Computer Networks

Traceroute and wireshark
Spring 2022
With Monty, Edan, Jason, and Mark!

Administrivia

- Project 1 is out! Due April 18th at 11:00pm
    - Can be done in groups of 2-3
    - Can be done in any language (recommend Java / Python)
        - Future labs will be in Python
        - Intent is to allow you to become familiar with some languages Socket API!
- Homework 1 is out! Due April 14th at 11:00pm
    - That's tonight!
    - Read Chapter 1, specifically section 1.5 and beyond
- Homework 2 will be out soon, Due April 25th at 11:00pm

Download now! - Wireshark

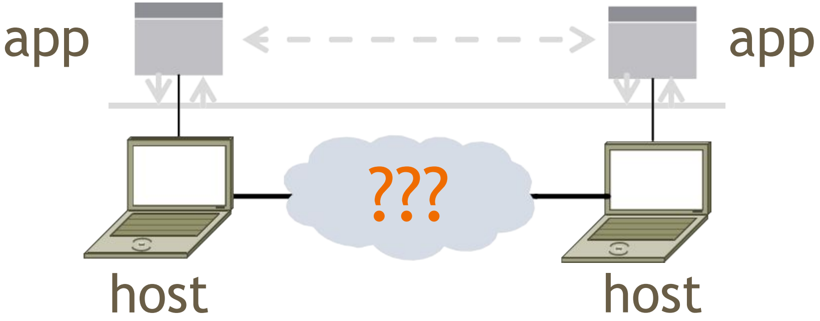
Download: https://www.wireshark.org/download.html

- Also available in most Linux package managers

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

Traceroute

- Apps talk to other apps but have no idea what is inside the network
  - This is good! But you may be curious … what route are packets possibly using?
- We can take a peek into the network with Traceroute!



Traceroute

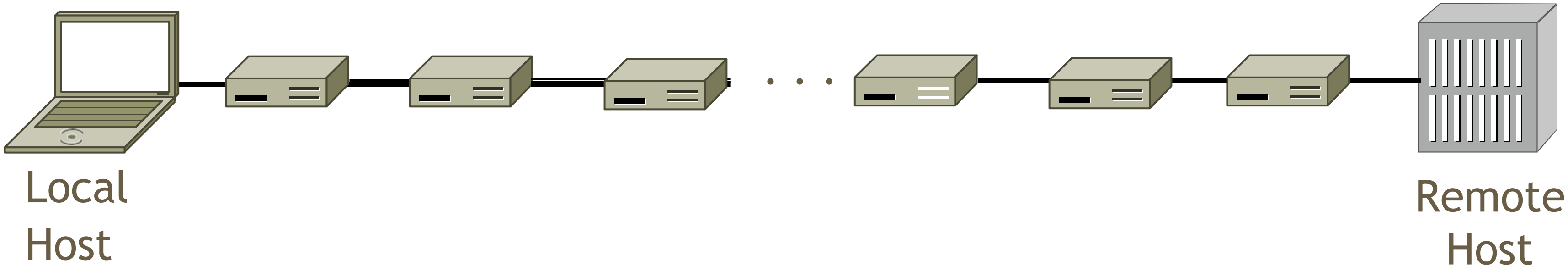
- Traceroute is a widely used command-line tool to let hosts peek inside the network
    - Implemented on all OSes (tracert on Windows)
    - Developed by Van Jacobson ~ 1987
    - Uses a network-network interface (IP) in ways we will explain later

Van Jacobson

Credit: Wikipedia

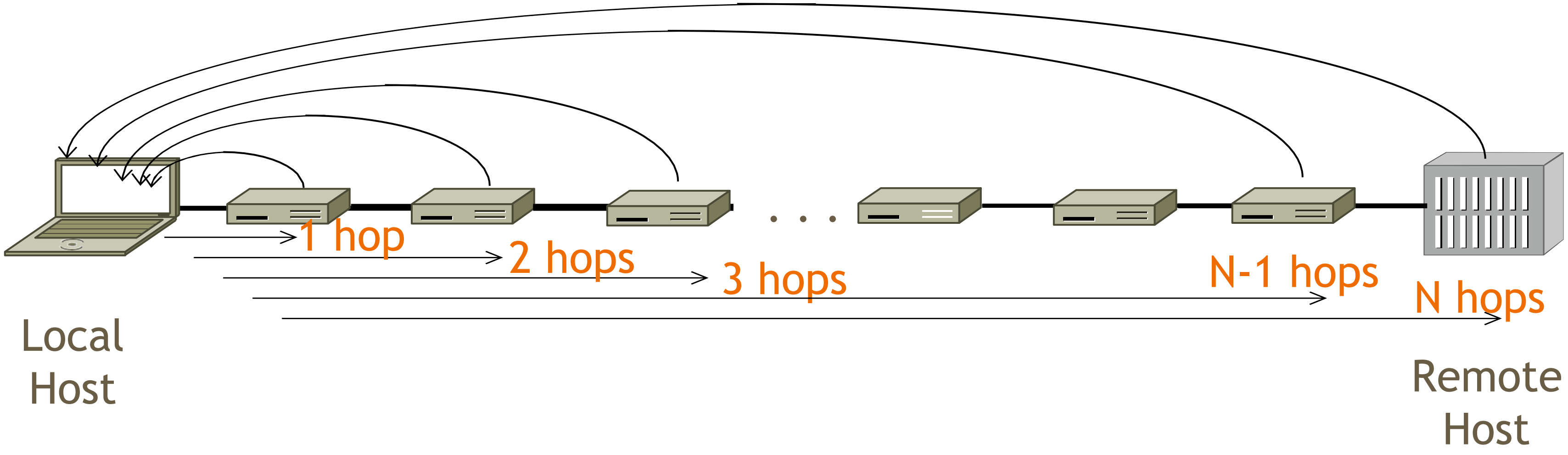
Traceroute

- We want to find network path from our system to a given remote host
- Core mechanism: Time-To-Live(TTL)
  - Time-To-Live: keeps packets from swirling in the network forever, usually measured in "hops"



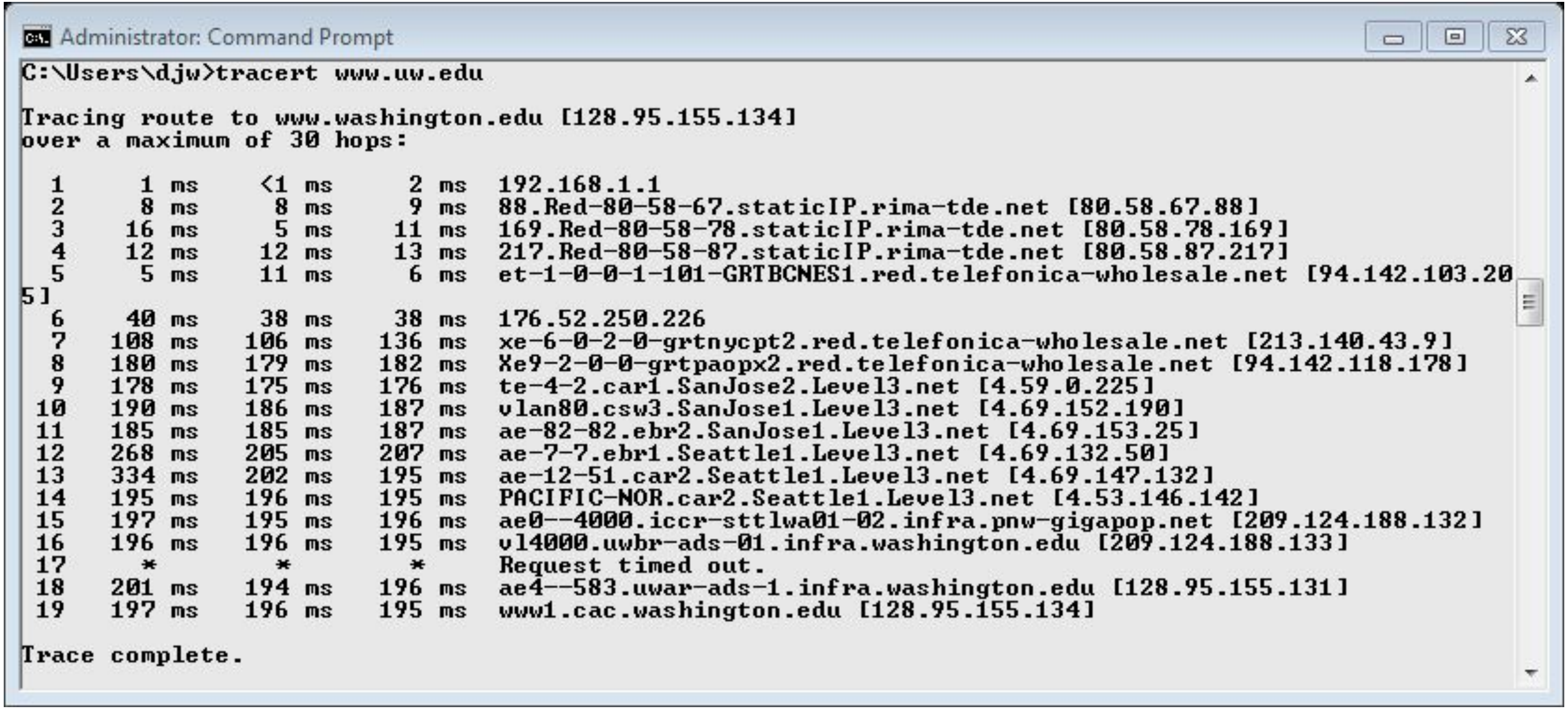
Traceroute

- We want to find network path from our system to a given remote host
- Core mechanism: Time-To-Live(TTL)
  - Time-To-Live: keeps packets from swirling in the network forever, usually measured in “hops”
  - Some information about a packets “death” is usually sent back to the local host



Traceroute Demo

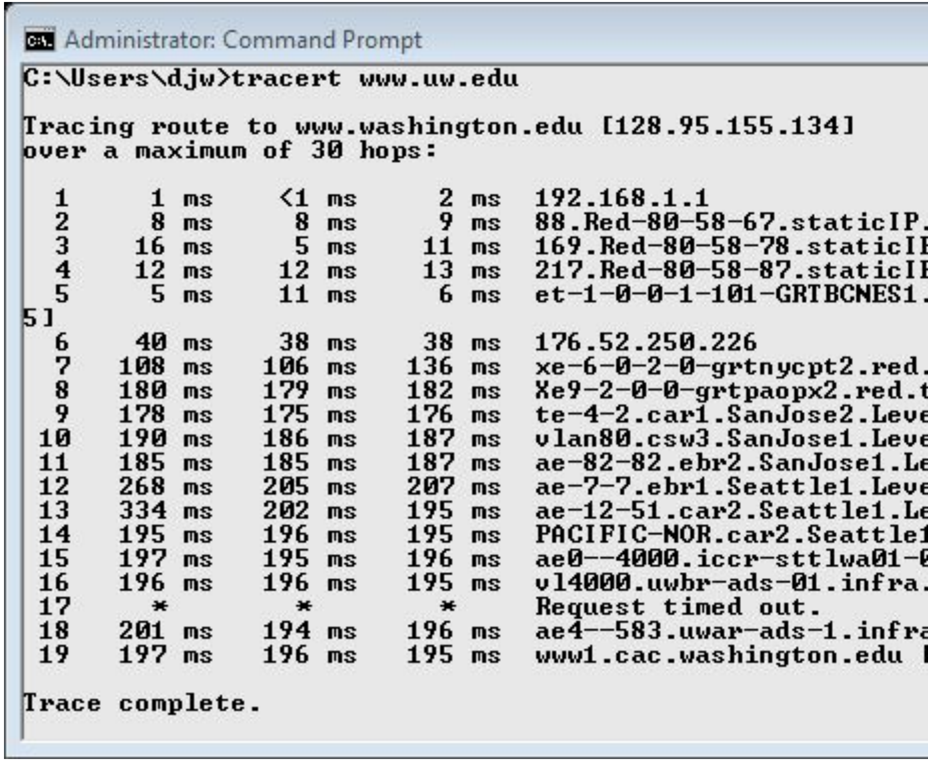
Using Traceroute - Exercise (groups of 2-3)



```
C:\Users\djw>tracert www.uw.edu

Tracing route to www.washington.edu [128.95.155.134]
over a maximum of 30 hops:

  1     1 ms    <1 ms     2 ms  192.168.1.1
  2     8 ms     8 ms     9 ms  88.Red-80-58-67.staticIP.rima-tde.net [80.58.67.88]
  3    16 ms     5 ms    11 ms  169.Red-80-58-78.staticIP.rima-tde.net [80.58.78.169]
  4    12 ms    12 ms    13 ms  217.Red-80-58-87.staticIP.rima-tde.net [80.58.87.217]
  5     5 ms    11 ms     6 ms  et-1-0-0-1-101-GRTBCNES1.red.telefonica-wholesale.net [94.142.103.20
5]
  6    40 ms    38 ms    38 ms  176.52.250.226
  7   108 ms   106 ms   136 ms  xe-6-0-2-0-grtnycpt2.red.telefonica-wholesale.net [213.140.43.9]
  8   180 ms   179 ms   182 ms  Xe9-2-0-0-grtpaopx2.red.telefonica-wholesale.net [94.142.118.178]
  9   178 ms   175 ms   176 ms  te-4-2.car1.SanJose2.Level3.net [4.59.0.225]
 10   190 ms   186 ms   187 ms  vlan80.csw3.SanJose1.Level3.net [4.69.152.190]
 11   185 ms   185 ms   187 ms  ae-82-82.ebr2.SanJose1.Level3.net [4.69.153.25]
 12   268 ms   205 ms   207 ms  ae-7-7.ebr1.Seattle1.Level3.net [4.69.132.50]
 13   334 ms   202 ms   195 ms  ae-12-51.car2.Seattle1.Level3.net [4.69.147.132]
 14   195 ms   196 ms   195 ms  PACIFIC-NOR.car2.Seattle1.Level3.net [4.53.146.142]
 15   197 ms   195 ms   196 ms  ae0--4000.iccr-sttlwa01-02.infra.pnw-gigapop.net [209.124.188.132]
 16   196 ms   196 ms   195 ms  v14000.uwbr-ads-01.infra.washington.edu [209.124.188.133]
 17     *        *        *     Request timed out.
 18   201 ms   194 ms   196 ms  ae4--583.uwar-ads-1.infra.washington.edu [128.95.155.131]
 19   197 ms   196 ms   195 ms  www1.cac.washington.edu [128.95.155.134]

Trace complete.
```

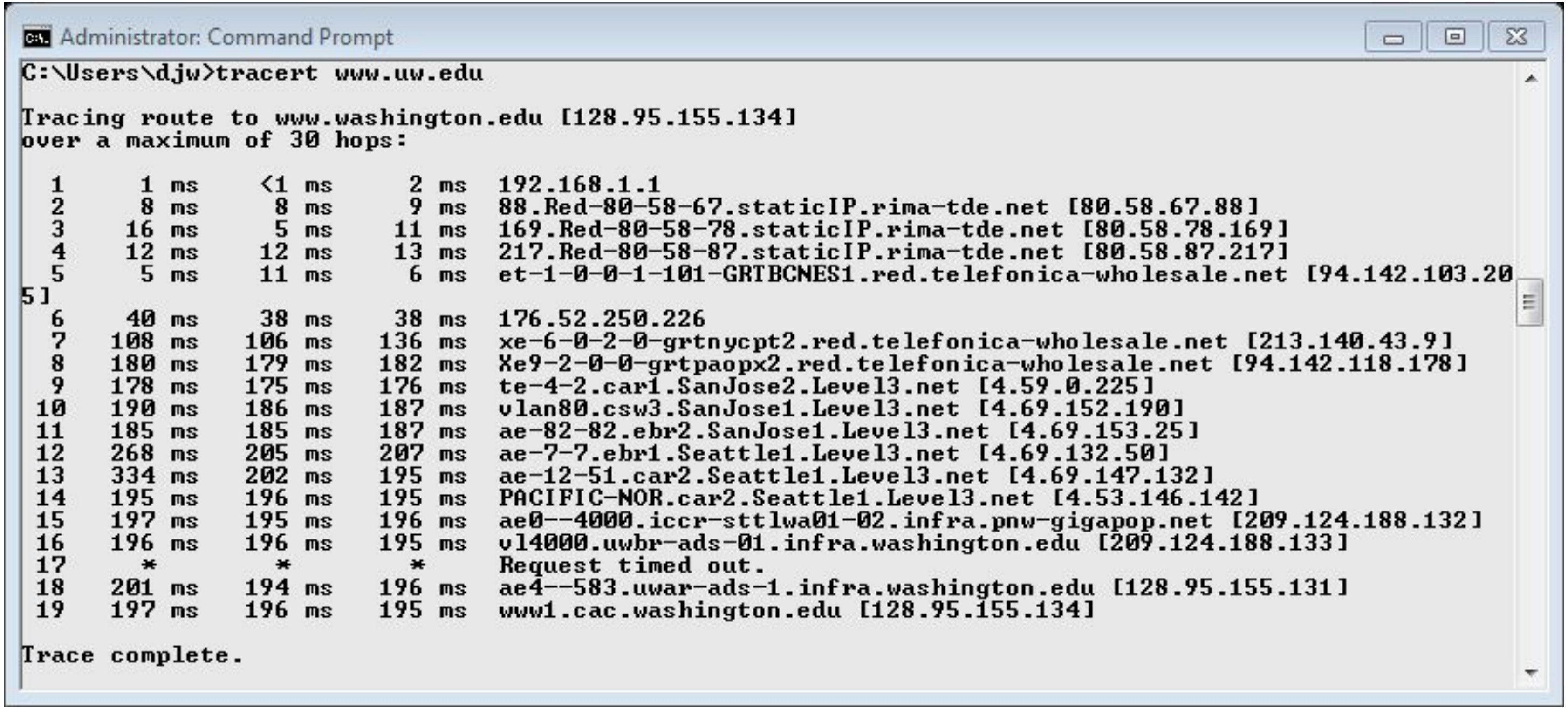
Using Traceroute - Exercise (groups of 2-3)

- What do the indices 1-19 represent?
- Why are there 3 times on each row, and why are they sometimes so different?
- Why are the times not strictly increasing for > number of hops?
- Why might the request have timed out on 17?
- What happens when TTL = 0? Are we out of luck?
- What is the utility of traceroute beyond helping us see the path that a packet takes?



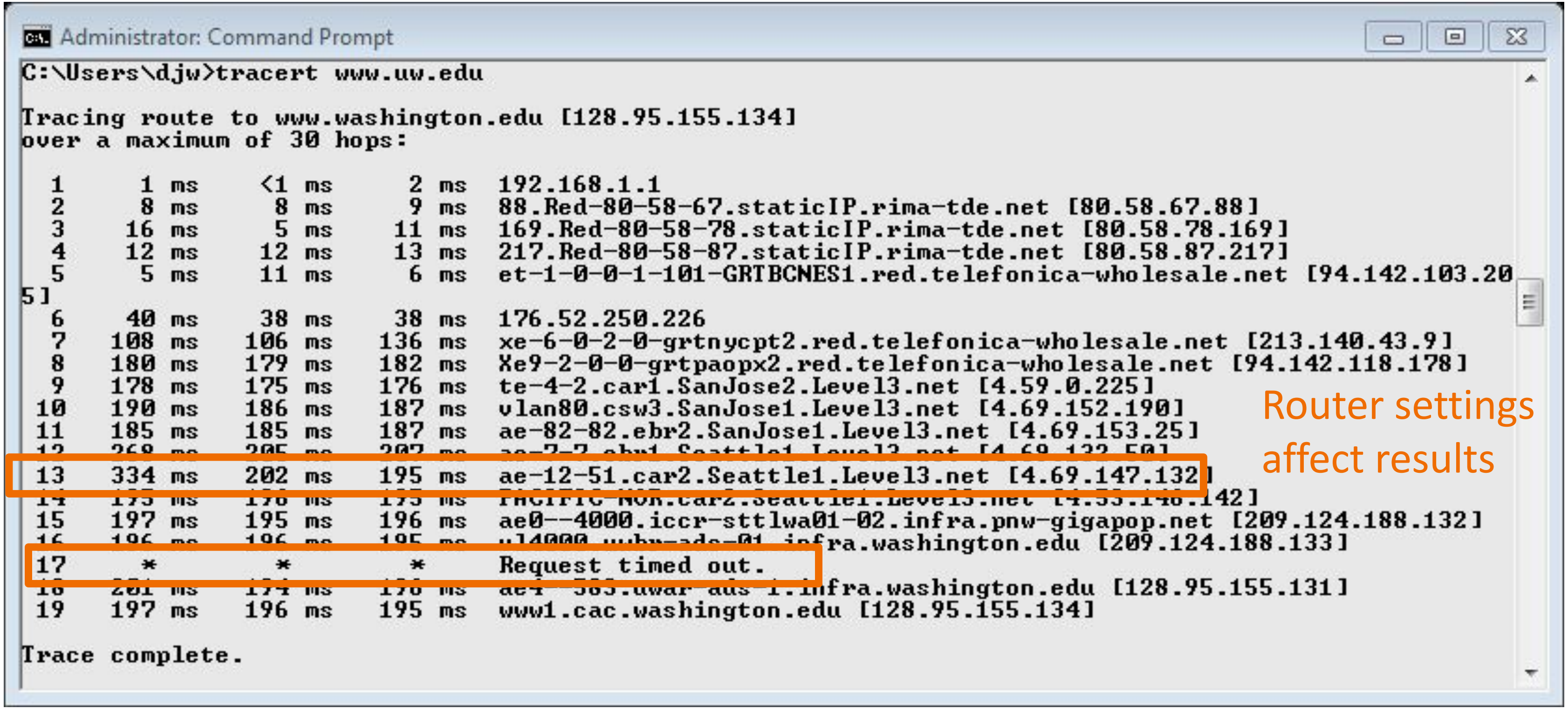
```
C:\Users\djw>tracert www.uw.edu

Tracing route to www.washington.edu [128.95.155.134]
over a maximum of 30 hops:

  1     1 ms    <1 ms     2 ms  192.168.1.1
  2     8 ms     8 ms     9 ms  88.Red-80-58-67.staticIP.
  3    16 ms     5 ms    11 ms  169.Red-80-58-78.staticIP
  4    12 ms    12 ms    13 ms  217.Red-80-58-87.staticIP
  5     5 ms    11 ms     6 ms  et-1-0-0-1-101-GRTBCNES1.
5]
  6    40 ms    38 ms    38 ms  176.52.250.226
  7   108 ms   106 ms   136 ms  xe-6-0-2-0-grtnycpt2.red.
  8   180 ms   179 ms   182 ms  Xe9-2-0-0-grtpaopx2.red.t
  9   178 ms   175 ms   176 ms  te-4-2.car1.SanJose2.Leve
 10   190 ms   186 ms   187 ms  vlan80.csw3.SanJose1.Leve
 11   185 ms   185 ms   187 ms  ae-82-82.ebr2.SanJose1.Le
 12   268 ms   205 ms   207 ms  ae-7-7.ebr1.Seattle1.Leve
 13   334 ms   202 ms   195 ms  ae-12-51.car2.Seattle1.Le
 14   195 ms   196 ms   195 ms  PACIFIC-NOR.car2.Seattle1
 15   197 ms   195 ms   196 ms  ae0--4000.iccr-sttlwa01-0
 16   196 ms   196 ms   195 ms  vl4000.uwbr-ads-01.infra.
 17     *        *        *     Request timed out.
 18   201 ms   194 ms   196 ms  ae4--583.uwar-ads-1.infra
 19   197 ms   196 ms   195 ms  www1.cac.washington.edu [

Trace complete.
```

Traceroute



```
C:\Users\djw>tracert www.uw.edu

Tracing route to www.washington.edu [128.95.155.134]
over a maximum of 30 hops:

  1     1 ms    <1 ms     2 ms  192.168.1.1
  2     8 ms     8 ms     9 ms  88.Red-80-58-67.staticIP.rima-tde.net [80.58.67.88]
  3    16 ms     5 ms    11 ms  169.Red-80-58-78.staticIP.rima-tde.net [80.58.78.169]
  4    12 ms    12 ms    13 ms  217.Red-80-58-87.staticIP.rima-tde.net [80.58.87.217]
  5     5 ms    11 ms     6 ms  et-1-0-0-1-101-GRTBCNES1.red.telefonica-wholesale.net [94.142.103.20
5]
  6    40 ms    38 ms    38 ms  176.52.250.226
  7   108 ms   106 ms   136 ms  xe-6-0-2-0-grtnycpt2.red.telefonica-wholesale.net [213.140.43.9]
  8   180 ms   179 ms   182 ms  Xe9-2-0-0-grtpaopx2.red.telefonica-wholesale.net [94.142.118.178]
  9   178 ms   175 ms   176 ms  te-4-2.car1.SanJose2.Level3.net [4.59.0.225]
 10   190 ms   186 ms   187 ms  vlan80.csw3.SanJose1.Level3.net [4.69.152.190]
 11   185 ms   185 ms   187 ms  ae-82-82.ebr2.SanJose1.Level3.net [4.69.153.25]
 12   268 ms   205 ms   207 ms  ae-7-7.ebr1.Seattle1.Level3.net [4.69.132.50]
 13   334 ms   202 ms   195 ms  ae-12-51.car2.Seattle1.Level3.net [4.69.147.132]
 14   195 ms   196 ms   195 ms  PACIFIC-NOR.car2.Seattle1.Level3.net [4.53.146.142]
 15   197 ms   195 ms   196 ms  ae0--4000.iccr-sttlwa01-02.infra.pnw-gigapop.net [209.124.188.132]
 16   196 ms   196 ms   195 ms  vl4000.uwbr-ads-01.infra.washington.edu [209.124.188.133]
 17     *        *        *     Request timed out.
 18   201 ms   194 ms   196 ms  ae4--583.uwar-ads-1.infra.washington.edu [128.95.155.131]
 19   197 ms   196 ms   195 ms  www1.cac.washington.edu [128.95.155.134]

Trace complete.
```

Traceroute



```
C:\Users\djw>tracert www.uw.edu

Tracing route to www.washington.edu [128.95.155.134]
over a maximum of 30 hops:

  1     1 ms    <1 ms     2 ms  192.168.1.1
  2     8 ms     8 ms     9 ms  88.Red-80-58-67.staticIP.rima-tde.net [80.58.67.88]
  3    16 ms     5 ms    11 ms  169.Red-80-58-78.staticIP.rima-tde.net [80.58.78.169]
  4    12 ms    12 ms    13 ms  217.Red-80-58-87.staticIP.rima-tde.net [80.58.87.217]
  5     5 ms    11 ms     6 ms  et-1-0-0-1-101-GRTBCNES1.red.telefonica-wholesale.net [94.142.103.20
5]
  6    40 ms    38 ms    38 ms  176.52.250.226
  7   108 ms   106 ms   136 ms  xe-6-0-2-0-grtnycpt2.red.telefonica-wholesale.net [213.140.43.9]
  8   180 ms   179 ms   182 ms  Xe9-2-0-0-grtpaopx2.red.telefonica-wholesale.net [94.142.118.178]
  9   178 ms   175 ms   176 ms  te-4-2.car1.SanJose2.Level3.net [4.59.0.225]
 10   190 ms   186 ms   187 ms  vlan80.csw3.SanJose1.Level3.net [4.69.152.190]
 11   185 ms   185 ms   187 ms  ae-82-82.ebr2.SanJose1.Level3.net [4.69.153.25]
 12   268 ms   205 ms   207 ms  ae-7-7.ebr1.Seattle1.Level3.net [4.69.132.50]
 13   334 ms   202 ms   195 ms  ae-12-51.car2.Seattle1.Level3.net [4.69.147.132]
 14   195 ms   196 ms   195 ms  PACIFIC-NOR.car2.Seattle1.Level3.net [4.53.146.142]
 15   197 ms   195 ms   196 ms  ae0--4000.iccr-sttlwa01-02.infra.pnw-gigapop.net [209.124.188.132]
 16   196 ms   196 ms   195 ms  v14000.uwbr-ads-01.infra.washington.edu [209.124.188.133]
 17     *        *        *     Request timed out.
 18   201 ms   194 ms   196 ms  ae4--583.uwar-ads-1.infra.washington.edu [128.95.155.131]
 19   197 ms   196 ms   195 ms  www1.cac.washington.edu [128.95.155.134]

Trace complete.
```

Router settings affect results

Wireshark

What is Wireshark

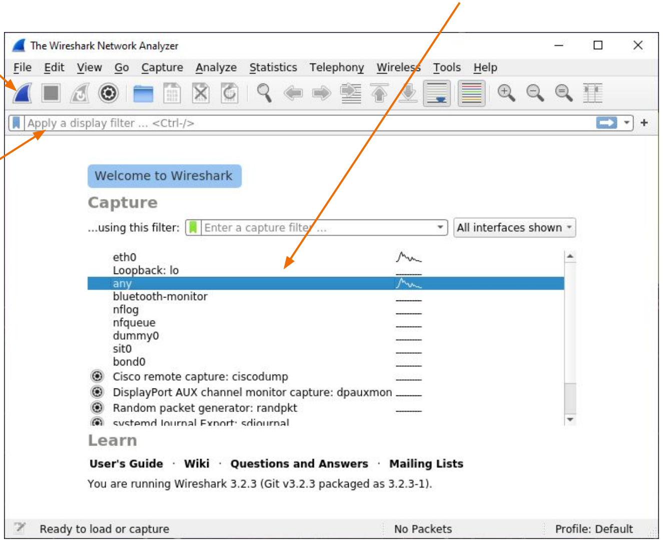
- It’s a tool that captures and analyzes packets sent over the network!
  - Very commonly used in Network Forensics
  - Captures all packets through a network interface (ethernet, WiFi)
  - Analyzes packets and decodes raw data if the protocol is recognized
  - Filters packets based on user’s input

Wireshark Interface

Start Packet Capture

Interface Selection

Display filter for captured packets


The Wireshark Network Analyzer
File
Edit
View
Go
Capture
Analyze
Statistics
Telephony
Wireless
Tools
Help
Apply a display filter ... <Ctrl-/>
Welcome to Wireshark
Capture
Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists

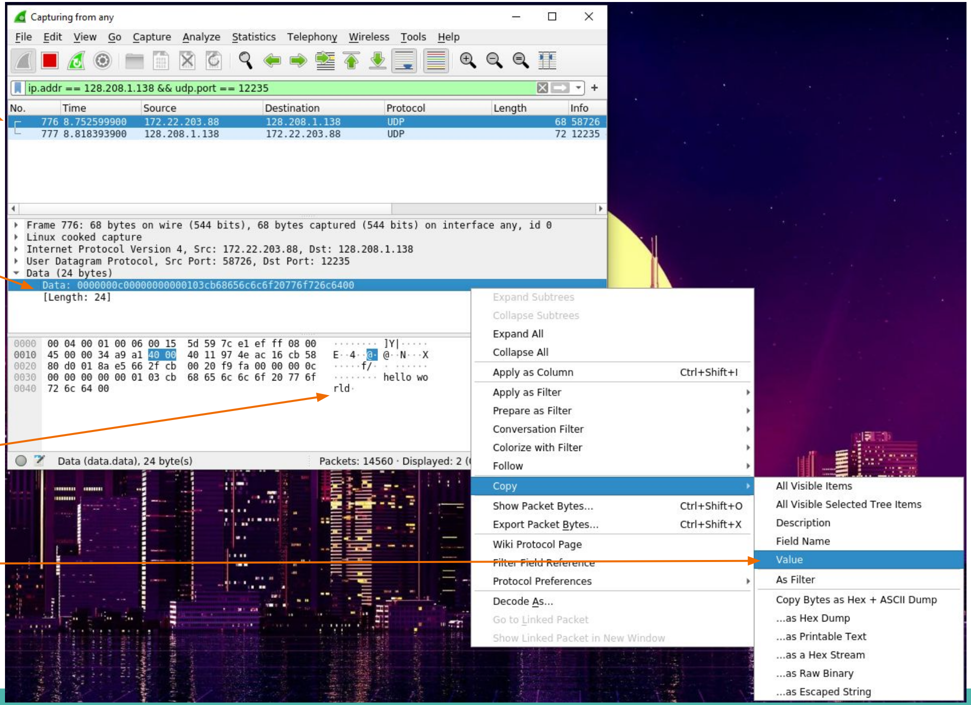
...using this filter: Enter a capture filter ... All interfaces shown

eth0
Loopback: lo
any
bluetooth-monitor
nflog
nfqueue
dummy0
sit0
bond0
Cisco remote capture: ciscodump
DisplayPort AUX channel monitor capture: dpauxmon
Random packet generator: randpkt
systemd Journal Export: sdjournal

You are running Wireshark 3.2.3 (Git v3.2.3 packaged as 3.2.3-1).

Ready to load or capture   No Packets   Profile: Default

Wireshark Captured Packets Interface



Captured packets

Hexadecimal data contained in the UDP packet

ASCII Decoding of data

Copy data value as hexadecimal string

Capturing from any

ip.addr == 128.208.1.138 && udp.port == 12235

```
No.   Time           Source          Destination     Protocol   Length  Info
  776 8.752599900    172.22.203.88   128.208.1.138   UDP            68  58726
  777 8.818393900    128.208.1.138   172.22.203.88   UDP            72  12235
```

```
▸ Frame 776: 68 bytes on wire (544 bits), 68 bytes captured (544 bits) on interface any, id 0
▸ Linux cooked capture
▸ Internet Protocol Version 4, Src: 172.22.203.88, Dst: 128.208.1.138
▸ User Datagram Protocol, Src Port: 58726, Dst Port: 12235
▾ Data (24 bytes)
    Data: 0000000c00000000000103cb68656c6c6f20776f726c6400
    [Length: 24]
```

```
0000  00 04 00 01 00 06 00 15  5d 59 7c e1 ef ff 08 00   ········ ]Y|·····
0010  45 00 00 34 a9 a1 40 00  40 11 97 4e ac 16 cb 58   E··4··@· @··N···X
0020  80 d0 01 8a e5 66 2f cb  00 20 f9 fa 00 00 00 0c   ·····f/· · ······
0030  00 00 00 00 00 01 03 cb  68 65 6c 6c 6f 20 77 6f   ········ hello wo
0040  72 6c 64 00                                        rld·
```

Data (data.data), 24 byte(s)

Packets: 14560 · Displayed: 2 (

Wireshark Demo

- Close as many other browser tabs as possible
  - This will complicate what you see on the interface
- Start capturing packets on Wireshark (What interface should you listen on?)
- Open youtube.com (or any other website!) and start streaming a video or downloading a file
- Stop capturing packets (if you let it go for too long, you will be trying to store loads of data!)
- Can you find the Youtube stream in Wireshark?
  - Is this the right interface? What do the interfaces represent?
  - What is the easiest way to isolate web traffic?
    - Is there a particular protocol or port that’s always allocated to browsing data?

Debugging P1 with Wireshark

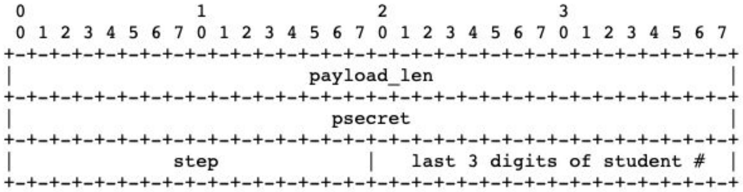
Lots of packets are being sent while your computer is connected to a network.

- *Filtering 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`
- *Filtering on the* ***port number***
  - `udp.port == 12235`
  - `tcp.port == portNumber`
- *Applying 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

Debugging using Hex Dumps

The data structures in p1 aren’t recognized by Wireshark

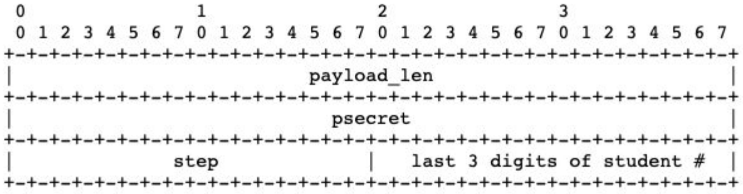
- You will only be able to view the data you sent in hexadecimal or binary format
  - It will attempt to decode ASCII data - so you should see ‘hello world’ at the end of the first packet
- Viewing the integer values of data will require manually decoding/converting from bytes



```
 0               1               2               3
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                          payload_len                          |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                            psecret                            |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              step             |   last 3 digits of student #  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

More pcap to analyze - CIC data

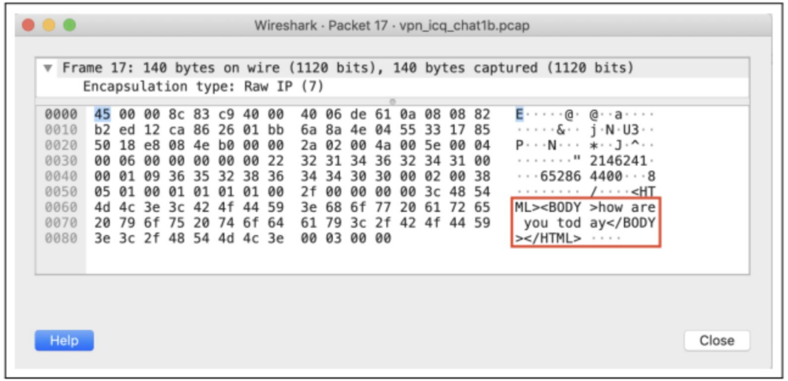
- Copy the hexadecimal string of data from wireshark
- Python console can be handy for decoding - or use any other tool you like
  - `pbytes = bytes.fromhex(‘0000000c00000000000103cb68656c6c6f20776f726c6400’)`
    - Be mindful of endianness - wireshark displays data in Big Endian
  - You can now take 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’)`



```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                           payload_len                         |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                             psecret                           |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|              step             |   last 3 digits of student #  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

More pcap to analyze - CIC data

- Canadian Institute of Cybersecurity : [VPN-nonVPN dataset (ISCXVPN2016)]



```
▼ Frame 17: 140 bytes on wire (1120 bits), 140 bytes captured (1120 bits)
     Encapsulation type: Raw IP (7)

0000  45 00 00 8c 83 c9 40 00  40 06 de 61 0a 08 08 82   E·····@· @··a····
0010  b2 ed 12 ca 86 26 01 bb  6a 8a 4e 04 55 33 17 85   ·····&·· j·N·U3··
0020  50 18 e8 08 4e b0 00 00  2a 02 00 4a 00 5e 00 04   P···N··· *··J·^··
0030  00 06 00 00 00 00 00 22  32 31 34 36 32 34 31 00   ·······" 2146241·
0040  00 01 09 36 35 32 38 36  34 34 30 30 00 02 00 38   ···65286 4400···8
0050  05 01 00 01 01 01 01 00  2f 00 00 00 00 3c 48 54   ········ /····<HT
0060  4d 4c 3e 3c 42 4f 44 59  3e 68 6f 77 20 61 72 65   ML><BODY >how are
0070  20 79 6f 75 20 74 6f 64  61 79 3c 2f 42 4f 44 59    you tod ay</BODY
0080  3e 3c 2f 48 54 4d 4c 3e  00 03 00 00               ></HTML> ····
```

Fig. 1. The unencrypted payload of the 17th packet in the ICQ chat VPN capture of the ISCXVPN2016 dataset. The IP address of this capture also matches a known ICQ server, and other connections can be distinguished in the capture.

Thanks for coming!