Basic Wireshark Intro

Debug Project 1 by capturing and analyzing packets

Administrivia

- Quiz
- HW2 due on Monday
  - Needs materials discussed in the lecture
- Project 1 due on Monday
  - Today: a short section followed by additional office hours

Wireshark

Download: https://www.wireshark.org/download.html

- Also available in most Linux package managers

User's Guide: https://www.wireshark.org/docs/wsug_html_chunked/

What is Wireshark

It's a tool that captures and analyzes packets sent over the network

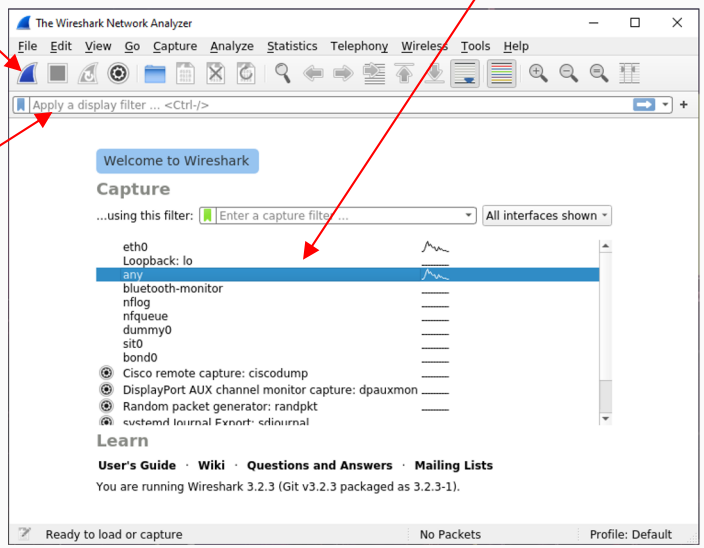
- Very commonly used in Network Forensics
- Captures all packets through a network interface (ethernet, WiFi)
  - Can capture packets on all network interfaces at the same time
- Analyzes packets and decodes raw data if the protocol is recognized
- Filters packets based on user's input

Wireshark Interface

Start Packet Capture

Interface Selection

Display filter for captured packets



Debugging P1 with Wireshark

Lots of packets are being sent while your computer is connected to a network.

- Apply a filter to show only packets to/from attu's IP address
  - How to find the IP address of attu?
    - Run `ifconfig` on attu (through SSH)
    - `nslookup attu2.cs.washington.edu` (from any computer)
    - traceroute will print out the IP address as well
  - ip.addr == 128.208.1.138
- What if you are ssh'd into attu?
  - You will then be capturing packets from SSH as well
- Narrow down further by filtering on the port number
  - udp.port == 12235 (or whatever port number you want to capture on)
  - tcp.port == portNumber
- Can apply boolean logic to combine filters: ==, &&, ||, !
  - ip.addr == 128.208.1.138 && udp.port == 12235
    - Will only show packets to/from attu2 on udp port 12235

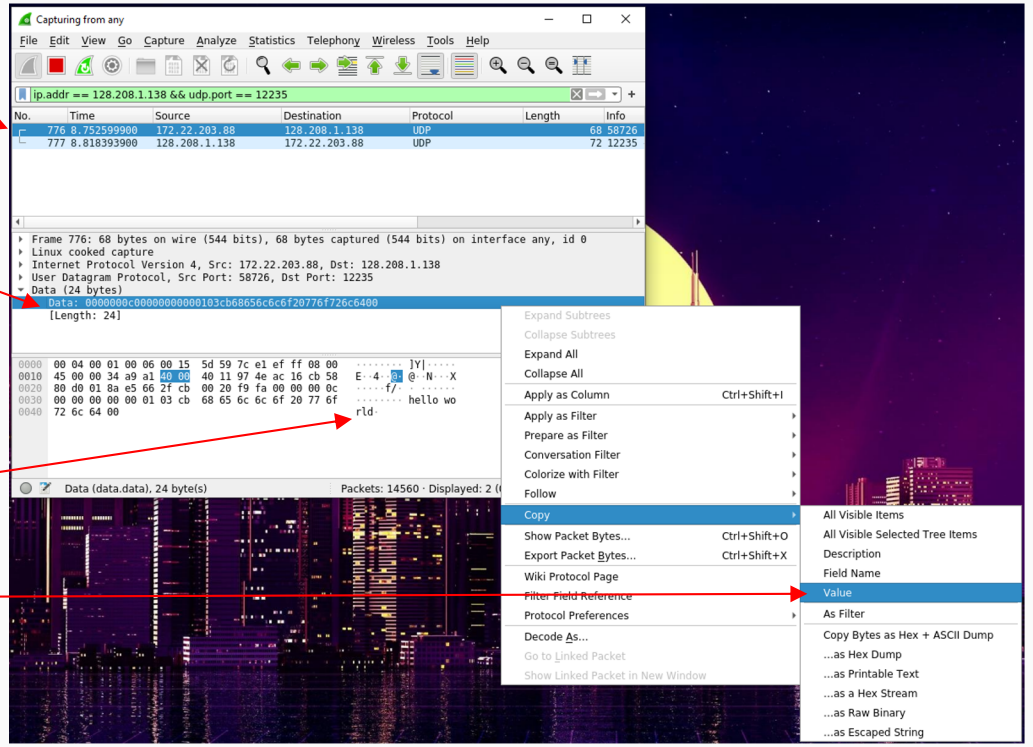
Wireshark Captured Packets Interface

Captured packets

Hexadecimal data contained in the UDP packet

ASCII Decoding of data

Copy data value as hexadecimal string



Debugging using Hex Dumps

In the case of project 1, Wireshark will only recognize/analyze the packet to the UDP or TCP headers

The data structures in p1 aren't recognized by Wireshark

- You will only be able to view the data you sent in hexadecimal or binary format
  - It will attempt to decode ASCII data - so you should see 'hello world' at the end of the first packet
- Viewing the integer values of data will require manually decoding/converting from bytes
- Copy the hexadecimal string of data from wireshark
- Python console can be handy for decoding - or use any other tool you like
  - pbytes = bytes.fromhex('0000000c00000000000103cb68656c6c6f20776f726c6400')
    - Be mindful of endianness - wireshark displays data in Big Endian
    - Make sure the endianness of what you copied matches what is displayed in Wireshark
  - You can now take slices from pbytes and convert them to the appropriate types
    - header_payload_len = int.from_bytes(pbytes[0:4], byteorder='big')
    - header_student_id = int.from_bytes(pbytes[10:12], byteorder='big')

To be continued...

Working with the hexadecimal data can be very tedious.

This isn't a common use-case for Wireshark, but can be useful if all other debugging fails.

We will run through a complete demo of Wireshark in section later in the quarter (or not?)

The rest of today's section will be additional office hours for Project 1.