# Capturing & Analyzing Network Traffic: tcpdump/tshark and Wireshark

**EE 122: Intro to Communication Networks**

**Vern Paxson / Jorge Ortiz / Dilip Anthony Joseph**



## Overview

- Examples of network protocols
- Protocol Analysis
  - Verify Correctness
  - Analyze performance
  - Better understanding of existing protocols
  - Optimization and debugging of new protocols
- Tools
  - tcpdump & tshark
  - Wireshark



## Network Protocol Examples

- Defines the rules of exchange between a pair (or more) machines over a communication network
- HTTP (Hypertext Transfer Protocol)
  - Defines how web pages are fetched and sent across a network
- TCP (Transmission Control Protocol)
  - Provides reliable, in-order delivery of a stream of bytes
- Your protocol here



## Protocol Analysis

- Verify correctness
- Debug/detect incorrect behavior
- Analyze performance
- Gain deeper understanding of existing protocols by "seeing" how they behave in actual use



## Analysis Methods

- Instrument the code
  - Difficult task, even for experienced network programmers
  - Tedious and time consuming
- Use available tools
  - tcpdump / tshark
  - Wireshark
  - ipsumdump
- Write your own tool
  - libpcap



## Tools overview

- Tcpdump
  - Unix-based command-line tool used to intercept packets
    - Including filtering to just the packets of interest
  - Reads "live traffic" from interface specified using `-i` option …
  - … or from a previously recorded *trace file* specified using `-r` option
    - You create these when capturing live traffic using `-w` option
- Tshark
  - Tcpdump-like capture program that comes w/ Wireshark
  - Very similar behavior & flags to tcpdump
- Wireshark
  - GUI for displaying tcpdump/tshark packet traces

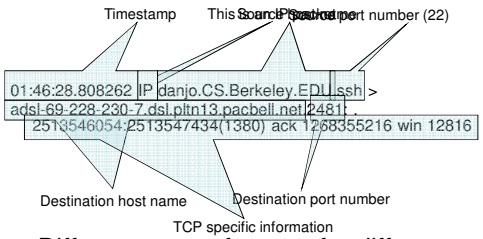
## Tcpdump example

- Ran tcpdump on the machine danjo.cs.berkeley.edu
- First few lines of the output:

```
01:46:28.808262 IP danjo.CS.Berkeley.EDU.ssh > adsl-69-228-230-7.dsl.pltn13.pacbell.net.2481: . 2513546054:2513547434(1380) ack 1268355216 win 12816
01:46:28.808271 IP danjo.CS.Berkeley.EDU.ssh > adsl-69-228-230-7.dsl.pltn13.pacbell.net.2481: P 1380:2128(748) ack 1 win 12816
01:46:28.808276 IP danjo.CS.Berkeley.EDU.ssh > adsl-69-228-230-7.dsl.pltn13.pacbell.net.2481: . 2128:3508(1380) ack 1 win 12816
01:46:28.890021 IP adsl-69-228-230-7.dsl.pltn13.pacbell.net.2481 > danjo.CS.Berkeley.EDU.ssh: P 1:49(48) ack 1380 win 16560
```

7

## What does a line convey?




- Different output formats for different packet types

8

## Similar Output from Tshark

```
1190003744.940437 61.184.241.230 -> 128.32.48.169 SSH Encrypted request packet len=48
1190003744.940916 128.32.48.169 -> 61.184.241.230 SSH Encrypted response packet len=48
1190003744.955764 61.184.241.230 -> 128.32.48.169 TCP 6943 > ssh [ACK] Seq=48 Ack=48 Win=65514 Len=0 TSV=445871583 TSER=632535493
1190003745.035678 61.184.241.230 -> 128.32.48.169 SSH Encrypted request packet len=48
1190003745.036004 128.32.48.169 -> 61.184.241.230 SSH Encrypted response packet len=48
1190003745.050970 61.184.241.230 -> 128.32.48.169 TCP 6943 > ssh [ACK] Seq=96 Ack=96 Win=65514 Len=0 TSV=445871583 TSER=632535502
```

9

## Demo 1 – Basic Run

- Syntax:

  *tcpdump [options] [filter expression]*
- Run the following command on the machine *c199.eecs.berkeley.edu*:

  tcpdump
- Observe the output

10

## Filters

- We are often not interested in all packets flowing through the network
- Use filters to capture only packets of interest to us

11

## Demo 2

1. Capture only udp packets
   - tcpdump “udp”
2. Capture only tcp packets
   - tcpdump “tcp”

12

## Demo 2 (contd.)

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



## Demo 2 (contd.)

1. Capture only packets destined to quasar.cs.berkeley.edu
   - tcpdump “dst host quasar.cs.berkeley.edu”
2. Capture both DNS packets and TCP packets to/from quasar.cs.berkeley.edu
   - tcpdump “(tcp and host quasar.cs.berkeley.edu) or udp port 53”



## How to write filters

- Refer cheat sheet slides at the end of this presentation
- Refer the tcpdump/tshark man page



## Running tcpdump

- Requires superuser/administrator privileges
- EECS instructional accounts
  - You have access to setuid versions of tcpdump/tshark
    - /share/b/ee122/tcpdump
    - /share/b/ee122/{i86pc,sun4u}/bin/tshark ← Wireshark here too
    - /bin/bash
    - alias tcpdump='/share/b/ee122/tcpdump'
  - Only works on Solaris 10 machines listed at http://inst.eecs.berkeley.edu/cgi-bin/clients.cgi?choice=servers
- Non EECS instructional accounts
  - tcpdump, tshark & wireshark work on many different operating systems
  - Download the version for your personal desktop/laptop from

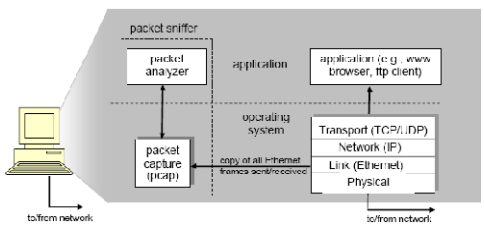


## Security/Privacy Issues

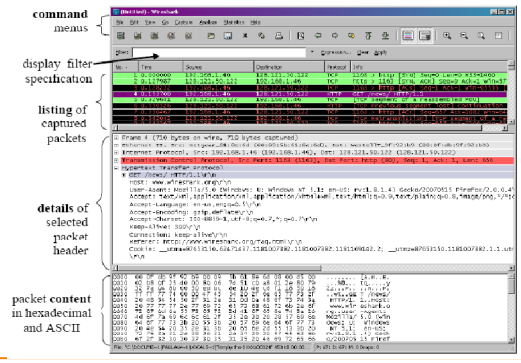
- Tcpdump/tshark/wireshark allow you to monitor other people’s traffic
- WARNING: Do **NOT** use these to violate privacy or security
- Use filtering to restrict packet analysis to only the traffic associated with your assignment. E.g., for project #1:
  - tcpdump –s 0 –w all_pkts.trace tcp port 7788



## Wireshark System Overview



## Wireshark Interface




19

## Demonstration

- Questions?

20

## Other Useful Tools

- IPsumdump
  - Handy "Swiss army knife" for displaying in ASCII fields of interest in packet trace files
  - http://www.cs.ucla.edu/~kohler/ipsumdump/
  - For instructions to use IPsumdump on EECS instructional accounts, see slide "Appendix: IPsumdump on EECS instructional accounts"
- Libpcap
  - Unix packet capture library on which tcpdump/tshark are built
  - http://www.tcpdump.org/

21

## Assignment Requirements

- **tcpdump *-w <dump_file_name> -s 0*** options must be used for the traces submitted as part of the assignments
  - tshark doesn't require -s 0 (default)
- Appropriately name each dump file you submit and briefly describe what each dump file contains/illustrates in the README file associated with the assignment submission

22

## Cheat Sheet – Commonly Used Tcpdump Options

- ***-n*** Don't convert host addresses to names. Avoids DNS lookups. It can save you time.
- ***-w <filename>*** Write the raw packets to the specified file instead of parsing and printing them out. Useful for saving a packet capture session and running multiple filters against it later
- ***-r <filename>*** Read packets from the specified file instead of live capture. The file should have been created with –w option
- ***-q*** Quiet output. Prints less information per output line

23

## Cheat Sheet – Commonly Used Options (contd.)

- ***-s 0*** tcpdump usually does not analyze and store the entire packet. This option ensures that the entire packet is stored and analyzed. NOTE: You must use this option while generating the traces for your assignments. (Default in tshark)
- ***-A (or –X in some versions)*** Print each packet in ASCII. Useful when capturing web pages. NOTE: The contents of the packet before the payload (for example, IP and TCP headers) often contain unprintable ASCII characters which will cause the initial part of each packet to look like rubbish

24

## 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

25

## Cheat Sheet – Writing Filters (2)

- Specifying ICMP packets
  - "icmp"
- Specifying UDP packets
  - "udp"
- Specifying TCP packets
  - "tcp"

26

## Cheat Sheet – Writing Filters (2)

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

    *tcpdump "tcp and !* host quasar.cs.berkeley.edu*"*
  - Lots of examples in the EXAMPLES section of the man page

27

## Appendix: IPsumdump on EECS instructional accounts

- Download and untar the latest IPsumdump source distribution from http://www.cs.ucla.edu/~kohler/ipsumdump/
- Set the following PATH and LD_LIBRARY_PATH environment variables by using *setenv* or *export* (bash shell)
  - *setenv PATH /usr/ccs/bin:$PATH*
  - *setenv LD_LIBRARY_PATH /usr/sww/lib*
- Run *./configure* followed by *make*. The executable is created in the *src/* subdirectory
- Use ipsumdump to analyze trace files generated by tcpdump (using –w option).
  - For example: *ipsumdump -r tracefile -s --payload* prints the source and payload of the packets in *tracefile* in an easy-to-read format
- (Note, these instructions are from Fall 2006 - let us know if you encounter problems with them)

28