Lab: Wireshark

Introduction

The purpose of today's lab is to solidify your background in the 'nuts & bolts' of Internet technologies. It will also give some empirical experience in 'peeling back layers' of the Internet.

This should hopefully be a fun bit of poking around with what your computer is actually doing all the time — for better or worse, I always find something new every time I look at the firehose of packets coming in and out of my machine.

Warning: In general, be careful when sniffing traffic. It can be illegal to monitor communications you were not supposed to have access to.

Goals

- Set up Wireshark on your system
- Understand how to use Wireshark to inspect communication
- Explore communication on your computer

Equipment

- Computer

Partners

- This lab should be done individually

Table of Contents

List of Checkoffs

You must be checked off by course staff to receive credit for this lab. This can be the instructor, TA, or PM during a Friday lab session or during office hours.

- Section 2: Explain interfaces in Wireshark
- Section 3: Wireshark practice
    a) HTTP request/response
    b) DNS request/response
- Section 4: Ping packets
- Section 5: Intentional traffic
- Section 6: Unknown traffic

1. Install Wireshark

Wireshark is available here: https://www.wireshark.org/

Wireshark works on Windows, MacOS, and Linux. You can install it right on your host or inside a virtual machine if you prefer. Virtual machines will take a little extra care to make sure they can access network interfaces on your computer.

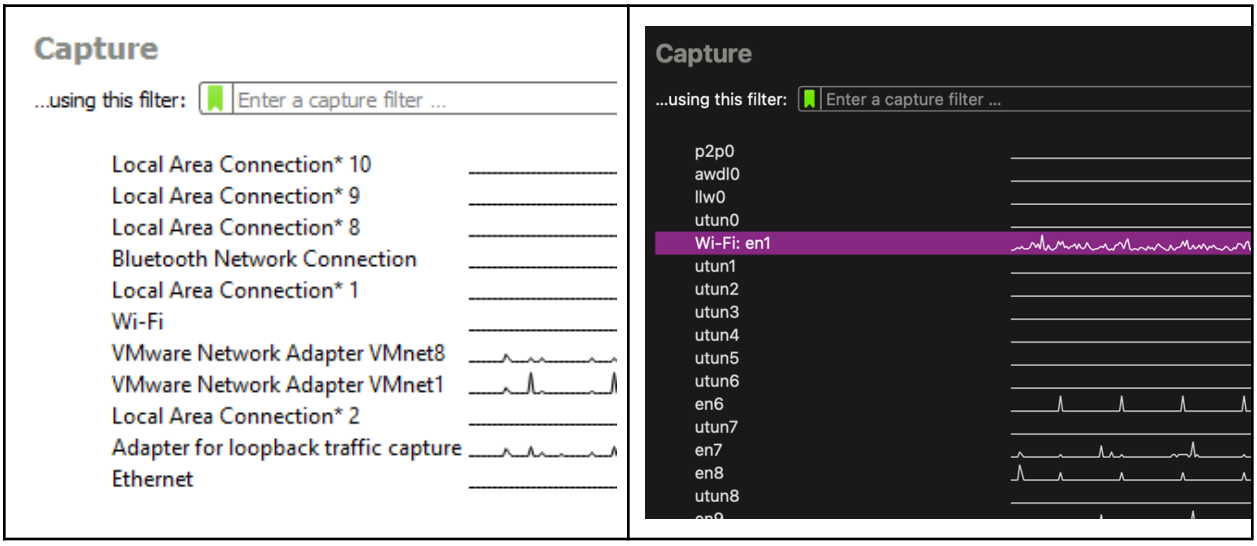
Sometimes, you can run into some permission headaches getting wireshark access to your network traffic. The modern installers are pretty good at getting all the permissions it needs, but if you have issues, Google is the best place to go.

Note: It is not a good idea to run Wireshark as root/administrator — it'll get all the packets, sure, but that's really opening yourself up for trouble. See more details here: https://superuser.com/questions/139206/concern-over-running-wireshark-as-root

CHECKOFF: None. Continue to the next section.

2. Understanding Interfaces

When you start Wireshark, it provides a list of interfaces that can be used to capture packets. Depending on your OS, you might get rather cryptic names



CHECKOFF: Explain what physical or digital thing each of the interfaces on your machine corresponds to (e.g., “en1 is my WiFi card”)

- Group interfaces as appropriate.
- If you’re not sure about what an interface is, look around on Google for a bit. If you’re still not sure, answer “Don’t know”. Don’t spend too long stuck on any one interface.
- TIP: we can do multiple checkoffs at once. So if it’s busy, just move on to the next step and come back to this later.

3. Wireshark Practice

Especially if you've never used Wireshark before, Jim Kurose (he wrote the Networks textbook) has some excellent labs that can help you understand and practice with it. I strongly recommend you walk through these. It'll only take like 20 minutes to do so and they will teach you a lot about how Wireshark works.

- Getting Started Lab:
  http://www-net.cs.umass.edu/wireshark-labs/Wireshark_Intro_v8.0.pdf

- DNS Query Lab:
  http://www-net.cs.umass.edu/wireshark-labs/Wireshark_DNS_v8.0.pdf

There are various other labs also available:
https://gaia.cs.umass.edu/kurose_ross/wireshark.php

CHECKOFF: Demonstrate capturing an HTTP request/response in Wireshark.

- **Note:** some students have found HTTP packets were disabled by default. They had to go to Analyze->Enabled Protocols and then enable HTTP.

- **Note:** some students have found that HTTP packets don't show up at all (even after enabling HTTP packets). If you're using HTTPS in your browser, Wireshark doesn't label HTTPS traffic with the HTTP protocol. Instead it will show up under TCP or TLS protocol (you can filter with "tcp" or "ssl"). To see HTTP packets you can send a request using plain, insecure HTTP. You can go to something like http://neverssl.com/. Just make sure that the URL has "http://" at the start. Alternatively, you can send a regular HTTP request using the "curl" terminal command. If that doesn't work and you are using the Chrome browser, you might want to disable QUIC packets (in Analyze->Enabled Protocols).

- TIP: If you're doing this outside of lab, save the Wireshark trace so you can open it up again when you're going to get the checkoff.

CHECKOFF: Demonstrate capturing a DNS request/response in Wireshark.

- **Note:** Firefox sends DNS traffic over HTTPS requests (DoH) by default, so you'll have to use a different browser (or a terminal) to make the request if you want to see DNS traffic.

4. Inspect Ping Traffic

1. Open a terminal window (this works in Command Prompt on Windows too) and run:

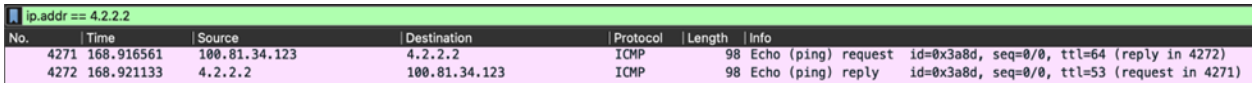
   ```
   ping 4.2.2.2
   ```

2. Open wireshark and start collecting traffic on your default interface.

3. Add a filter:

   ```
   ip.addr==4.2.2.2
   ```

4. You should see something like this:



5. Explore a bit in Wireshark and see what you can learn about the packets you're observing.

CHECKOFF: Understand the Ping traffic. Describe the following:

- What does ICMP stand for?
- For one of your *ping* packets, start from the PHY and list each of the layers that were used to send the packet, and which technology was used. Showing the packet data in Wireshark would be helpful here. (You might notice one of your layers being labeled as "Ethernet II" when you're not using Ethernet at all! You can read more here.)

5. Investigate Intentional Traffic

Other than ping, use some application that you know will communicate over the internet and use Wireshark to find that communication. Pick something other than just visiting a website. You could play a Youtube video, play new songs on Spotify, start a video game, use Zoom, etc. Closing other non-essential applications on your computer can be helpful here to narrow down the possibilities.

CHECKOFF: Explain your investigation:

- What does an example packet look like?
- What method did you use to find traffic from that application?
- Did you notice anything interesting about the traffic?

6. Investigate Unknown Traffic

Pick some traffic that looks interesting to you and investigate what communication is occurring. Maybe an interesting Protocol or to a Source/Destination you don't immediately recognize. Can you determine what the purpose of the communication was?

CHECKOFF: Explain your investigation.

- What does your unknown packet look like?
- How did you determine what the traffic corresponded to?
- Explain something interesting you noticed about the traffic.