Wireshark 101

Qiao Zhang
CSE 461 15sp Section #1

Slides adapted from
Ravi Bhoraskar

What is Wireshark?

- Wireshark is a **network packet analyzer**
  - uses libpcap to capture packets
  - logs all packets seen by NIC
  - can display packet captured in real-time
  - can save packet trace as a file (*.pcap)

- Wireshark understands and decodes protocols
  - knows how packets are encapsulated
  - displays header in human-readable format
  - follows protocol sequence e.g. track a TCP stream

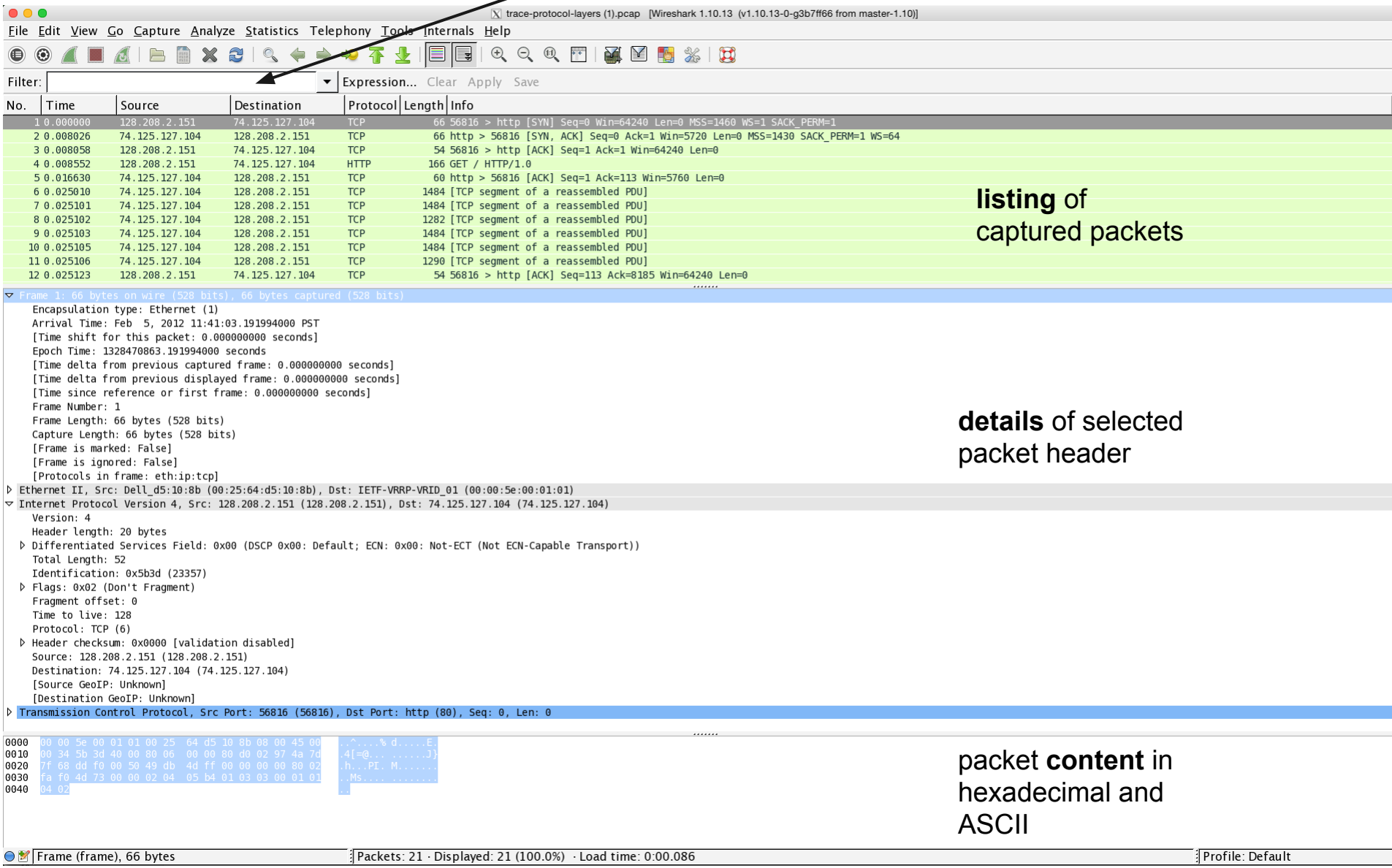
Why use Wireshark?

- Protocol analysis
  - verify correctness
  - analyze performance
  - better understanding of existing protocols
  - optimization and debugging of new protocols

- Works on Linux, OS X and Windows
  - works for both ethernet/wireless medium

- Has a GUI! Easier to use than tcpdump

display **filter** specification



listing of captured packets

details of selected packet header

packet **content** in hexadecimal and ASCII

Network Interfaces

- Need to specify one for Wireshark to snoop on

- Show network interfaces:
  - On a linux box: “ifconfig”
  - Windows: “ipconfig /a”
  - Wireshark menu: Capture->Interfaces

- Must select loopback interface (lo0) to see packets from your own machine to itself e.g. “ping localhost”

Demo 1 – Basic Run

- Run wireshark on en1

Filters

- We are often not interested in all packets flowing through the network
- Use filters to capture only packets of interest to us
- Two kind of filters
  - Capture Filter: Filtered while capturing. Like TCPDump
  - Display Filter: More detailed filtering. Allows to compare values in packets. Not real time

Demo 2

- Capture only udp packets
  - Capture filter = “udp”
- Capture only tcp packets
  - Capture filter = “tcp”

Demo 2 (contd.)

- Capture only UDP packets with destination port 53 (DNS requests)
  - “udp dst port 53”
- Capture only UDP packets with source port 53 (DNS replies)
  - “udp src port 53”
- Capture only UDP packets with source or destination port 53 (DNS requests and replies)
  - “udp port 53”

Demo 2 (contd.)

- Capture only packets destined to www.cs.washington.edu
  - “dst host www.cs.washington.edu”

- Capture both DNS packets and TCP packets to/from www.cs.washington.edu
  - “(tcp and host www.cs.washington.edu) or udp port 53”

Display Filters

- Different Syntax
  - frame.len > 10
  - ip.addr == 129.111.0.0/16 [CIDR masking]
- More expressive
  - eth.src[1–2] == 00:83 [Check only bytes 1 and 2]
- Go crazy with logical expressions
  - tcp.dst[0:3] == 0.6.29 xor udp.src[1] == 42

- Cheat sheet

http://packetlife.net/media/library/13/Wireshark_Display_Filters.pdf

How to write filters

- Refer cheat sheet slides at the end of this presentation
- Refer the tcpdump man page and wireshark documentation
    - capture filters https://wiki.wireshark.org/CaptureFilters
    - display filters https://wiki.wireshark.org/DisplayFilters

Security/Privacy Issues

- Wireshark allows you to monitor other people's traffic
- WARNING: Do NOT use wireshark to violate privacy or security
- Use filtering to restrict packet analysis to only the traffic associated with your program
  - filter based on port that your application uses

Thank You

Cheat Sheet – Writing Filters (1)

- Specifying the hosts we are interested in
  - “dst host <name/IP>”
  - “src host <name/IP>”
  - “host <name/IP>” (either source or destination is name/IP)
- Specifying the ports we are interested in
  - “dst port <number>”
  - “src port <number>”
  - “port <number>”
- Makes sense only for TCP and UDP packets

Cheat Sheet – Writing Filters (2)

- Specifying ICMP packets
  - “icmp”
- Specifying UDP packets
  - “udp”
- Specifying TCP packets
  - “tcp”

Cheat Sheet – Writing Filters (2)

- Combining filters
  - *and (&&)*
  - *or (||)*
  - *not (!)*
- Example:
  - *“tcp and ! host quasar.cs.berkeley.edu”*
  - All tcp packets which are not from or to host quas