Hands-on Lab: Network Protocol Analyzers

Estimated time needed: **45** minutes

About This Lab

In this lab, we will be using Wireshark. Wireshark is an open-source app that can capture and display data that reads the contents of each packet as it travels across the network.

Objectives

In this hands-on lab, you will:

- Install Wireshark.
- Capture packets using Wireshark.
- Review capture results.

Exercise 1: Install Wireshark

Screenshots of the Wireshark installation were intentionally omitted. The installation was completed successfully using the standard download and installer from the official Wireshark website, which did not require any unusual configuration or advanced setup. Instead of documenting routine steps, I focused on the core learning objectives of the lab: performing live packet captures and reviewing traffic details using Wireshark's inspection tools. Screenshots and analysis of the packet capture session are included below.

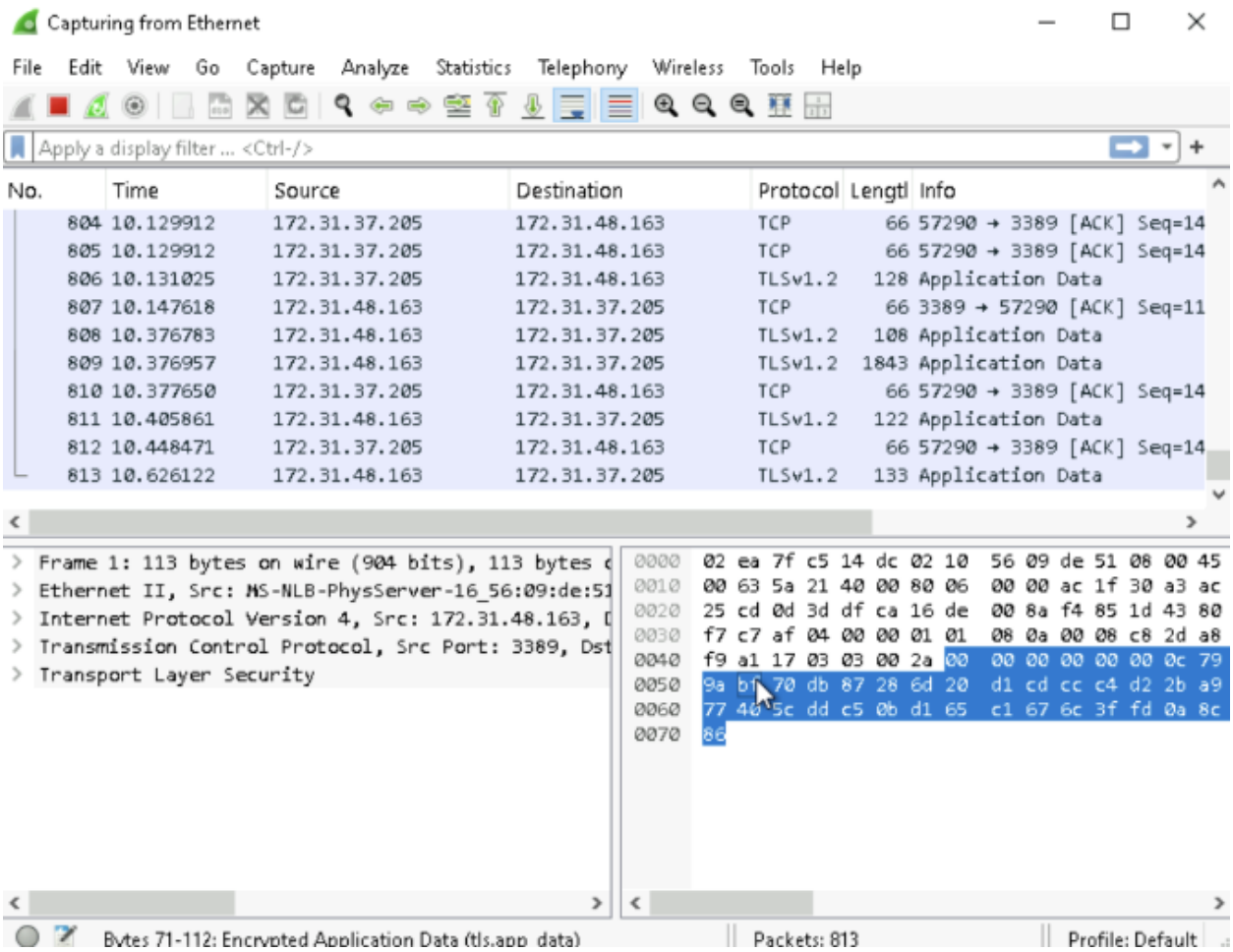
In this exercise we will install Wireshark.

1. Open Chrome browser.
2. Navigate to **wireshark.org**.
3. Click **Download**.
4. Click **Windows x64 Installer**.
5. Click the wireshark executable file to start the installation wizard.
6. Click **Next** to continue.
7. Click **Noted** to acknowledge licensing guidelines.
8. Click **Next** to continue.
9. Click **Next** to keep the recommended components to install.
10. Click the box next to **Wireshark Desktop Icon**, then click **Next** to continue.

11. Click **Next** to keep the default destination folder as C:\Program Files\Wireshark.
12. Click **Next** to continue.
13. Click **Install**.
14. Installation may take several minutes and click **I Agree**
15. Once installation is complete, click **Next** to finalize installation.
16. Click **Finish**

Exercise 2: Capture Packets using Wireshark

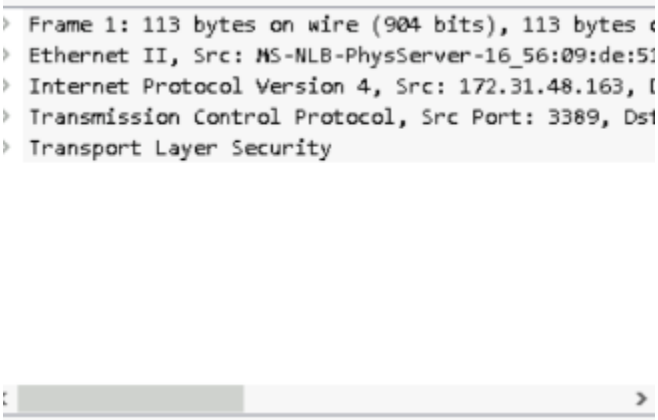
1. Double-click the **Wireshark** icon on the desktop to run the application.
2. The first screen shows the network(s) available on this computer. Select and highlight **Ethernet 3**.
3. Click **Capture**.
4. Select **Start** from the dropdown menu.



5. Wireshark will start to collect information about all packets that are sent and received. Click the Red **Stop Capturing Packets** icon located next to the shark fin.

Exercise 3: Review the Output

1. Familiarize yourself with the packet list pane.

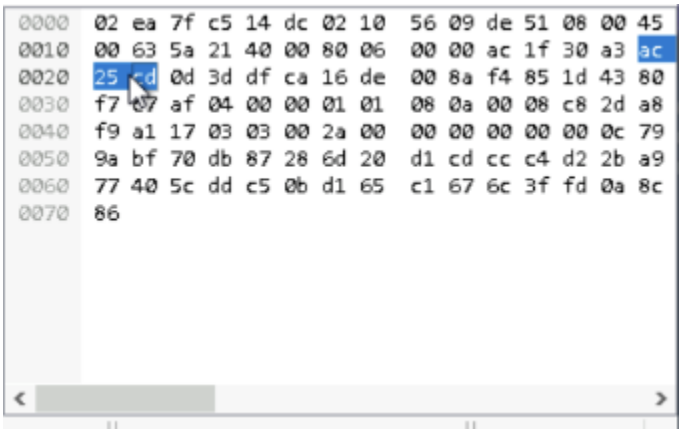


Note the following:

The packet list pane (the top pane) shows packets that were found. Every packet has its own row along with an assigned number. The following data points are included in this pane:

- **No.**: This is blank until you click on a packet. A checkmark will appear in the **No.** column to indicate other packets that are a part of the same conversation as the selected packet.
- **Time**: This indicates when the packet was captured. The default format for this entry is the number of seconds or partial seconds since the capture was created. This can be changed to a time-of-day format.
- **Source**: This indicates the IP address where the packet was sent from.
- **Destination**: This indicates the address that the packet was sent to.
- **Protocol**: This indicates the packet's protocol (i.e. TCP).
- **Length**: This indicates the packet length in bytes.
- **Info**: This indicates any additional details about the packet (if applicable).

2. Familiarize yourself with the packet details pane (the bottom left pane).



1. This pane shows protocols and protocol fields of the selected packet. You can apply filters based on specific details by right-clicking on an item.
3. Familiarize yourself with the packet bytes pane (the bottom right pane).

This pane shows the raw data of the selected packet. The data is in hexadecimal format (a number system using 0-9 and A-F). To change the format from hexadecimal to bits, you can right-click anywhere within the pane and select **as bits**.

Summary

In this lab you installed Wireshark, captured packets that were entering and exiting your network interface, and reviewed the resulting output

My summary

In this lab, I installed Wireshark and successfully captured live network packets traveling through my system's Ethernet interface. While I did not document each step of the installation process (as it followed a standard download and install from the official Wireshark website), I focused on the core objectives of the lab: initiating packet capture and analyzing the results.

During the capture session, I collected packets in real time and used Wireshark's interface to examine key packet details, including source/destination IPs, protocols (TCP, TLSv1.2), ports, and payload size. I reviewed the packet list pane, the detailed protocol breakdown pane, and the raw hexadecimal output in the bytes pane. Several packets using encrypted TLS and TCP protocols were observed, and their structure and handshake patterns were analyzed for common communication behaviors.

This exercise helped reinforce how packet analysis works at a low level and highlighted the usefulness of Wireshark for both security analysis and general network troubleshooting.