NETWORK SECURITY FUNDAMENTALS V2

Lab 4: Analyzing Packet Captures

Document Version: 2023-10-16

Contents

Introduction

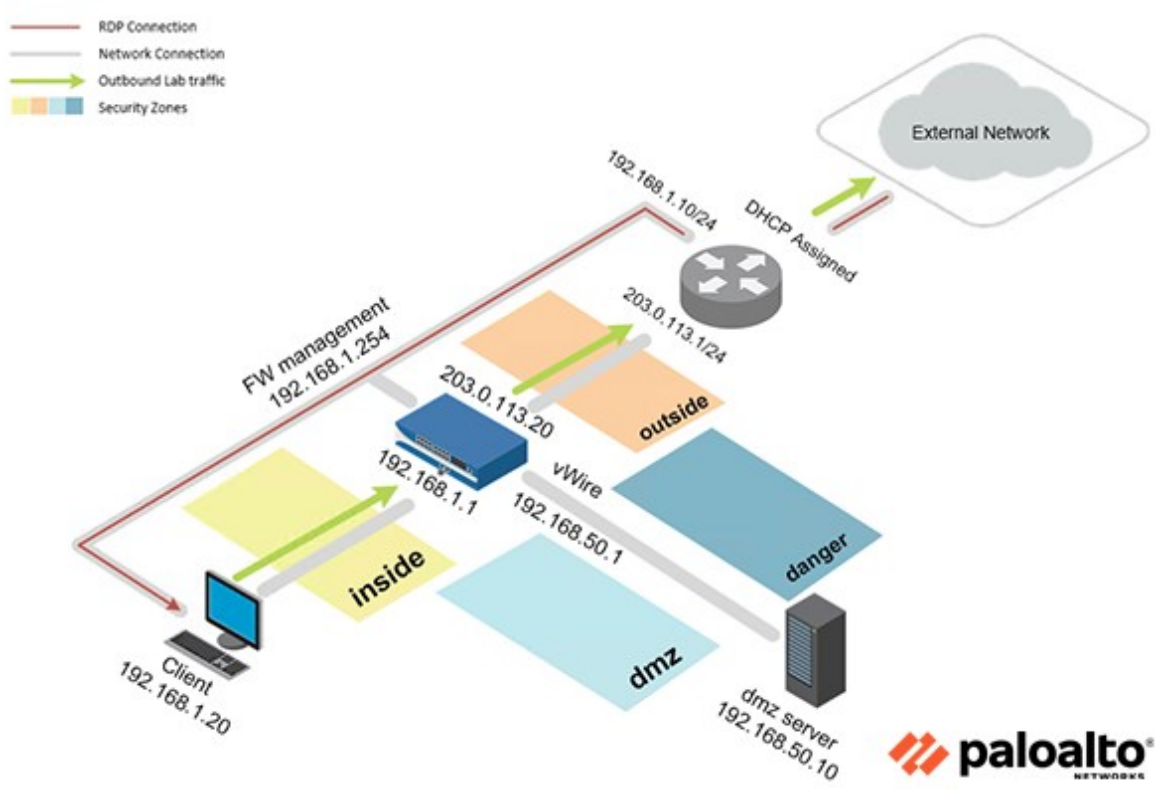
In this lab, you will configure *Wireshark* to decrypt SSL traffic and perform a packet capture to an SSL enabled site. Then, you will use *Wireshark* to explore capture files and examine the data within the packet.

Objective

In this lab, you will perform the following tasks:

- Configure Wireshark to decrypt packet captures from HTTPS sites
- Examine a HTTP Wireshark trace
- Examine a Wireshark decrypted packet trace

Lab Topology



Lab Settings

The information in the table below will be needed in order to complete the lab. The task sections below provide details on the use of this information.

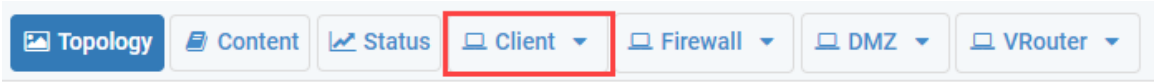
Virtual Machine	**IP Address**	**Account** (if needed)	**Password** (if needed)
Client	192.168.1.20	`lab-user`	`Pal0Alt0!`
DMZ	192.168.50.10	`root`	`Pal0Alt0!`
Firewall	192.168.1.254	`admin`	`Pal0Alt0!`

1 Analyzing Packet Captures

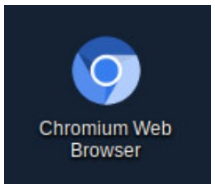
1.0 Load Lab Configuration

In this section, you will load the Firewall configuration file.

1. Click on the **Client** tab to access the Client PC.

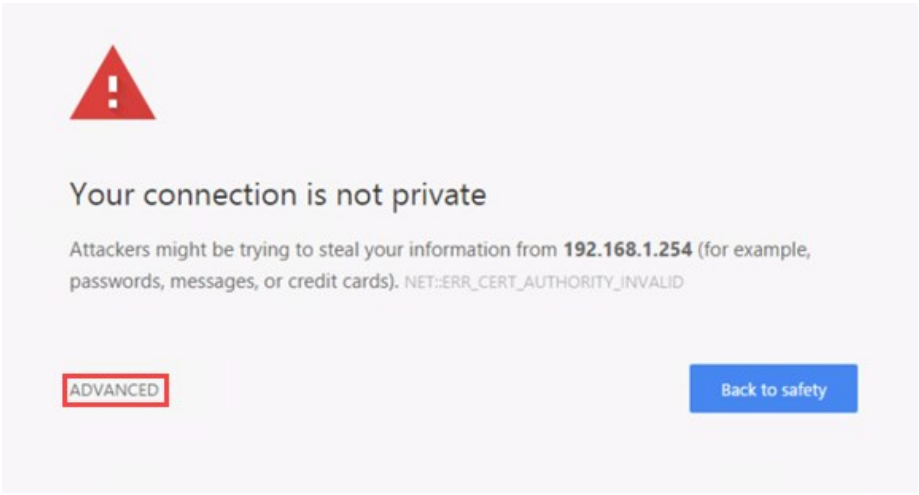


2. Log in to the Client PC with the username `lab-user` and password `Pal0Alt0!`.
3. Double-click the **Google Chrome** icon located on the Desktop.



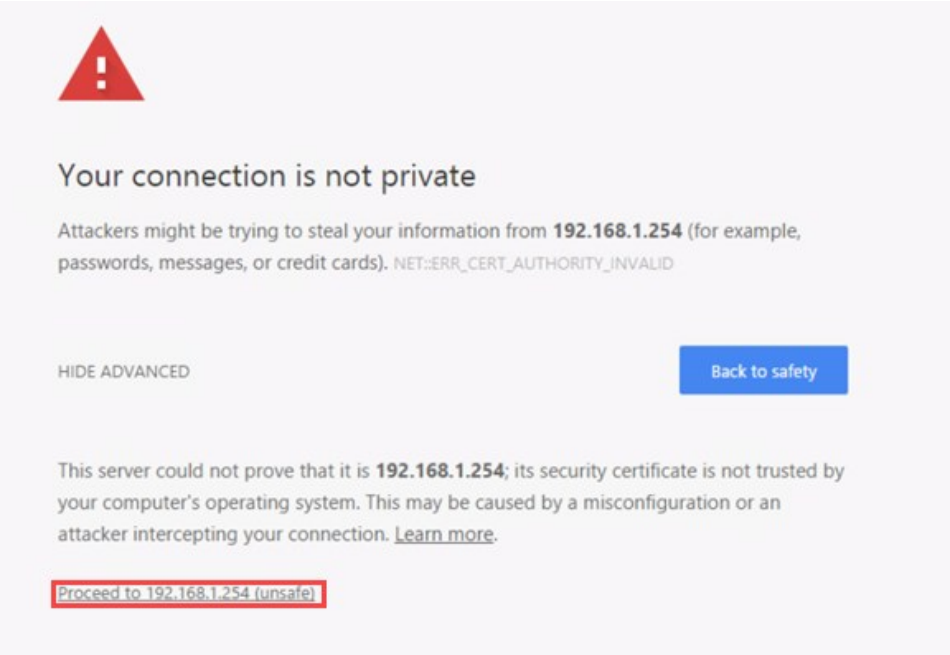
4. In the *Chromium* address field, type `https://192.168.1.254` and press **Enter**.

5. You will see a "*Your connection is not private*" message. Click on the **ADVANCED** link.



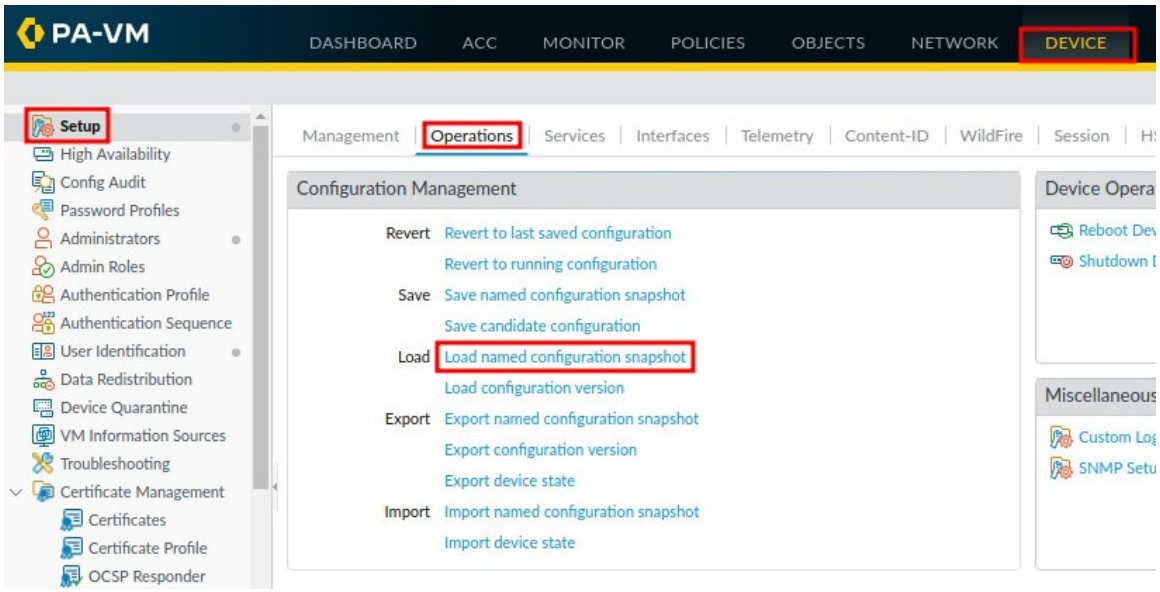
If you experience the "Unable to connect" or "502 Bad Gateway" message while attempting to connect to the specified IP above, please wait an additional 1-3 minutes for the Firewall to fully initialize. Refresh the page to continue.

6. Click on **Proceed to 192.168.1.254 (unsafe)**.

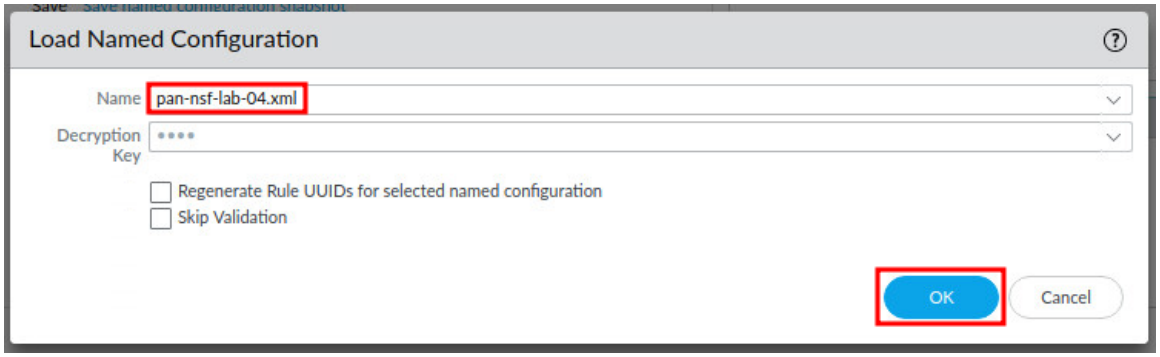


7. Log in to the Firewall web interface as username **admin**, password `Pal0Alt0!.`

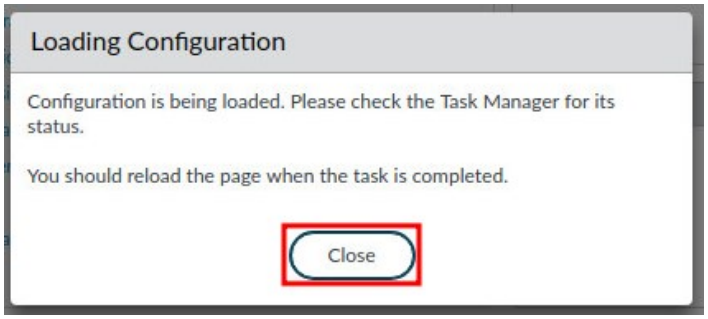
8. In the web interface, navigate to **Device > Setup > Operations** and click on **Load named configuration snapshot** underneath the *Configuration Management* section.



9. In the *Load Named Configuration* window, select **pan-nsf-lab-04.xml** from the *Name* dropdown box and click **OK**.

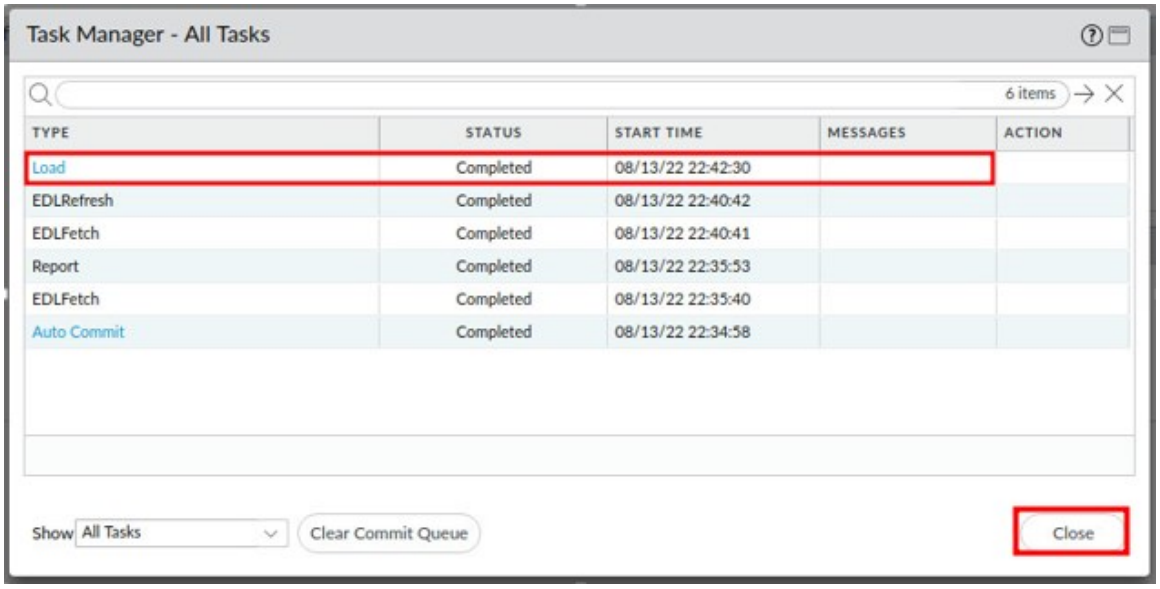


10. In the Loading Configuration window, a message will show *Configuration is being loaded. Please check the Task Manager for its status. You should reload the page when the task is completed.* Click **Close** to continue.

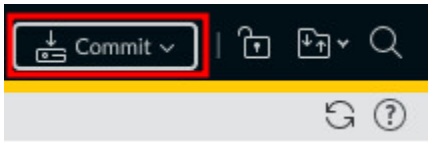


11. Click the **Tasks** icon located at the bottom-right of the web interface.

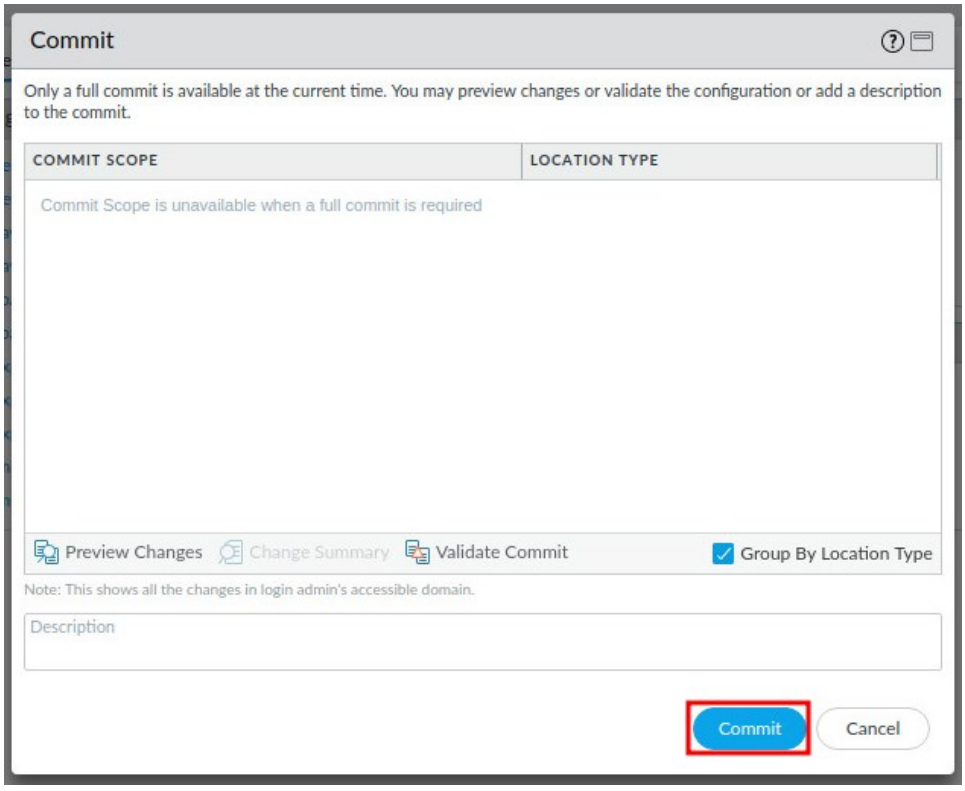
12. In the *Task Manager – All Tasks* window, verify the *Load* type has successfully completed. Click **Close**.



13. Click the **Commit** link located at the top-right of the web interface.



14. In the *Commit* window, click **Commit** to proceed with committing the changes.



15. When the commit operation successfully completes, click **Close** to continue.

The commit process takes changes made to the Firewall and copies them to the running configuration, which will activate all configuration changes since the last commit.

16. Close the web browser to continue on with the next task.

1.1 Configure Wireshark to Decrypt Packet Captures

In this section, you will configure Wireshark to decrypt SSL packet captures.

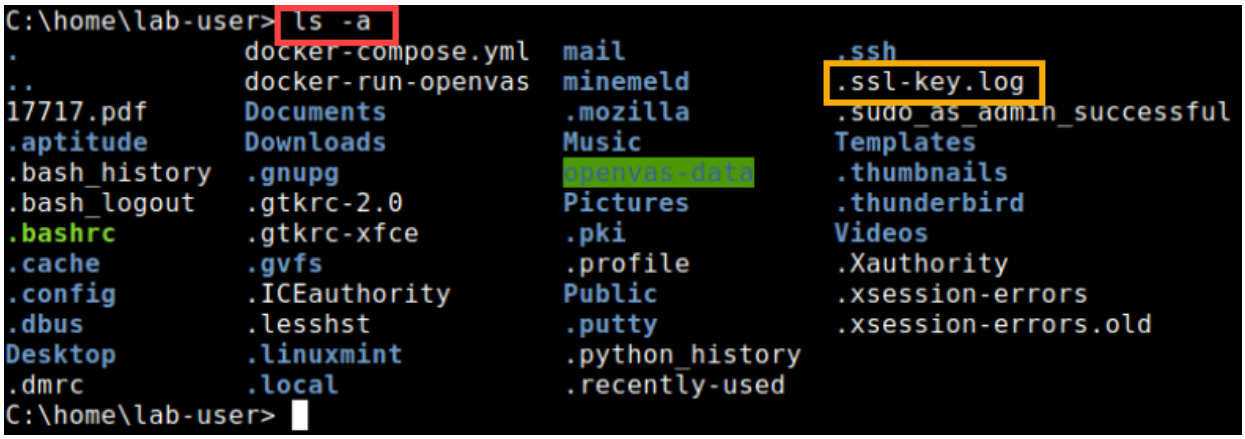
1. Open a new terminal by clicking on the **terminal** icon in the taskbar.

2. In the terminal, enter the command below to create a file named **.ssl-key.log**.

```
touch .ssl-key.log
```

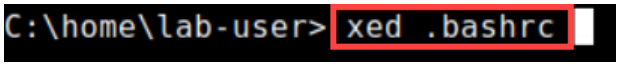
3. Enter the **ls** command below to verify the file you created is there.

```
ls -a
```



4. Enter the command below to open the **.bashrc** file using a text editor.

```
xed .bashrc
```



5. In the text editor, scroll all the way to the bottom, and at the end of the file, add the following line. Once finished, save your changes and close the text editor.

```
export SSLKEYLOGFILE=/home/lab-user/.ssl-key.log
```

```
# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
if ! shopt -oq posix; then
  if [ -f /usr/share/bash-completion/bash_completion ]; then
    . /usr/share/bash-completion/bash_completion
  elif [ -f /etc/bash_completion ]; then
    . /etc/bash_completion
  fi
fi
undercover && export PS1='C:${PWD//\//\\}> '
export SSLKEYLOGFILE=/home/lab-user/.ssl-key.log
```

6. Test the *bashrc* configuration by entering the command below in your terminal.

```
source .bashrc
```

```
C:\home\lab-user> source .bashrc
C:\home\lab-user>
```

7. Next, enter the follow-up command below. The response should be as shown in the screenshot below. If it isn't, go back and complete the steps above once more.

```
echo $SSLKEYLOGFILE
```

```
C:\home\lab-user> echo $SSLKEYLOGFILE
/home/lab-user/.ssl-key.log
C:\home\lab-user>
```

8. Confirm that the web browser is using the *ssl-key.log* file. In the terminal, enter the command below to open the web browser.

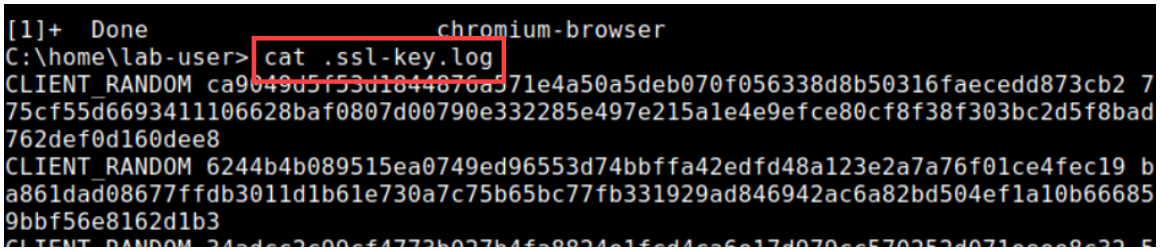
```
chromium-browser &
```

```
C:\home\lab-user> chromium-browser &
```


9. In the newly launched *Chromium* web browser, navigate to the URL: https://www.paloaltonetworks.com/academy. Once the website loads, wait one minute, and then close the web browser.

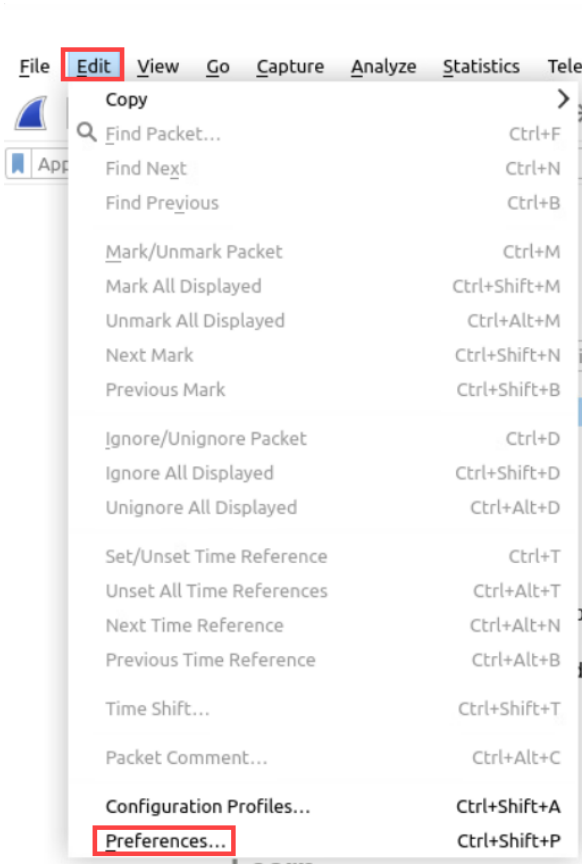
10. Back in the same terminal window, press the **Enter** key to return the terminal prompt and enter the command below. You should see an output similar to the screenshot below. If not, repeat the steps above.

```
cat .ssl-key.log
```

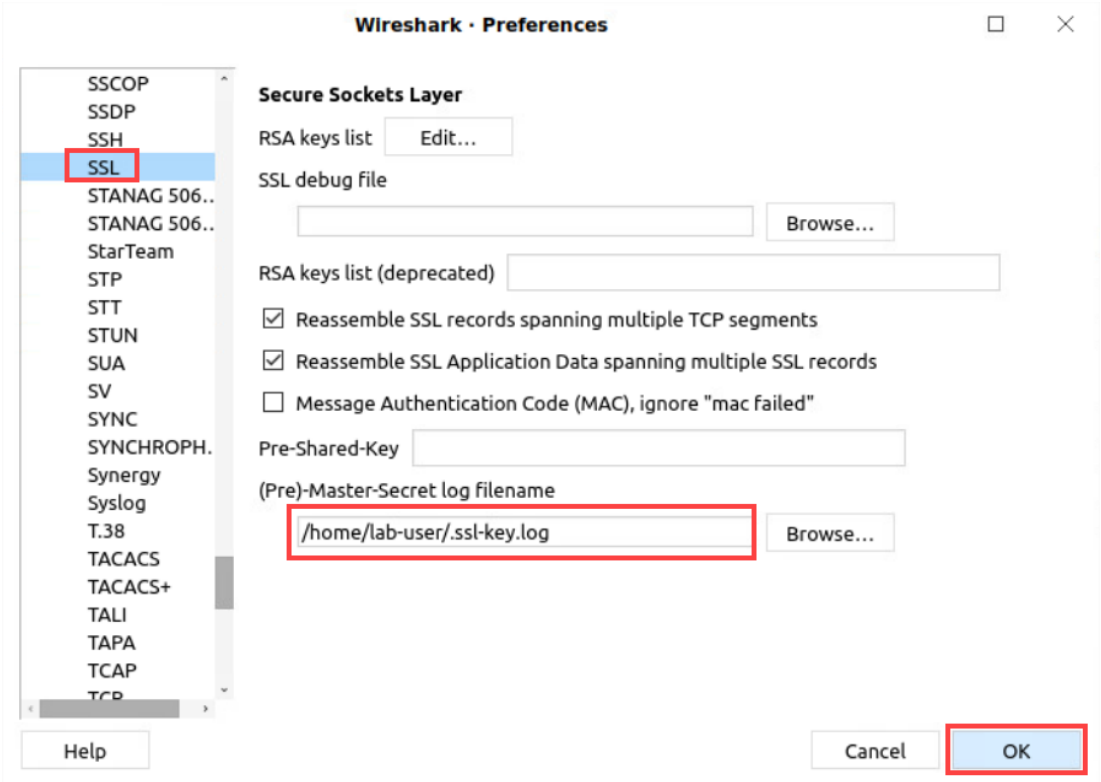


11. Minimize the terminal window and double-click the **Wireshark** icon on the desktop to open the application.

12. Once *Wireshark* launches, navigate to **Edit > Preferences**.

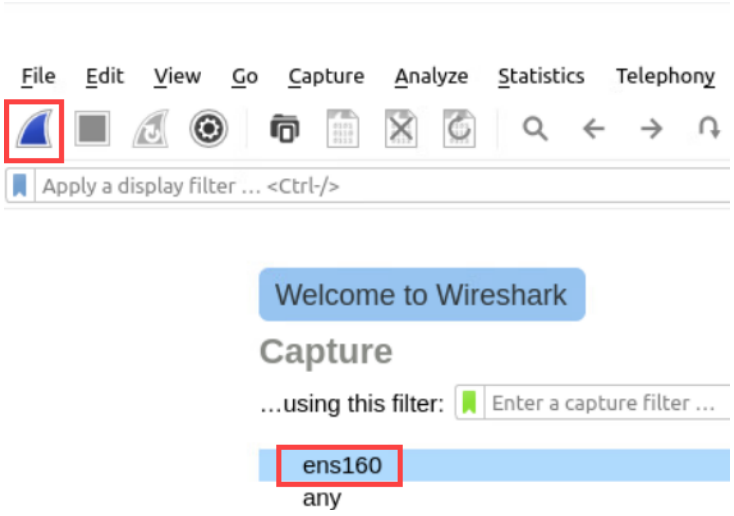


13. In the *Wireshark Preferences* window, expand **Protocols** in the left pane and then select **SSL**. Enter the full file path to the *.ssl-key.log* file in the *(Pre)-Master-Secret log filename* box: `/home/lab-user/.ssl-key.log`. Once finished, click **OK**.



14. Close the **Wireshark** application and make sure all opened web browser applications are closed.
15. Double-click on the **Wireshark** icon found on the Desktop to launch a new window.

16. In the *Wireshark* window, make sure that **ens160** is selected and click the **Start capturing packets** button.

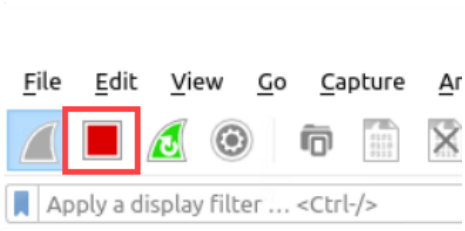


17. Change focus back to your existing terminal window and enter the command below to open a web browser.

```
chromium-browser &
```

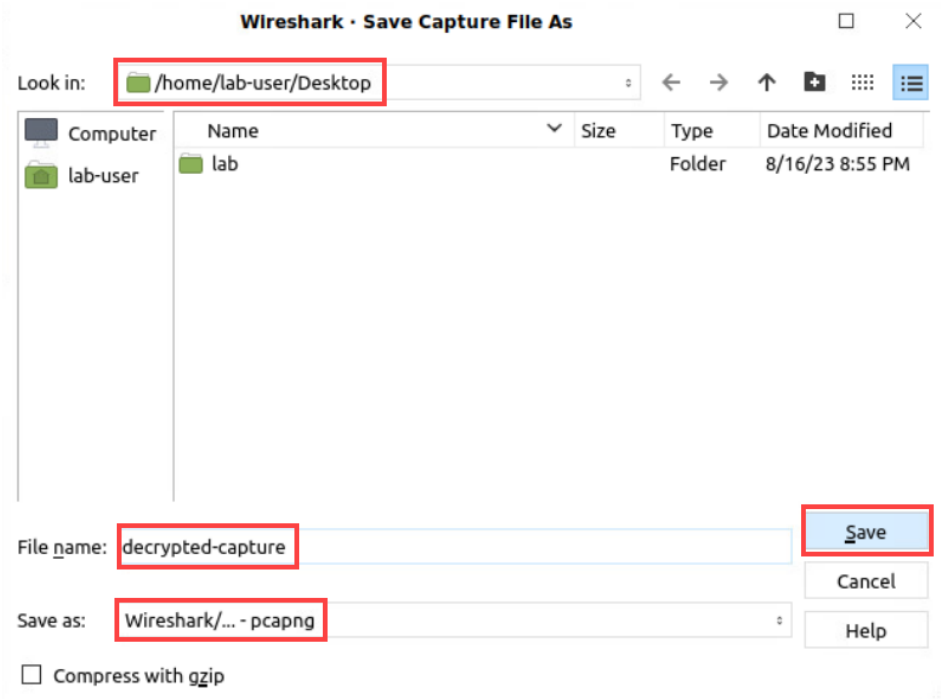
18. In the newly launched *Chromium* web browser, navigate to the URL: https://www.paloaltonetworks.com/academy. Once loaded, wait one minute.

19. Change focus to the Wireshark window and click the Stop capturing packets button.



20. Save the packet capture by navigating to **File > Save As**.

21. In the Wireshark Save Capture File As window, name the file `decrypted-capture` using **pcapng** as the filetype. Select the **Desktop** as the save to location. Click **Save**.



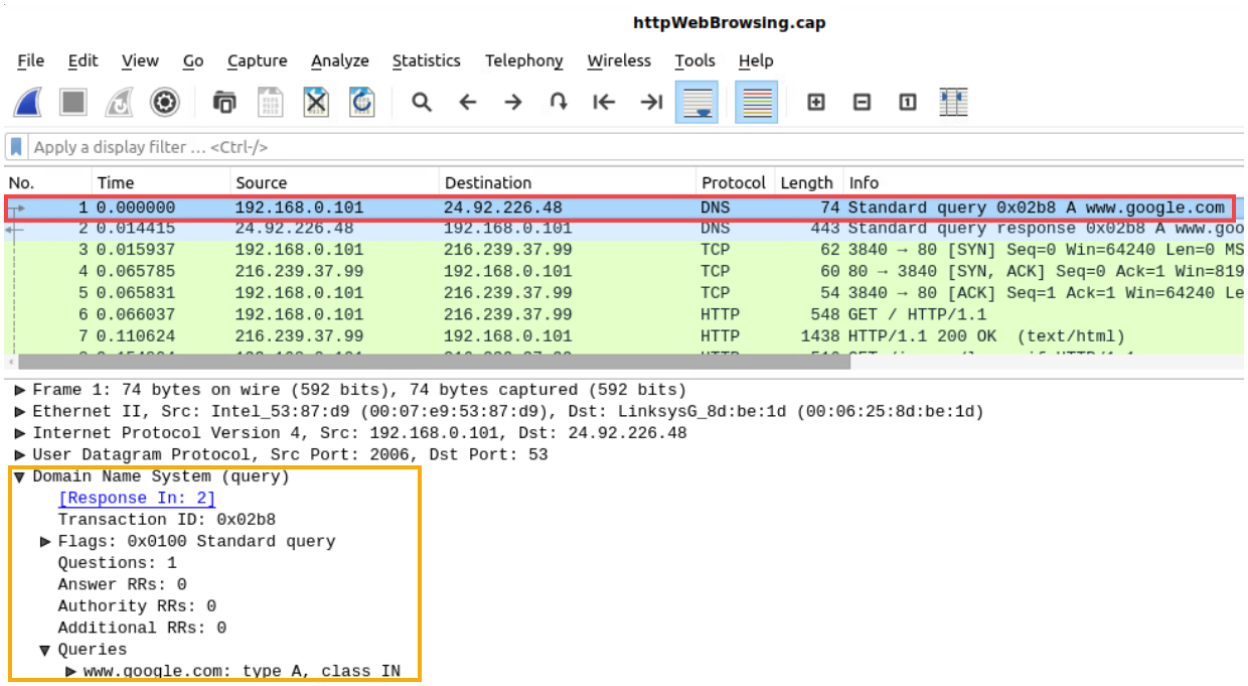
22. Close the **Wireshark** window as well as any open web browser windows.

1.2 Analyze PCAP Files with Wireshark

In this section, you will analyze the traffic capture using Wireshark.

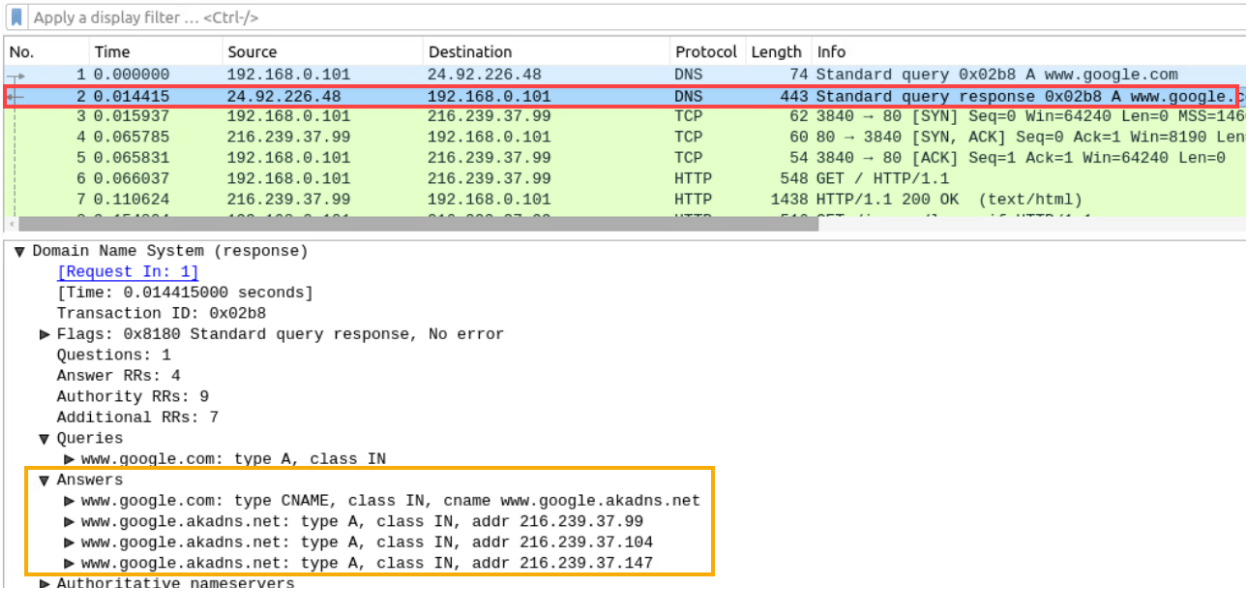
1. The first packet capture for examination is a packet capture that did not utilize an SSL/TLS encrypted connection. This will allow you to see the TCP/IP protocols without using complex Wireshark filters. You will then examine a decrypted packet capture for comparison. On the *Desktop*, double-click the **httpWebBrowsing.cap** packet capture to open it in Wireshark.

2. The first protocol you will analyze is DNS. Select the **first DNS packet** to examine it.



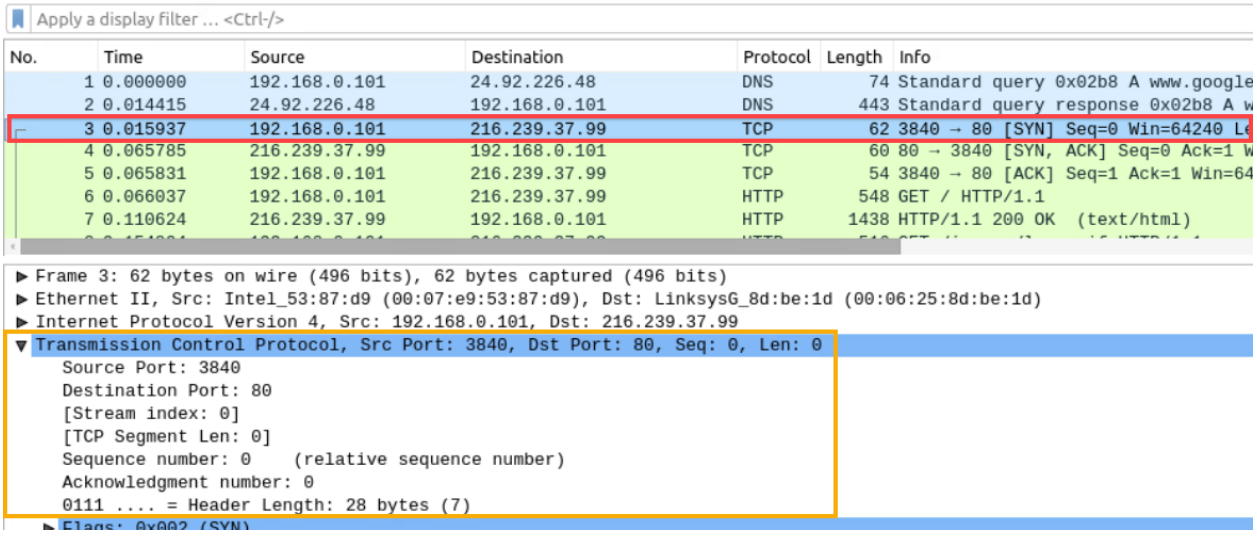
In the first packet, the source is *192.168.0.101*, while the destination is *24.92.226.48*. The source is using *24.92.226.48* as its *DNS* server. The info column shows it is a standard query asking for the A (IPv4) record for *ww.google.com*.

3. Select the **second DNS packet** and examine the contents.



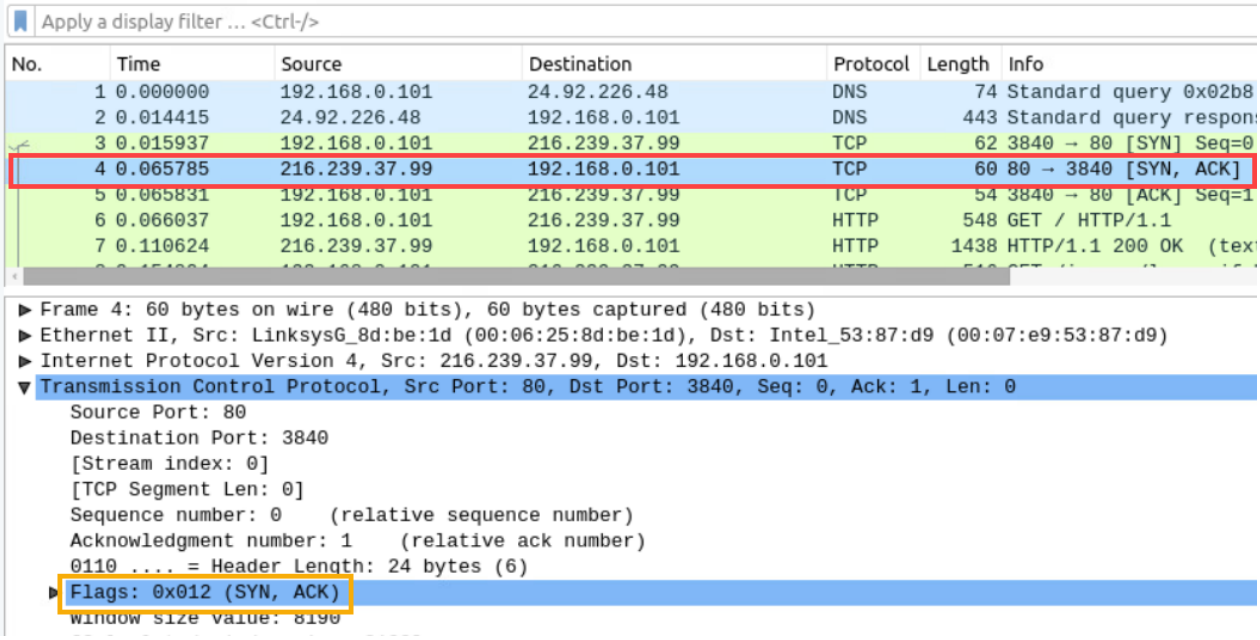
In the second packet, the source is the DNS server (*24.92.226.48*), while the destination is *192.168.0.101*. The info column shows a standard query answer for A record www.google.com. The answers show 3 IP addresses that could be used to connect to the www.google.com web server.

4. Next, examine the 3-way handshake between *192.168.0.101* and the Google Server (*216.239.37.99*). Select **packet #3**.



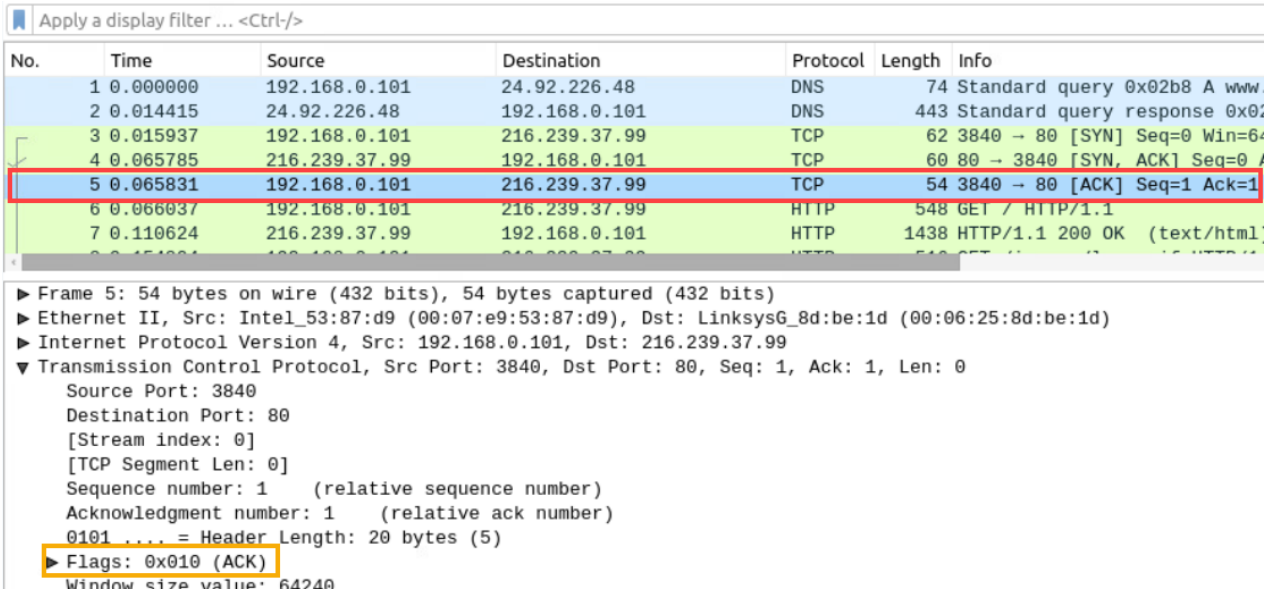
In the first part of the 3-way handshake, the source (*192.168.0.101*), sends a packet requesting a new session which is indicated by the TCP SYN flag being set in the header. To the destination *Google* web server (*216.239.37.99*).

5. Select **packet #4** and examine the packet information.



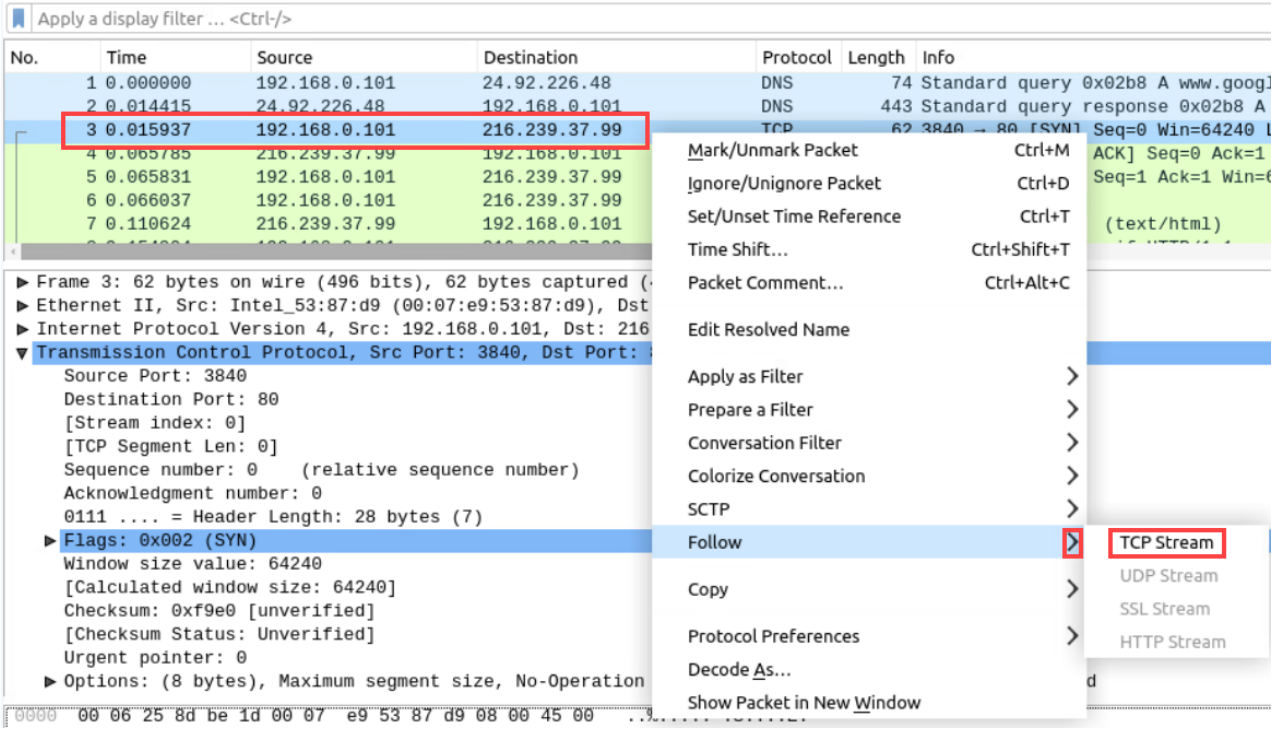
In the second part of the 3-way handshake, the source (*216.239.37.99*) sends a TCP packet with the flags *SYN* and *ACK* set in the header, to the destination (*192.168.0.101*). The server is acknowledging the original *SYN* packet it received and is confirming that it is ready to establish a new session with *192.168.0.101*.

6. Select **packet #5** and examine the packet information.

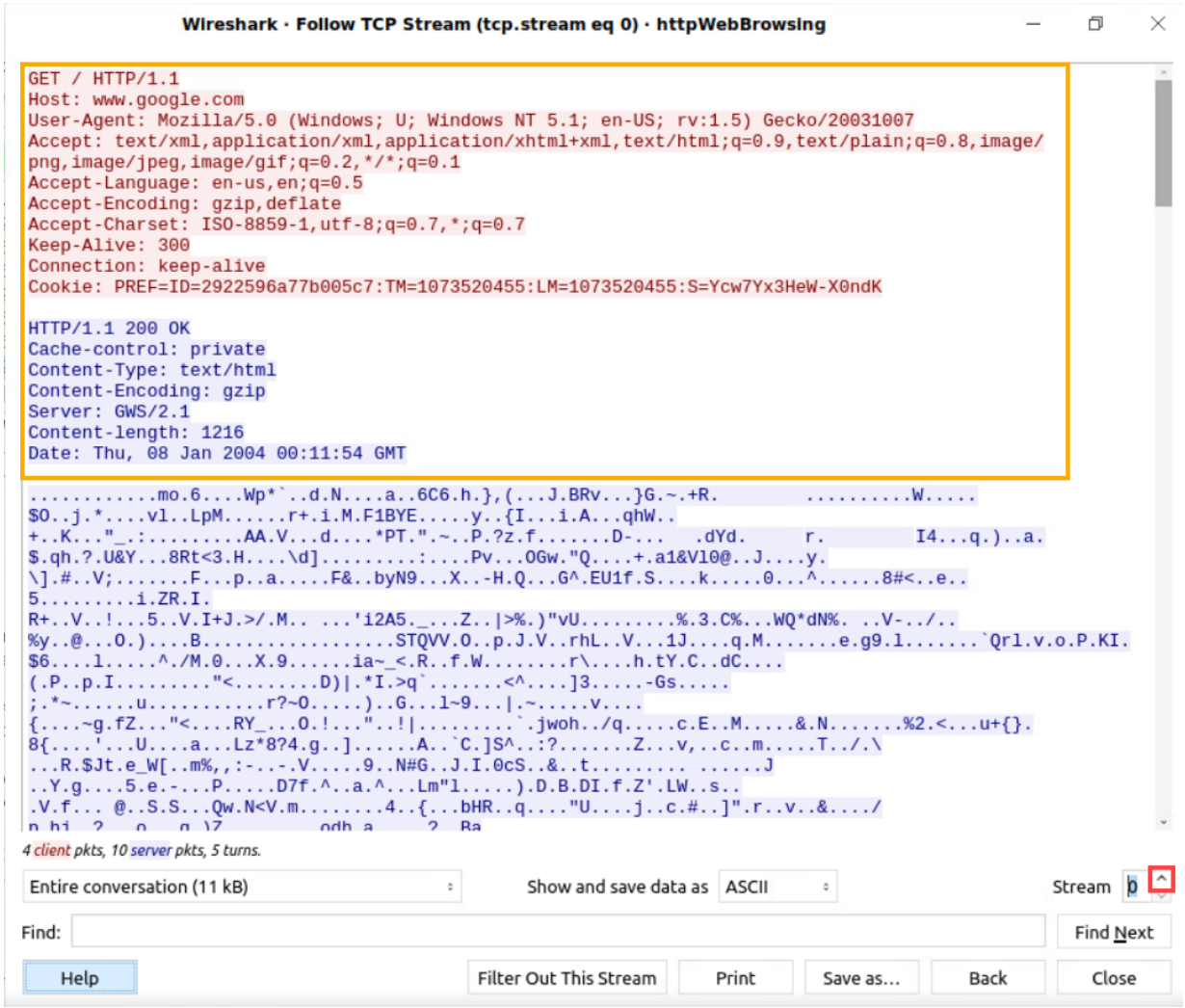


In the third part of the 3-way handshake, the source (*192.168.0.101*) sends a TCP packet with the flag *ACK* set in the header to the destination (*216.239.37.99*). The session between the two systems is now established for data transmission.

7. All the packets ranging from #3 - #20 represent the *TCP Stream* that makes up the communication between the two systems for the *TCP Established* connection. View all the packets comprising this session by right-clicking **packet #3** and selecting **Follow > TCP Stream**.



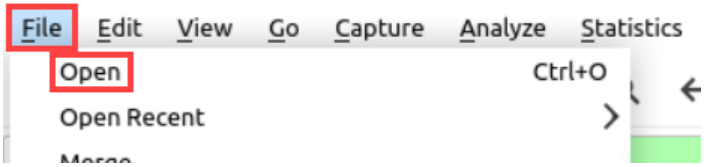
8. Observe the TCP Stream information presented in the new window. Cycle through the other TCP Streams in this capture by clicking the **top arrow** next to *Stream* on the bottom right of the window.



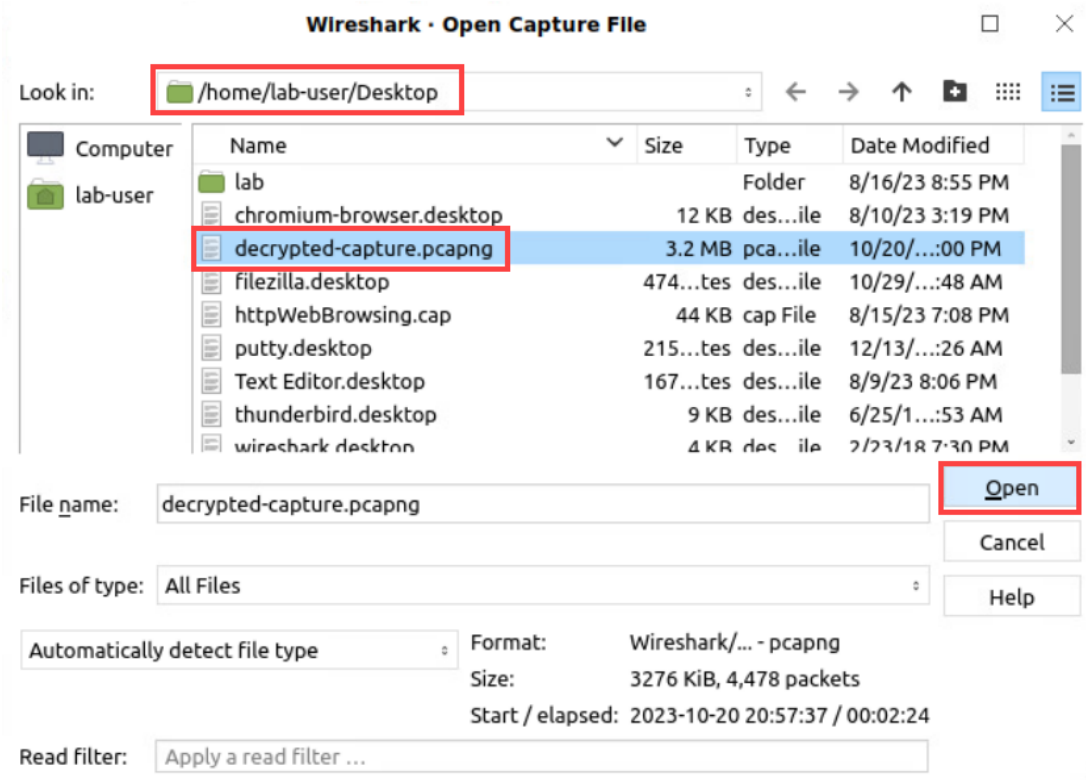
Notice the assembled packets represent the HTML website visited, with the red representing requests from the original source *192.168.0.101* and the blue representing the server destination *216.239.37.99*.

9. Close the **Follow TCP Stream** window.

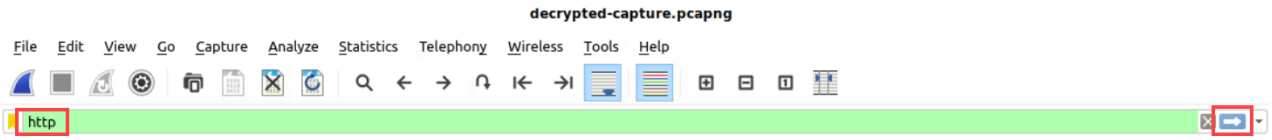
10. Almost all web traffic is encrypted by SSL/TLS and so to view all the protocols in a packet capture, you first need to decrypt it. You did this in a previous task and saved the packet capture to your *Desktop*. In the *Wireshark* window, navigate to **File > Open**.



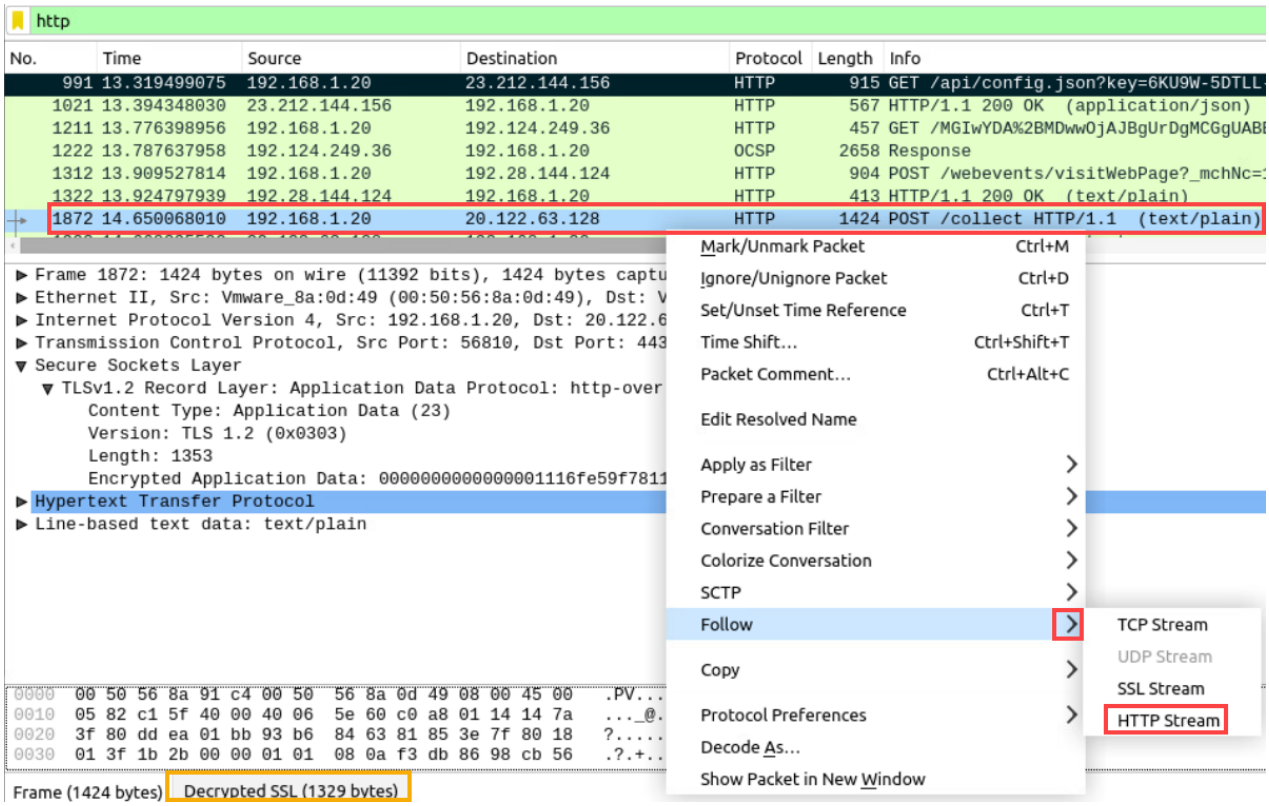
11. In the *Wireshark Open Capture File* window, view the **Desktop** directory, select **decrypted-capture.pcapnp**, and click **Open**.



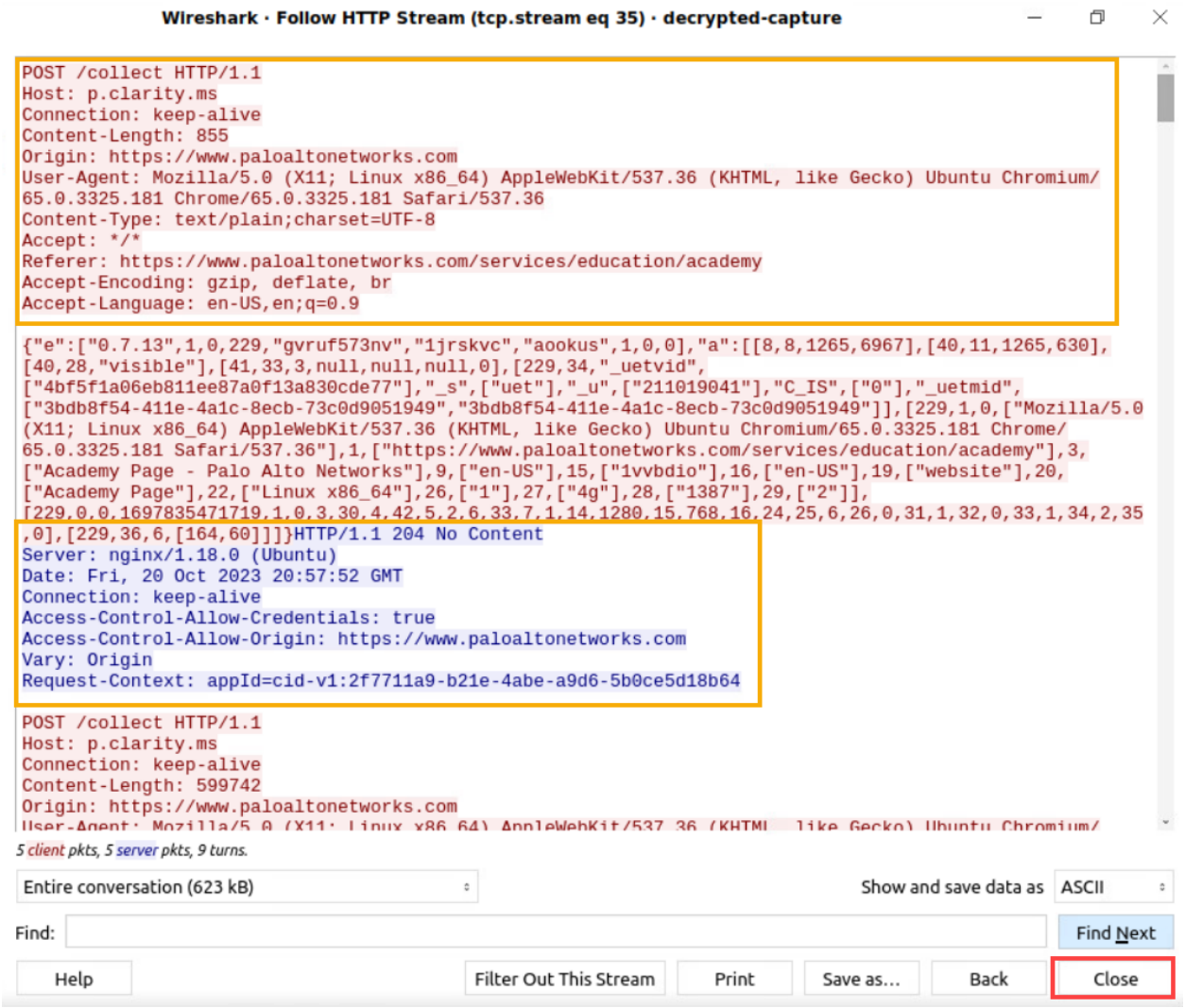
12. Type `http` into the *Wireshark* filter box and apply the filter.



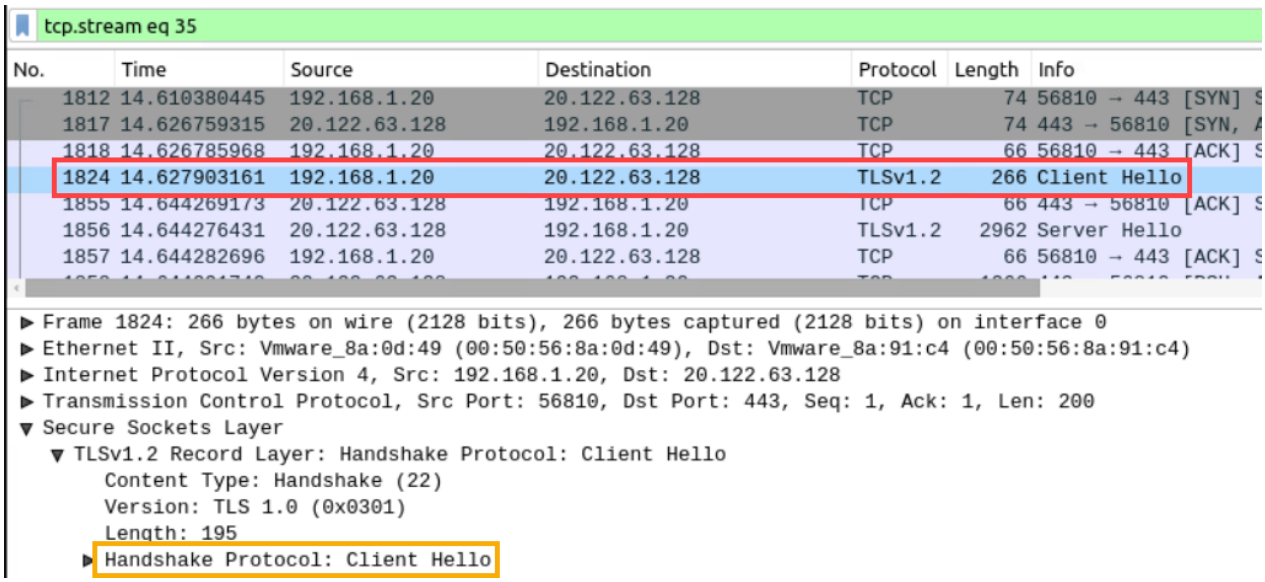
13. Identify the first **HTTP POST /collect** packet. Right-click the packet and select **Follow > HTTP Stream** to view the decrypted HTTP stream.



14. Observe the HTTP Stream information that is provided in the stream. Once finished, close the HTTP Stream window.

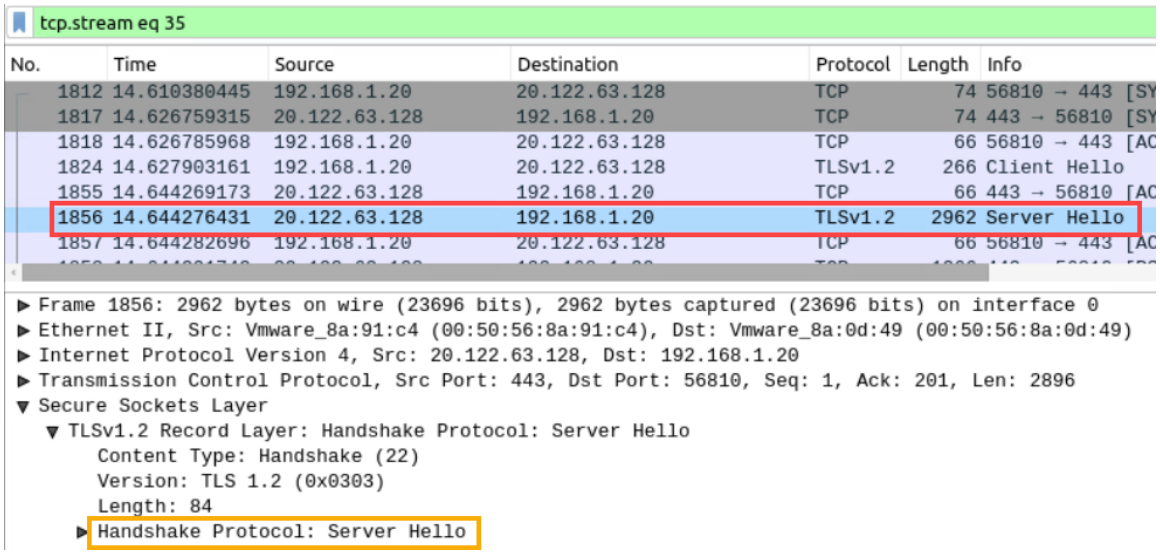


15. Notice when the window is exited, a filter is automatically pre-applied. Leave this filter applied as it will be different from each lab. Examine the SSL/TLS handshake between the Client and the server. Scroll up and identify the first **Client Hello** packet with the source *192.168.1.20* and select this packet. The source of this packet is the Client (*192.168.1.20*), and the destination is the server (*20.122.63.128*). This packet under the SSL layer tells the server that the Client is ready to establish an encrypted session.

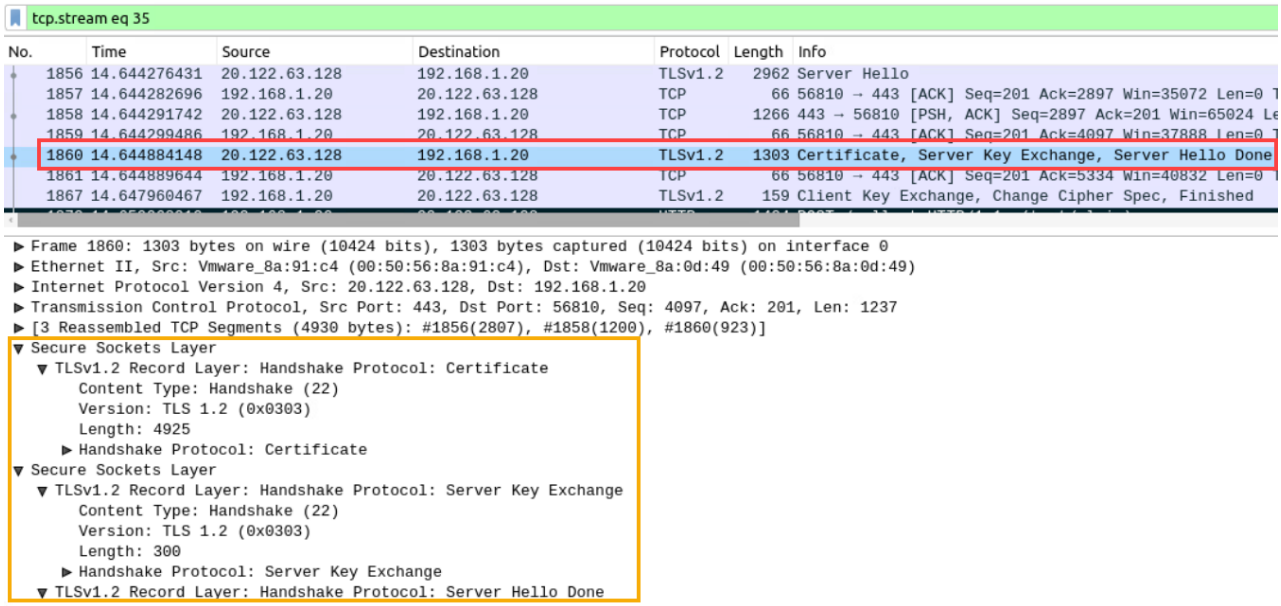


The destination IP addresses and packet capture numbers in your packet capture may differ from these examples due to the nature of the lab environment.

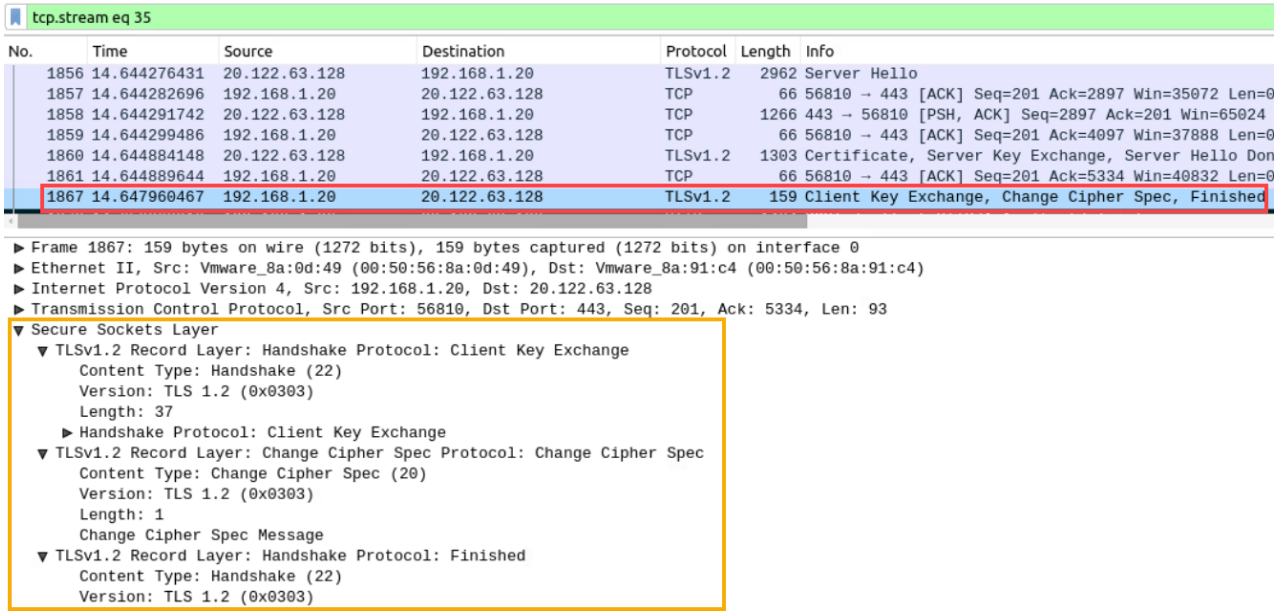
16. Identify the corresponding **Server Hello** packet that shares the same IP address from the destination now set as the source. The source of this packet is the server (*20.122.63.128*), and the destination is the Client (*192.168.1.20*). In this packet, the server acknowledges the Client's request.



17. Next, identify the follow-up packet from the server, **Server Hello Done**. In this packet, the server tells the Client what version of TLS to use and then sends its public certificate, including its public encryption key, to the Client.



18. Examine the corresponding **Client Key Exchange** packet. The source is the Client (*192.168.1.20*), and the destination is the server (*20.122.63.128*). When the Client receives the *Server Hello* packet, it first verifies that the server certificate is trusted. Then, the Client generates a session key using the Diffe-Hellman protocol. This session key will be used to encrypt and decrypt traffic from the server to the Client for the duration of the session. The Client securely sends this session key in this packet by encrypting the session key using the server's public key. When the server receives this packet, it will use its private key to decrypt and obtain the session key.



19. The lab is now complete; you may end the reservation.

