tcpdump Tutorial

Introduction

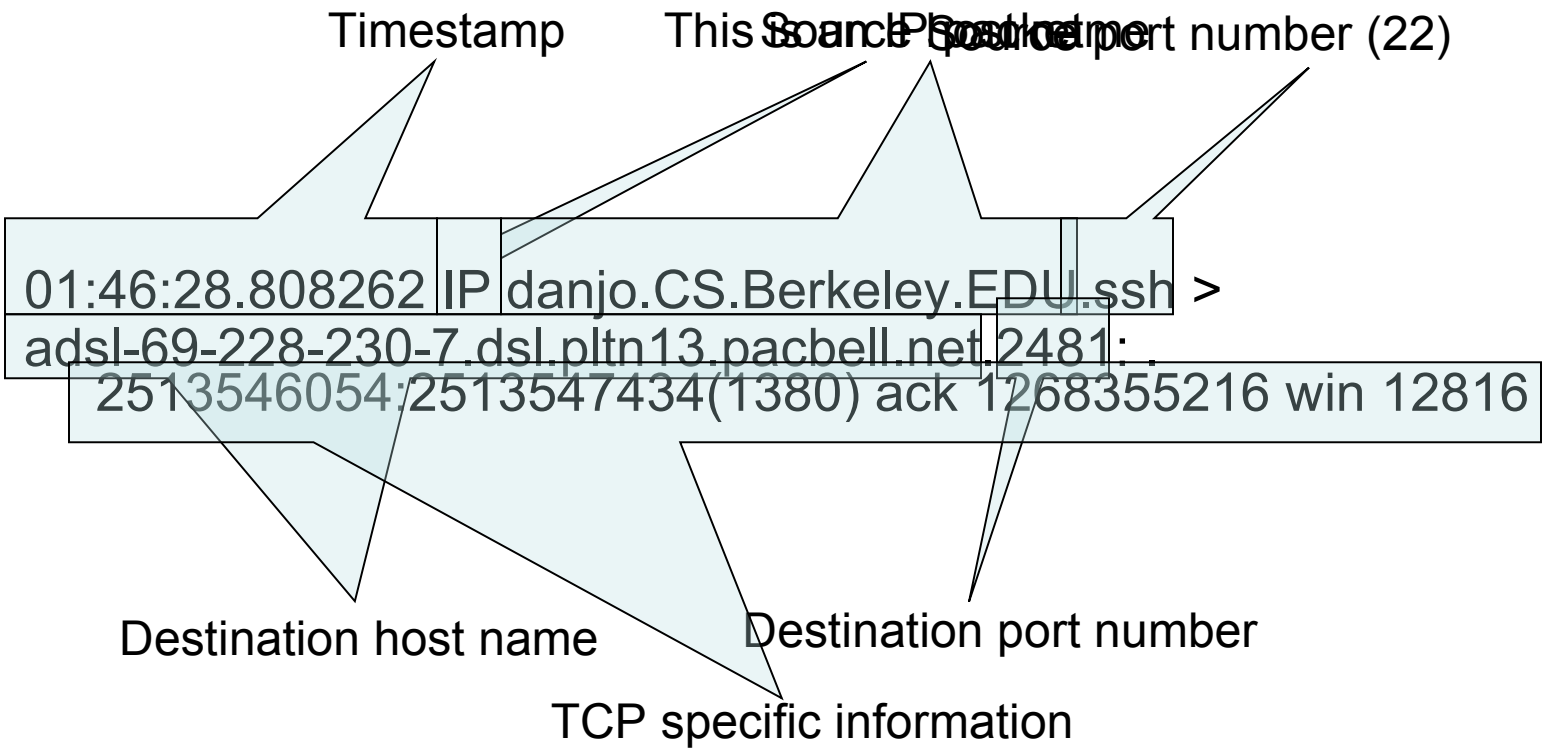
- Popular network debugging tool
- Used to intercept and display packets transmitted/received on a network
- Filters used to restrict analysis to packets of interest

Example Dump

- 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
```

What does a line convey?



- Different output formats for different packet types

Demo 1 – Basic Run

- Syntax:

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

  tcpdump
- Observe the output

Filters

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

Demo 2

1. Capture only udp packets
   - tcpdump “udp”
2. Capture only tcp packets
   - tcpdump “tcp”
- Capture only packets on a particular interface
   - tcpdump -D
   - tcpdump -i 4 -vv -XX “tcp port 12123”

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 man page

Other tools

- Ethereal
  - Easy to use graphical interface
  - http://www.ethereal.com
  - Will not currently work on EECS instructional accounts. Use on personal desktops/laptops
- IPsumdump
  - Summarize tcpdump output into human/machine readable form
  - http://www.cs.ucla.edu/~kohler/ipsumdump/
  - For instructions to use IPsumdump on EECS instructional accounts, see slide “Appendix: IPsumdump on EECS instructional accounts”

Security/Privacy Issues

- tcpdump allows you to monitor other people's traffic
- WARNING: Do NOT use tcpdump to violate privacy or security
- Use filtering to restrict packet analysis to only the traffic associated with your echo_client and echo_server. The following is one way to ensure that you see only traffic associated with your client:
  - tcpdump –s 0 –w all_pkts.trace
  - tcpdump –s 0 –r all_pkts.trace " –w my_pkts.trace "port 12345"
  - where 12345 is the ephemeral port which your echo_client uses to talk to the echo_server.

Cheat Sheet – Commonly Used 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

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.
- ***-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

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:
  - 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