Network "sniffing"—
packet capture and analysis

October 5, 2012

Administrative – submittal instructions

- answer the lab assignment's 13 questions in numbered list form, in a Word document file. (13th response is to embed a screenshot graphic.)
- email to csci530l@usc.edu
- exact subject title must be "snifflab"
- deadline is start of your lab session the following week
- reports not accepted (zero for lab) if
  - late
  - you did not attend the lab (except DEN or prior arrangement)
  - email subject title deviates

Administrative – free week

- no lab next Wednesday nor Friday Oct 10 and 12
- no lecture next Friday Oct 12

- this lab performed Oct 17 and19
- Next lecture Oct 19

- do your best on on midterm!

Calendar schedule

Fri lecture	Topic	lab meeting week of:	due:
9/7	Cryptography	9/10	
9/14	Authentication	9/17	
9/21	Authorization	9/24	
9/28	Application security	10/1	10/19
10/5	Packet sniffing		
	- 10/12 Midterm -		
	Packet sniffing	10/15	10/26 Fri
10/19	Firewalls	(remote)	10/29 Mon
10/26	Intrusion detection	10/29	11/9 Fri
11/2	arp spoofing	(remote)	11/12 Mon
11/9	Tunnels and VPNs	(remote)	11/19 Mon
11/16	Computer forensics	(remote)	11/30 Fri

DETER preparations

- coming soon
  - next lecture topic will be done on DETER
  - so will 3 of the 4 remaining thereafter
- you have an account
  - created a while ago
  - you received an advisory email at that time
- to-do for you - if not already done
  - make profile changes requested per your advisory email
  - do the "get your feet wet" exercise, per web page
  - review https://education.deterlab.net/DETERintro/DETERintro.html#access

DETER activity timing

- machine quotas in place for our whole class
  - software enforced by DETER
  - timed to match our assignment scchedule
- formal reservation intervals in DETER's schedule database per next slide
- assignments to be done during those intervals
- due dates per earlier slide or class web page

DETER calendar (4 labs)

Fri lecture	Topic	interval to do lab*
10/5	Packet sniffing	non-deter
→ 10/19	Firewalls	10/20 – 10/26 (inclusive)
10/26	Intrusion detection	non-deter
→ 11/2	arp spoofing	11/3 – 11/9
→ 11/9	Tunnels and VPNs	11/10 – 11/16
→ 11/16	Computer forensics	11/17 – 11/23

* when we have machine reservations in place with DETER administration

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
- is intimately "network-y"

Sniffing in security context

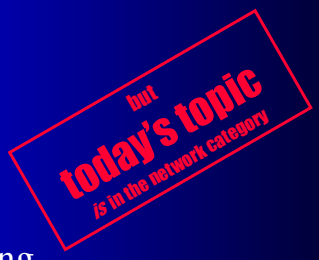
an introductory counterpoint

- conventional wisdom
  - "hacking" is emblematic of poplular security talk
  - and is all about the outside menace
  - popular conculsion: "security is about networks"
- reality
  - the outside is there

    but don't forget
  - the inside too!! does "security" vanish when net cable unplugged?

Half of security *unrelated* to nets

- purely local dimensions
  - physical security
  - BIOS/bootloader security
  - filesystem permissions
  - execution jails
  - encrypted filesystems
  - etc
- network aspects
  - packet sniffing
  - remote backup and logging
  - port scanning
  - tunnels



Wireshark product background

- principal author Gerald Combs
- original name "ethereal" (changed 2006, legal reasons)
- 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
  - snoop - Sun Microsystems
  - ettercap
  - snort

netcat product background

- a "general purpose" client and server
- there's more than one (hobbit's, GNU's)
  - different authors
  - different features
  - different syntax
- cryptcat
  - adds filestream en/de-cryption
- for you to generate something to send a server in this exercise

ssh – secure shell

- creates an encrypted network conversation
- for you to compare with an unencrypted one in this exercise
- by capturing both

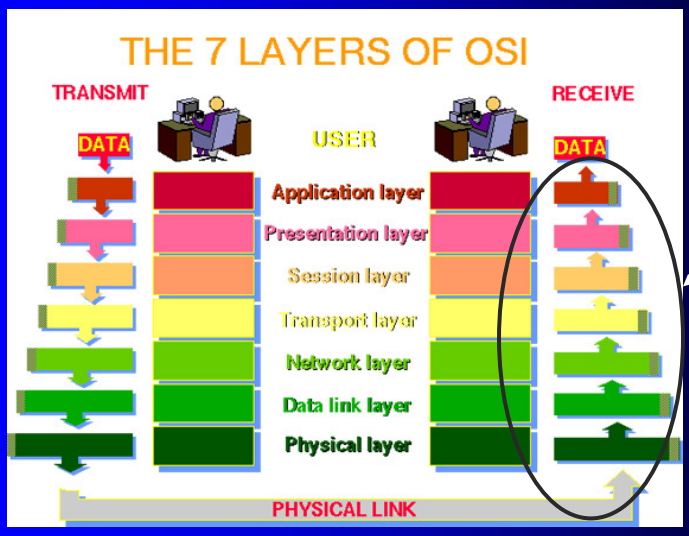
Foundation concept: frames

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

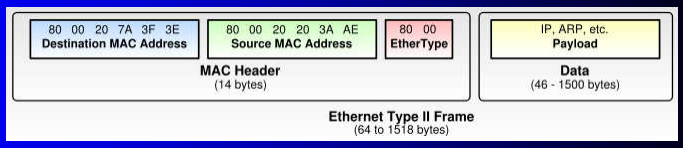
Nesting / successive enveloping

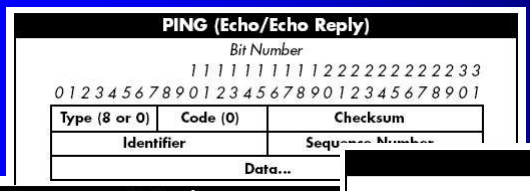
Russian laquer dolls

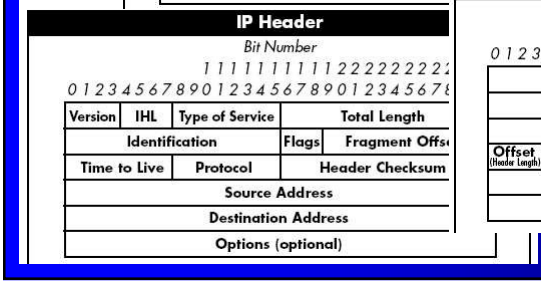
How data gets enveloped

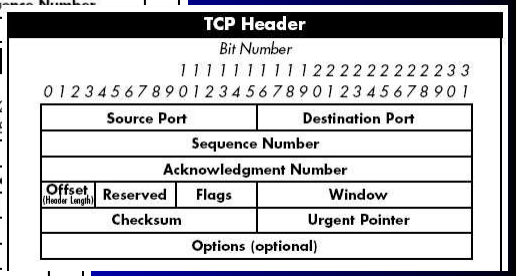


Packets have detailed structure

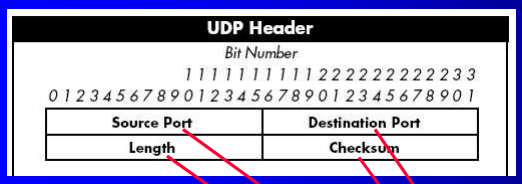


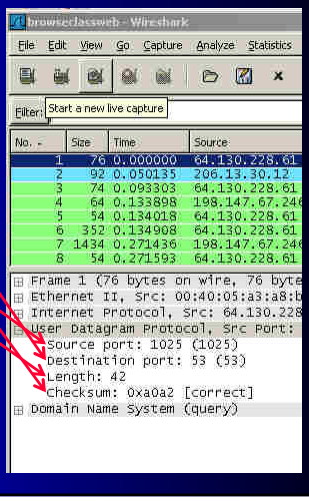






Packets have detailed structure





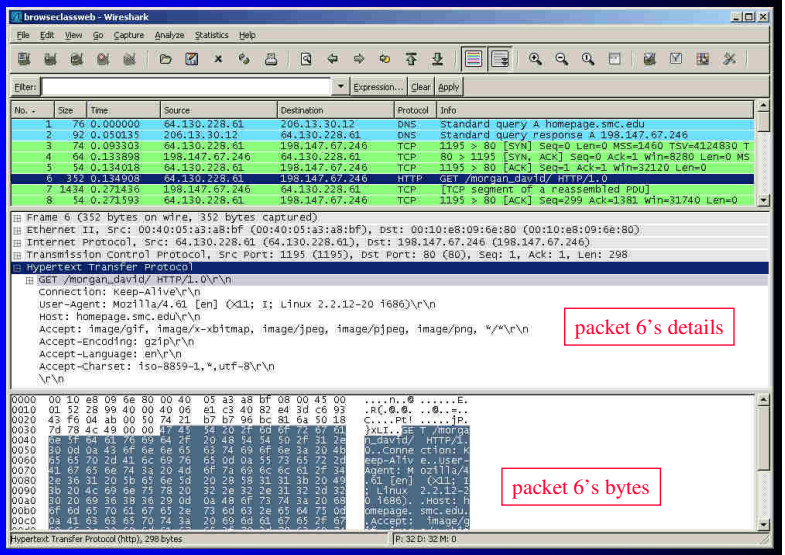
- Wireshark knows the structures
- for 800-900 protocols
- turns byte dump into intelligible decode, in the details pane

Wireshark interface components

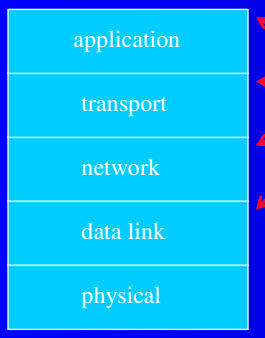
packet list pane →

packet details pane →

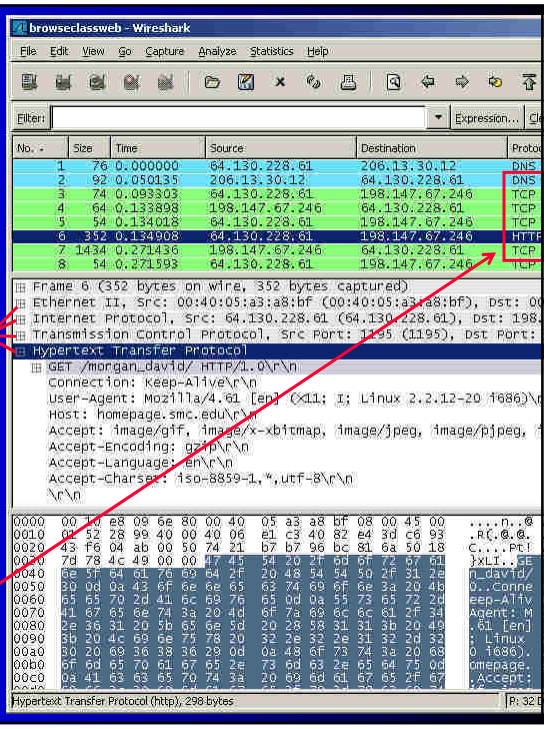
packet bytes pane →



Stack correlation



highest-layer protocol that each packet contains



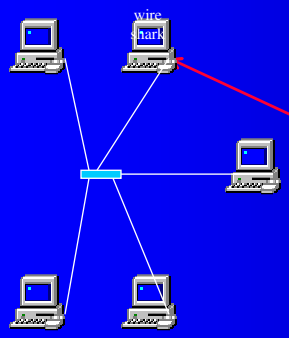
Wireshark taps interfaces

- 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

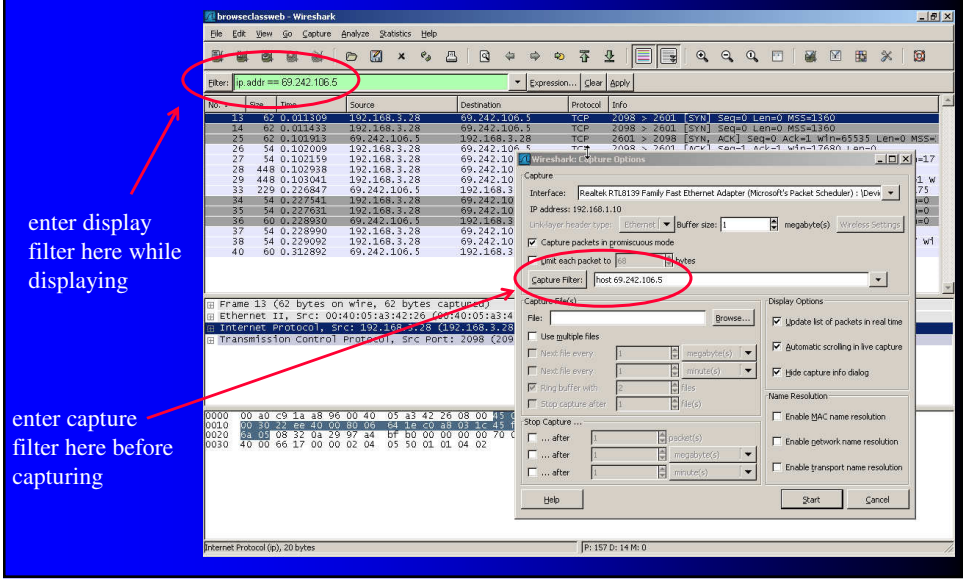
Booleans

- 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



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

info

- http://www.wireshark.org/
- http://wiki.wireshark.org/
- “Packet Sniffing In a Switched Environment”
  - http://www.sans.org/reading_room/whitepapers/networkdevs/244.php