Report on Analyzing a Secure HTTPS Web Application Session using Wireshark

Introduction

The objective of this report is to detail the process of analyzing a secure HTTPS web application session using Wireshark. We will follow a step-by-step approach to capture network traffic, filter relevant data, and examine key aspects of the session, such as the TLS version used and the Client and Server Key Exchange Mechanism.

Methodology

1. Clear the Cache in Firefox Browser

Before initiating the packet capture, it is essential to clear the browser cache to ensure that no cached data interferes with the session analysis.

2. Start a Packet Capture in Wireshark

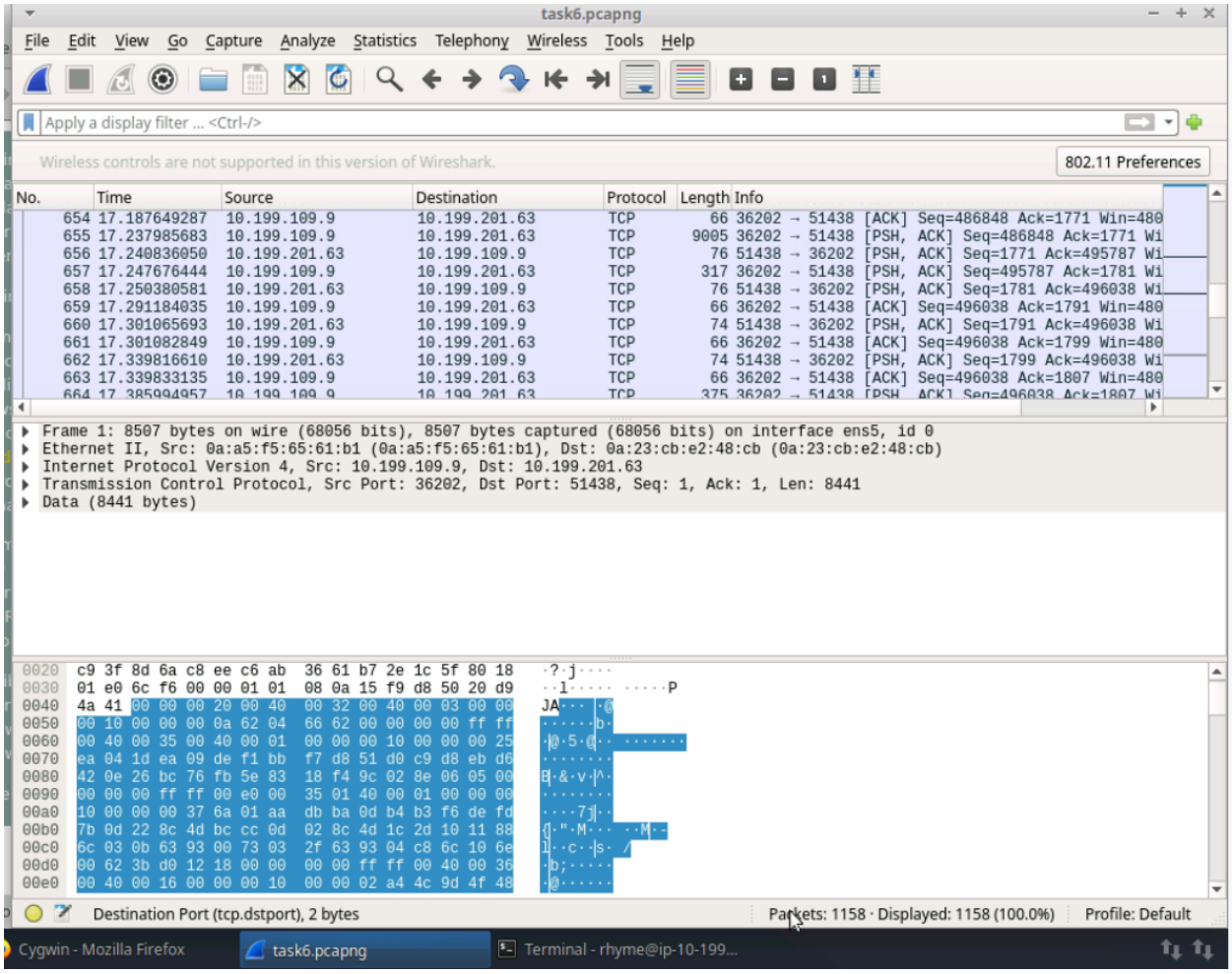
Launch Wireshark and begin capturing network traffic on the Ethernet interface. Ensure that Wireshark is set up to capture all relevant packets.

3. Clear the Browser Cache

Again, clear the browser cache to guarantee that the session data is not contaminated by cached resources.

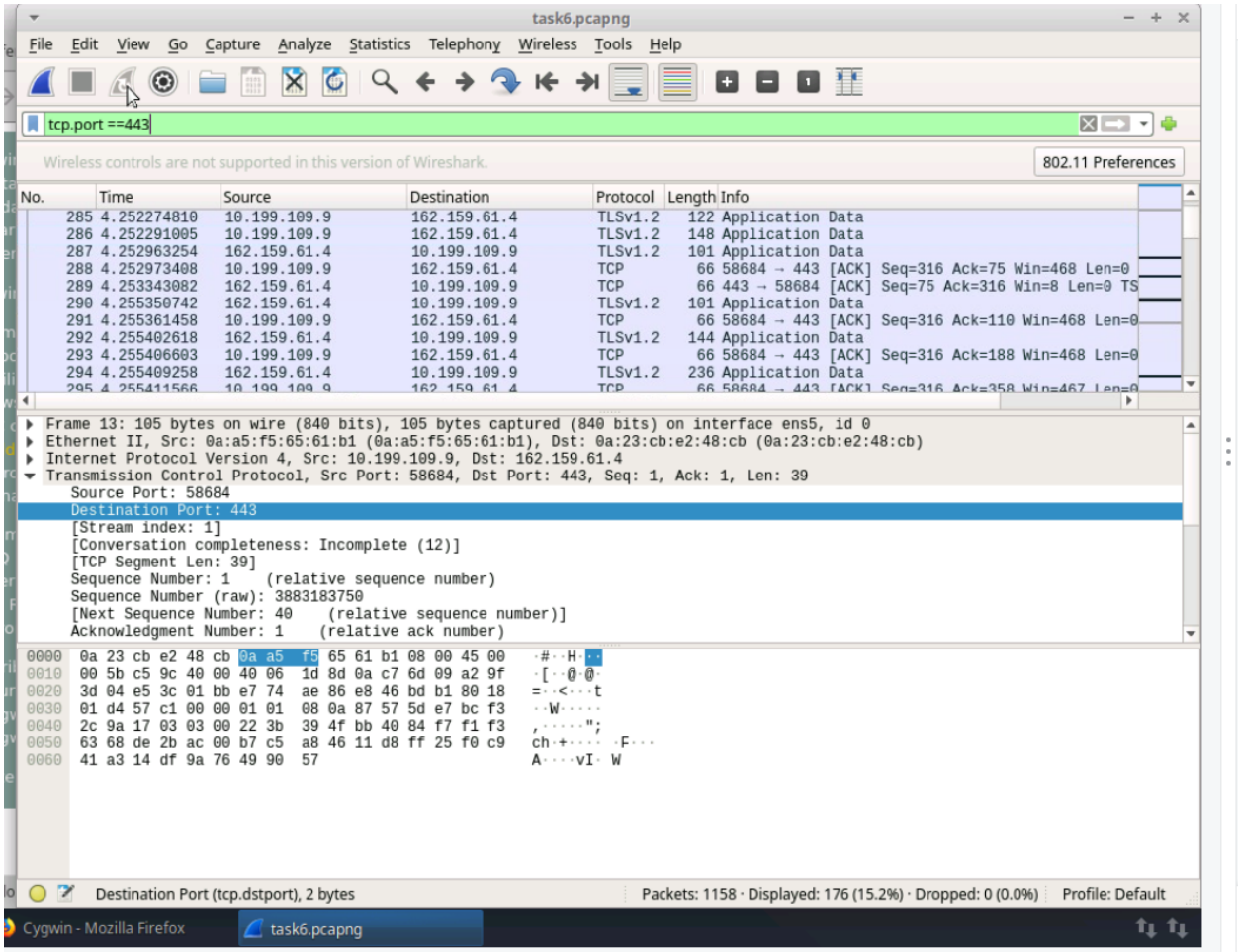
4. Stop the Packet Capture in Wireshark and Save

Once you have captured sufficient network traffic related to the HTTPS web application session, stop the packet capture in Wireshark and save the capture file for analysis.



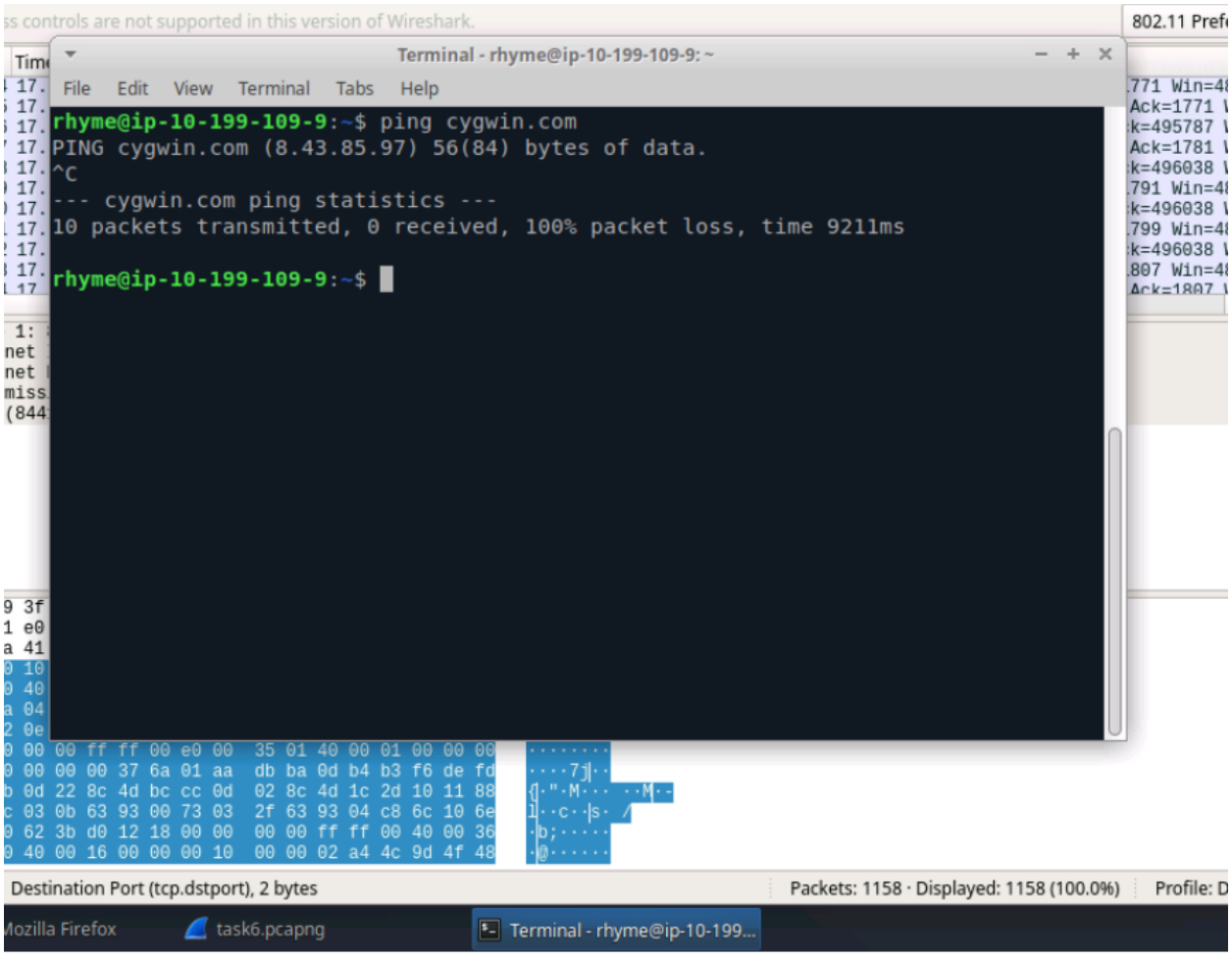
5. Create a Filter to Display Port 443 TCP Data

To narrow down the captured data to only display the relevant HTTPS traffic, create a display filter in Wireshark to show packets using port 443 (TCP), which is the standard port for HTTPS communication.



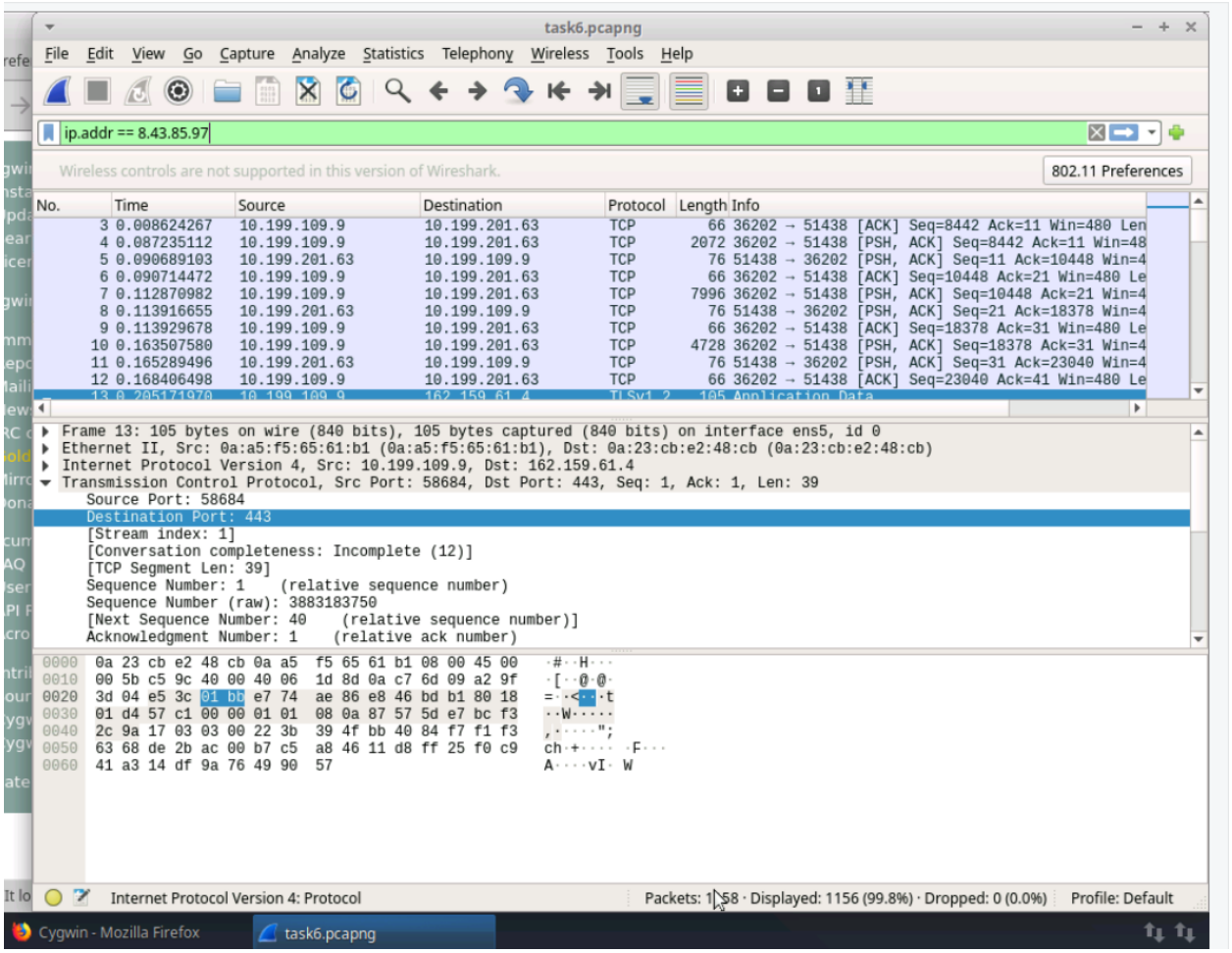
6. Create a Filter to View Only Traffic on Cygwin.com

To isolate traffic specific to the Cygwin.com domain, you can use the terminal and the **ping** command to find the IP address of Cygwin.com. Once you have the IP address, create a filter in Wireshark to view only packets involving that IP address.



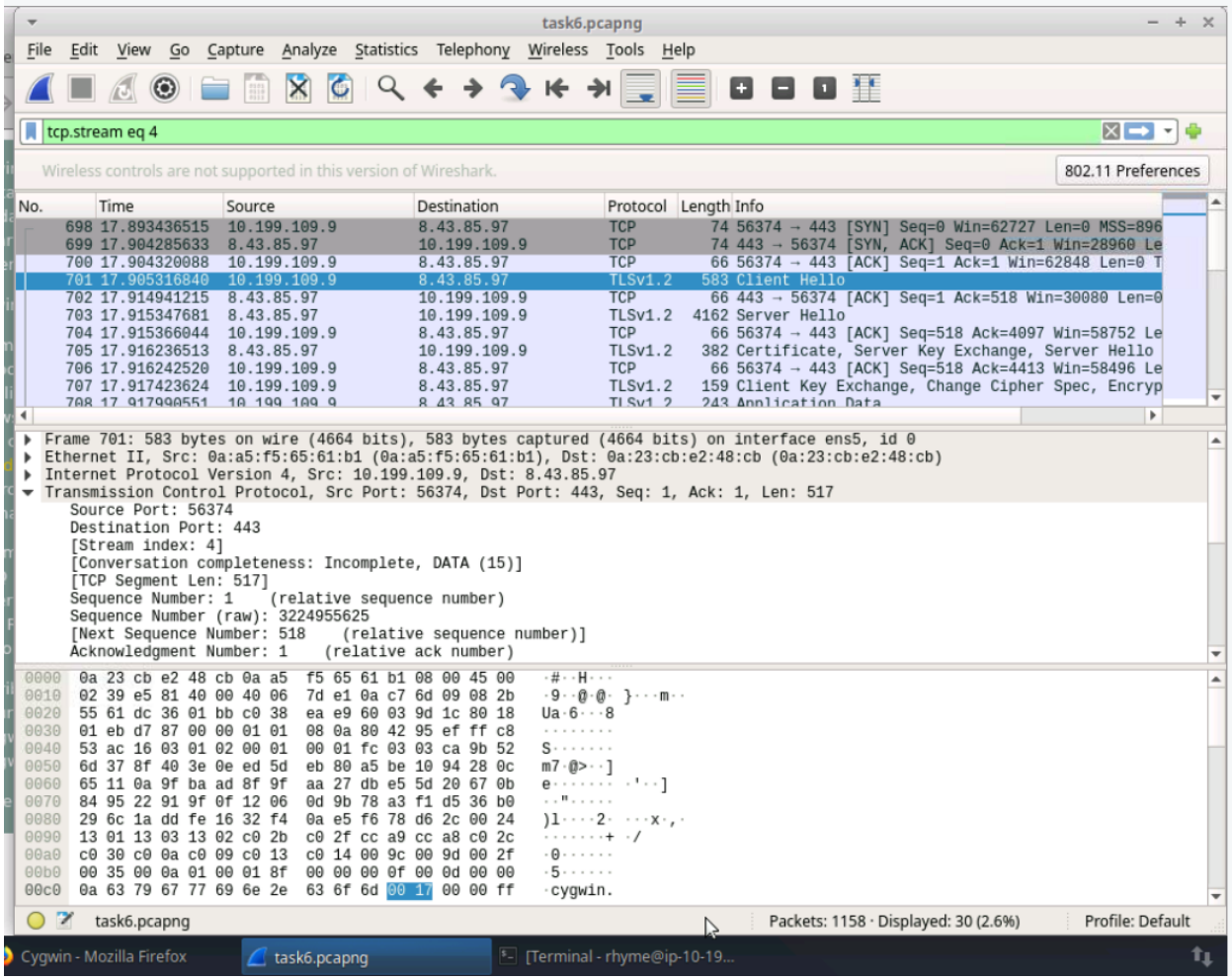
7. Use the Wireshark GUI to Follow the Cygwin.com Session Stream

Utilize the Wireshark GUI to follow the stream of the Cygwin.com session. This will provide a clear and organized view of the HTTPS communication between the client and the server.



8. Determine the TLS Version Used

In the Wireshark analysis, locate the TLS Handshake packets within the stream. Look for the "Client Hello" and "Server Hello" packets to determine the TLS version used. In this case, the TLS version was identified as TLS 1.2.



9. Note the Client and Server Key Exchange Mechanism

Within the TLS Handshake packets, identify the Client Key Exchange and Server Key Exchange messages. These messages are crucial for securing the communication session. Take note of the key exchange mechanisms used, such as RSA, Diffie-Hellman, or others.

Conclusion

In this report, we have outlined the process of analyzing a secure HTTPS web application session using Wireshark. This involves clearing the browser cache, capturing network traffic, filtering data, identifying the TLS version, and noting the Client and Server Key Exchange Mechanism along with their respective purposes. This analysis can provide valuable insights into the security and functioning of the web application's communication.