Basic Network Kung-Fu: Essential Packet Processing Tools

Christian Kreibich

International Computer Science Institute

christian@icir.org

11 December 2007
Outline

- Classes of packet processing tools
- New toys!
  - BPF, libpcap, tcpdump
  - Wireshark
  - Netdude
  - Click
- There are lots more
  - Often ad-hoc and purpose-specific
  - Reflects experimental nature
Classes of packet processing tools

- Sniffing & recording
  - Crucial for reproducible experiments
- Rendering & visualization
  - Crucial for understanding what's going on
- Filtering, Processing & Editing
  - That's Where all the action is
- Injection
  - Crucial for live environments
Sniffing: **Berkeley Packet Filter (BPF)**

- **De-facto standard for grabbing packets**
Recording: **libpcap & tcpdump**

- De-facto standard tools using BPF
- **libpcap** provides user-level programming interface for packet capture
  - (And some injection too.)
Recording: **libpcap & tcpdump**

- De-facto standard tools using BPF
- **libpcap** provides user-level programming interface for packet capture
  - (And some injection too.)
- You get for each packet:
Recording: libpcap & tcpdump

- De-facto standard tools using BPF
- **libpcap** provides user-level programming interface for packet capture
  - (And some injection too.)

You get for each packet:
- a header structure:
  ```c
  struct pcap_pkthdr {
    struct timeval ts;  /* time stamp */
    bpf_u_int32 caplen; /* length of portion present */
    bpf_u_int32 len;    /* length this packet (off wire) */
  };
  ```
Recording: **libpcap & tcpdump**

- De-facto standard tools using BPF
- **libpcap** provides user-level programming interface for packet capture
  - (And some injection too.)
- You get for each packet:
  - a header structure:
    ```c
    struct pcap_pkthdr {
      struct timeval ts; /* time stamp */
      bpf_u_int32 caplen; /* length of portion present */
      bpf_u_int32 len; /* length this packet (off wire) */
    };
    ```
  - a pointer to the raw data:
    ```c
    u_char *
    ```
Recording: **libpcap & tcpdump**

- **tcpdump** is a user-level tool for capturing and recording packets
- Supports complex network and transport-level filtering, translated into BPF expressions
- Can also read recorded traces and stdin
  - This enables *stream processing*

- [http://www.tcpdump.org](http://www.tcpdump.org)
Visualization: Wireshark

- Formerly known as **Ethereal**
- Visual inspection of captured packets
  - Great for understanding/learning “in the small”
- Lots of features
Visualization: **Wireshark**

- Formerly known as **Ethereal**
- Visual inspection of captured packets
  - Great for understanding/learning “in the small”
- Lots of features
  - that I never use :-(
Visualization: Wireshark

- Formerly known as Ethereal
- Visual inspection of captured packets
  - Great for understanding/learning “in the small”
- Lots of features
  - that I never use :-)
- Understands *lots* of protocols
Visualization: **Wireshark**

- Formerly known as **Ethereal**
- Visual inspection of captured packets
  - Great for understanding/learning “in the small”
- Lots of features
  - that I never use :-)
- Understands *lots* of protocols
  - but sucks at recognizing them
Visualization: Wireshark

- Formerly known as **Ethereal**
- Visual inspection of captured packets
  - Great for understanding/learning “in the small”
- Lots of features
  - that I never use :-)
- Understands *lots* of protocols
  - but sucks at recognizing them
- Additional tools included (tshark in particular)
- [http://www.wireshark.org/](http://www.wireshark.org/)
Editing & Visualization: Netdude

- Sometimes tcpdump is not enough
  - cannot modify packets
  - stream model not always adequate

- Developed to facilitate IDS evasion testing
  - currently purely trace-based

- Netdude provides
  - a plugin framework for packet processing
  - command line and graphical interfaces
  - smart memory usage

- [http://netdude.sourceforge.net](http://netdude.sourceforge.net)
Editing & Visualization: Netdude
Sniffing, Processing, Injection: Click

- Build your own packet processing engine
  - Very popular in the research community
- Originally meant as a routing platform
- Idea: “click” functional modules together
  - Lots of these exist – NAT, measuring, queueing, recording, ... 
- Supports live and recorded traffic
- Similar to Bro in some design aspects

http://www.read.cs.ucla.edu/click/
Recommended Reading

- **Security Power Tools**, O'Reilly
- **The Tao of Network Security Monitoring**, Addison/Wesley