

Lab: Wireless

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

Submission

- Write your answers up for each task and submit a PDF to [Gradescope](#).

Remember: I'm not looking for a formal lab report. Just your answers in any format that makes sense. The goal is to prove that you did the lab and spent some time thinking about it.

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.

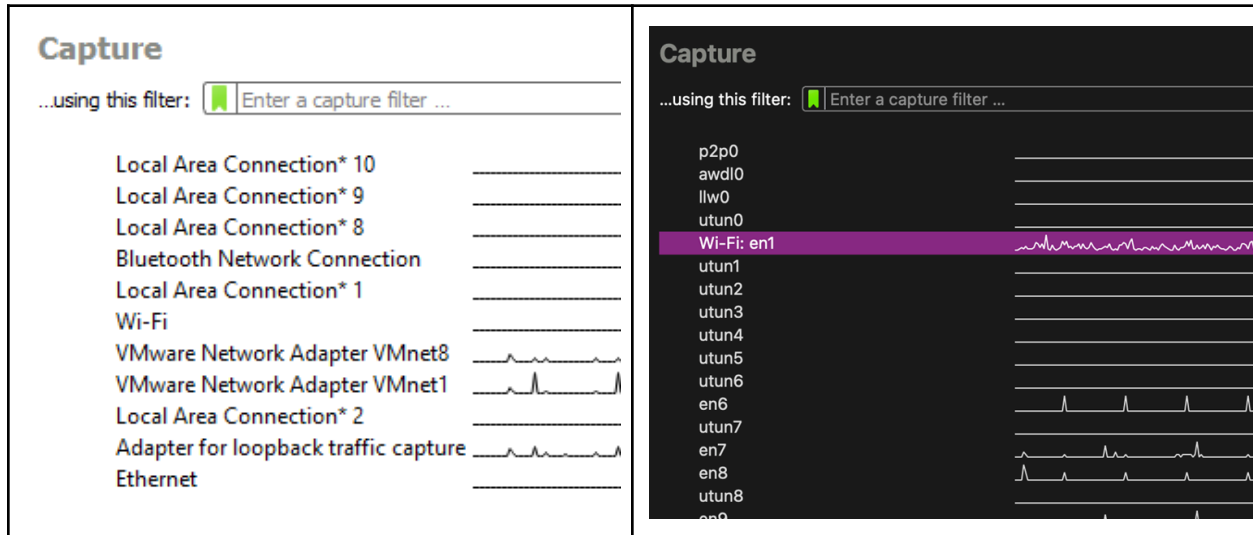
Sometimes, you can run into some permission headaches getting wireshark access to your network traffic. The modern installers are pretty good at getting all the permissions it needs, but if you have issues, Google is the best place to go.

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>

TASK: 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



TASK: Explain in English 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.

- Include a screenshot of the interfaces that Wireshark lists.
- 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.

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

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

- A screenshot makes a lot of sense here.
- **Note:** some students have found HTTP packets were disabled by default. They had to go to Analyze->Enabled Protocols and then enable HTTP.

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

- A screenshot makes a lot of sense here.
- **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

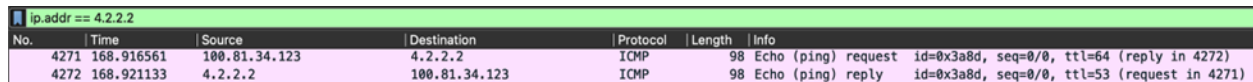
1. Open a terminal window (this works in Command Prompt on Windows too) and run:

```
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:



The screenshot shows a Wireshark packet capture window with the filter 'ip.addr == 4.2.2.2'. The packet list pane displays two ICMP packets:

No.	Time	Source	Destination	Protocol	Length	Info
4271	168.916561	100.81.34.123	4.2.2.2	ICMP	98	Echo (ping) request id=0x3a8d, seq=0/0, ttl=64 (reply in 4272)
4272	168.921133	4.2.2.2	100.81.34.123	ICMP	98	Echo (ping) reply id=0x3a8d, seq=0/0, ttl=53 (request in 4271)

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

TASK: Understand the Ping traffic.

- 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. A screenshot from Wireshark would be useful here as evidence.

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.

TASK: Document 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?

TASK: Document your investigation.

- What does your unknown packet look like?
- How did you determine what the traffic corresponded to?
- Did you notice anything interesting about the traffic?