Wireless & The IoT

Lab 1: Playing with your local network

Introduction

The purpose of today's lab is to solidify your background in the 'nuts & bolts' of Internet technologies, and to 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 `#NotASecurityProfessor`.

Tools & Logistics

We will use Wireshark, available here: https://www.wireshark.org/

Sometimes, you can run into some permission headaches getting wireshark access to your network traffic. The modern installer is pretty good at getting all the permissions it needs, but if you have issues, Google is going to be better bet for debugging than me. It is not a good idea to run Wireshark as root — it'll get all the packets, sure, but that's really opening yourself up for trouble `#OkayIDoListenToTheSecurityProfessorsSometimes`.

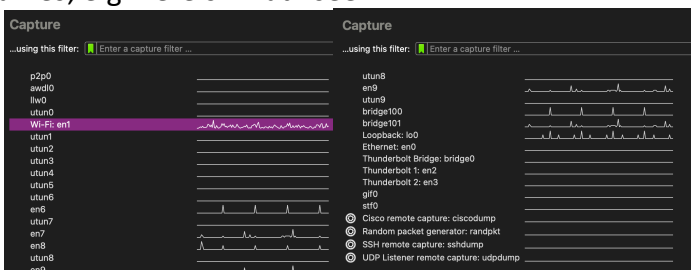
What to submit?

Please use this document as a template, add your responses directly, and export it as a PDF to Gradescope. Keep each response to its own page so stuff lines up neatly on Gradescope.

Reminder: This is **NOT** a super formal lab write-up or any of that. Please do not spend a ton of time making things overly pretty, etc. "Prove to me that you did this lab."

Q1: Interfaces

When you start Wireshark, you have to tell it *where* to capture packets from, which is a helpful list of cryptic, short names; e.g. here's what I see:



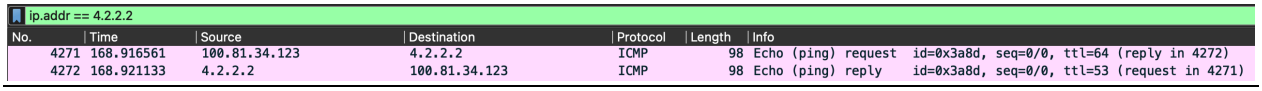
[I have active VMs on this machine, can you tell?]

Explain in English what physical (or digital)-world thing each of the interfaces on your machine corresponds to (e.g. "en1 is my WiFi card"); group interfaces as appropriate.

Don't spend more than ~5 minutes on this. For "don't know's, ask in chat, Google once, then just write "don't know" here if you still can't find it

Q2: Spy on some simple chat traffic

At the beginning of lab, I showed an example of sending ping traffic and finding the traffic I sent in the Wireshark stream.



Be careful sniffing traffic. It can be illegal to monitor communications you were not supposed to have access to.

Work with a partner. Use the netcat utility. One person will "listen" (`nc -l [PORT]`) and the other will connect (`nc [IP] [PORT]`).

Can you see your chat traffic in Wireshark? How? Give an example.

Who else could read your chat traffic?

Q3: Find your own application traffic

Can you find packets that belong to the Zoom meeting for the lab you are currently attending?

Give an example of a Zoom packet:

What method did you use to figure out how to filter for Zoom traffic?

Anything interesting you learned about Zoom from watching its traffic?

Q4: Sleuth application traffic

Can you figure out what's responsible for sending traffic you don't immediately recognize? Odds favor there's a *lot* of traffic flying by. Pick a packet that looks interesting / unfamiliar.

What application on your machine sent or received this packet? How did you figure it out?

Did you know that application was sending data prior to now? ☺

Filter for this application's traffic and watch it for a bit, how would you describe its traffic pattern?

