Network “sniffing”—packet capture and analysis

featuring Wireshark

David Morgan
Packet sniffer

- A tool that captures, interprets, and stores network packets for analysis
- also known as
  - network sniffer
  - network monitor
  - packet capture utility
  - protocol analyzer
Wireshark product background

- principal author Gerald Combs
- original name “ethereal,” still appears
- open source
- equivalent Linux and Windows versions
Related software

- **pcap**
  - the underlying library
  - pcap captures the packets
  - Wireshark displays them (graphically)

- **tcpdump**
  - rides on pcap like Wireshark
  - displays what pcap captures (character mode)
  - very widespread

- **others**
  - tshark, character mode version in Wireshark’s stable
  - Network Monitor - Microsoft
  - dSniff
  - Ettercap
  - snoop - Sun Microsystems
  - snort
Foundation concept: frames

- are what Wireshark is for capturing
- a.k.a. packets, datagrams, segments, protocol data units
- they come in nested groups
Nesting / successive enveloping

Russian laquer dolls
How data gets enveloped

THE 7 LAYERS OF OSI

TRANSMIT

DATA

APPLICATION LAYER

PRESENTATION LAYER

SESSION LAYER

TRANSPORT LAYER

NETWORK LAYER

DATA LINK LAYER

PHYSICAL LAYER

PHYSICAL LINK

RECEIVE

DATA

APPLICATION LAYER

PRESENTATION LAYER

SESSION LAYER

TRANSPORT LAYER

NETWORK LAYER

DATA LINK LAYER

PHYSICAL LAYER

PACKETS

© David Morgan 2007-16
Packets have detailed structure

**MAC Header**
- (14 bytes)
- Destination MAC Address
- Source MAC Address
- EtherType

**Ethernet Type II Frame**
- (64 to 1518 bytes)
- Data

**IP, ARP, etc. Payload**

**PING (Echo/Echo Reply)**
- Bit Number
  - 111111111112222222223
- Identifier
  - 01234567890123456789012345678901
- Code (0)
- Checksum
- Sequence Number

**TCP Header**
- Bit Number
  - 1111111111122222222223
- Source Port
- Destination Port
- Sequence Number
- Acknowledgment Number
- Source Address
- Destination Address
- Options (optional)
Packets have detailed structure

- Wireshark knows the structures
- for ~2000 protocols
- turns byte dump into intelligible decode, in the details pane
Wireshark interface components

- Packet list pane
- Packet details pane
- Packet bytes pane

Packet 6’s details
Packet 6’s bytes
Stack correlation

- application
- transport
- network
- data link
- physical

highest-layer protocol that each packet contains
Wireshark taps interfaces

- probe takes measurement “where it is”
- sees whatever is at the interface (e.g., NIC)
- sees nothing else
- does not see “what’s on the network”
- limits value on host connected to a switch (versus a hub)
It’s 70° in L.A.

No, it’s 70° right here
There's a port scan on the network

No, there's a port scan right here
Two what-to-capture restrictions

- Involuntary: can’t capture what doesn’t appear on the interface in the first place
- Voluntary: packet filter expressions
Packet filter expressions using address primitives

- host 200.2.2.1
- src host 200.2.2.2
- dst host 200.2.2.2
- ‘ip[16] >= 224’
- ‘ip[2:2] > 512’
- ‘ether[0] & 1 = 1’
Packet filter expressions using protocol primitives

- ip
- tcp
- udp
- icmp
Booleans

- and
- or
- not
2 different filters, 2 different syntaxes

- capture filters (during capture)
  - shares same syntax as tcpdump uses
- display filters (after the fact)
  - Wireshark’s own syntax
  - can auto-generate filter expression from model packet
These syntaxes semantically same

- Enter capture filter here before capturing
- Enter display filter here while displaying
If you want to see network traffic besides your own

- make sure NIC is in promiscuous mode
- operate in a network with a hub, not a switch
  - not your choice if you’re not net admin
- use a switch with a management port that receives all traffic
- sniff by remote access on computers at other places in the network, save the capture to a file, transfer the file to Wireshark
Hub – B gets A-to-C traffic

see https://www2.sans.org/resources/idfaq/switched_network.php
Switch – B denied A-to-C traffic

B of limited value for getting network-wide view of activity
info

- http://www.wireshark.org/
- http://wiki.wireshark.org/
- “Packet Sniffing In a Switched Environment”
  - https://www.sans.org/reading-room/whitepapers/networkdevs/packet-sniffing-switched-environment-244