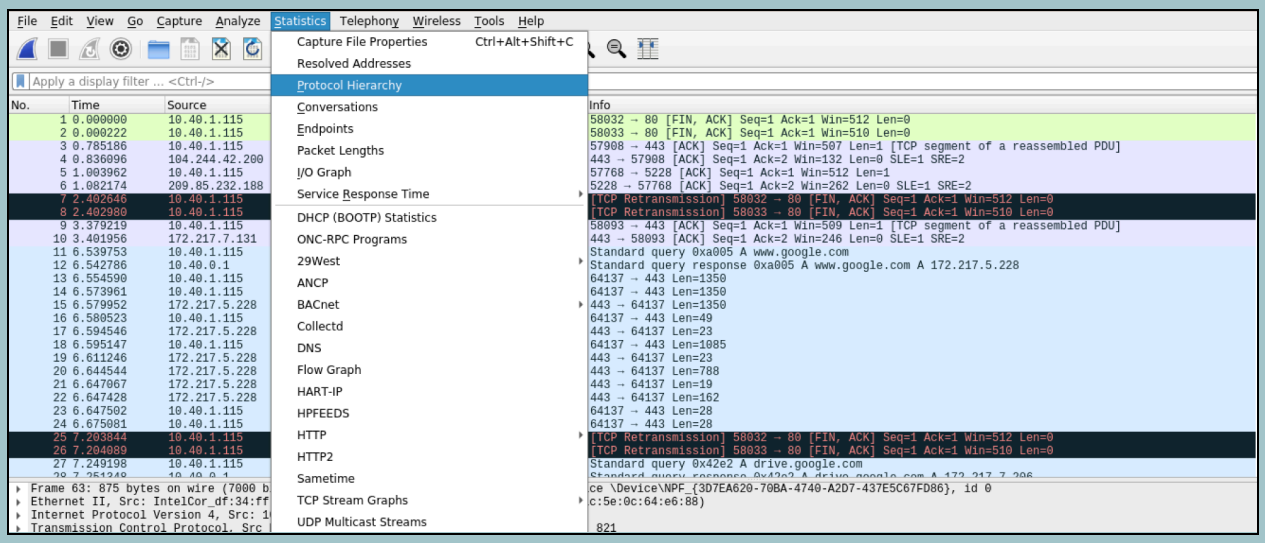

## 7. Expert Information and Analysis:

Expert Information:

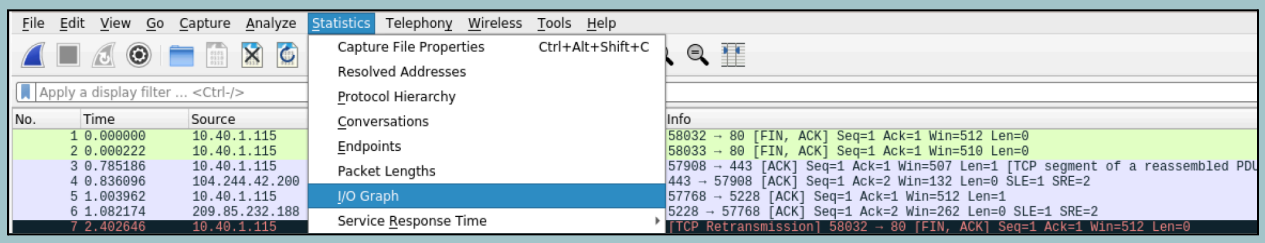
- Use Wireshark's expert system to identify potential issues and anomalies in the capture:
  - The expert information column provides alerts and warnings about packet characteristics.
  - Investigate expert warnings to troubleshoot network problems and diagnose issues.

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.