Packet Capture
Wireshark

Fakrul Alam and Yoshinobu Matsuzaki

Why we need to capture packet & how it's related to security?

- Hosts send and receive packets to communicate with each others. When you have network related issues, sometimes it's necessary to **see the actual packets to troubleshoot.**
- And also it helps to **understand protocols and devices' behaviors**

tcpdump Definition

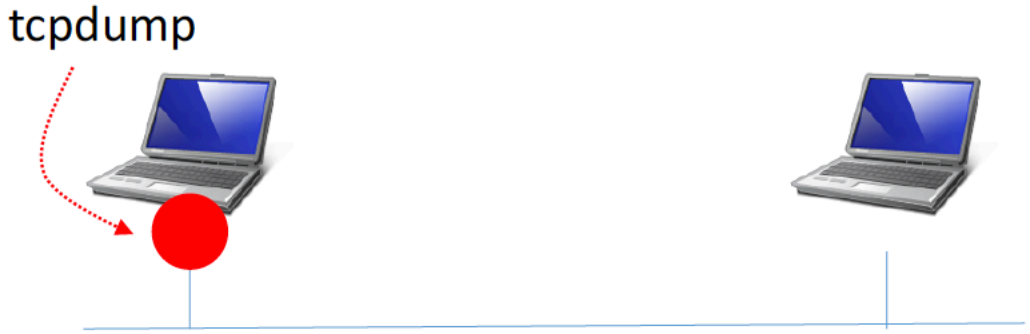
- tcpdump is a utility used to capture and analyze packets on network interfaces.
- Details about these packets can either be displayed to the screen or they can be saved to a file for later analysis.
- tcpdump utilizes the libpcap library for packet capturing.
- And there are variant of the software
  - Wireshark

What you can see

Packets arrived on your network interface

Packets to/from the host

Unicast, Multicast, Broadcast



You might not see others' communication

tcpdump command example

```
# tcpdump –nni eth0
# tcpdump –nni eth0 host 10.10.10.10
# tcpdump –nni eth0 dst host 10.10.10.10 and tcp
# tcpdump –nni eth0 src net 10.10.10.0/24 and tcp and portrange 1-1024
```

-nn = don’t use DNS to resolve IPs and display port no
-i = interface to watch
dst = watch only traffic destined to a net, host or port
src = watch only traffic whose src is a net, host or port
net = specifies network
host = specifies host
port = specifies a port
proto = protocol ie tcp or udp

tcpdump command example

```
# tcpdump –nni eth0 –s0

# tcpdump –nni eth0 not port 22 –s0 –c 1000 -v

# tcpdump –nni eth0 not port 22 and dst host 10.10.10.10 and not src
net 10.20.30.0/24
```

-s = packet length to be captured

-s0 = setting samples length to 0 means use the required length to catch whole packet

-c = number of packets

-v = print a more verbose description

tcpdump pcaps

```
# tcpdump –nni eth0 -w capture.pcap –vv –c 1000
# tcpdump –nni eth0 –r capture.pcap port 80
```

-w capture.pcap = save capture packet to capture.pcap

–vv = display number of packet captured

-r capture.pcap = read capture file

-c = number of packets

tcpdump Output

```
IP 199.59.148.139.443 > 192.168.1.8.54343: Flags [P.], seq 53:106,
ack 1, win 67, options [nop,nop,TS val 854797891 ecr 376933204],
length 53

IP 192.168.1.8.54343 > 199.59.148.139.443: Flags [.], ack 106, win
4092, options [nop,nop,TS val 376934736 ecr 854797891], length 0

IP 199.59.148.139.443 > 192.168.1.8.54343: Flags [P.], seq 106:159,
ack 1, win 67, options [nop,nop,TS val 854797891 ecr 376933204],
length 53

IP 192.168.1.8.54343 > 199.59.148.139.443: Flags [.], ack 159, win
4091, options [nop,nop,TS val 376934736 ecr 854797891], length 0
```

What is Wireshark?

- Wireshark is a network packet/protocol analyzer.
  - A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible.
- Wireshark is perhaps one of the best open source packet analyzers available today for **UNIX** and **Windows**.

Wireshark

- Yet another packet capturing and analysis software
- GUI
- It can handle the tcpdump file format (pcap)
- You can analyze packets which is captured on a remote server using tcpdump

About Wireshark

- Formerly known as “Ethereal”
  - Author, Gerald Combs quit Network Integration Services
  - Free
- https://www.wireshark.org/

  Packages for Windows and MacOS are available

  Most UNIX system has own package of Wireshark
- Requirement on windows
  - Need to install winpcap
  - Latest wireshark installer contains winpcap, don’t worry
  - (On Windows Vista) Need Administrator Privilege to capture

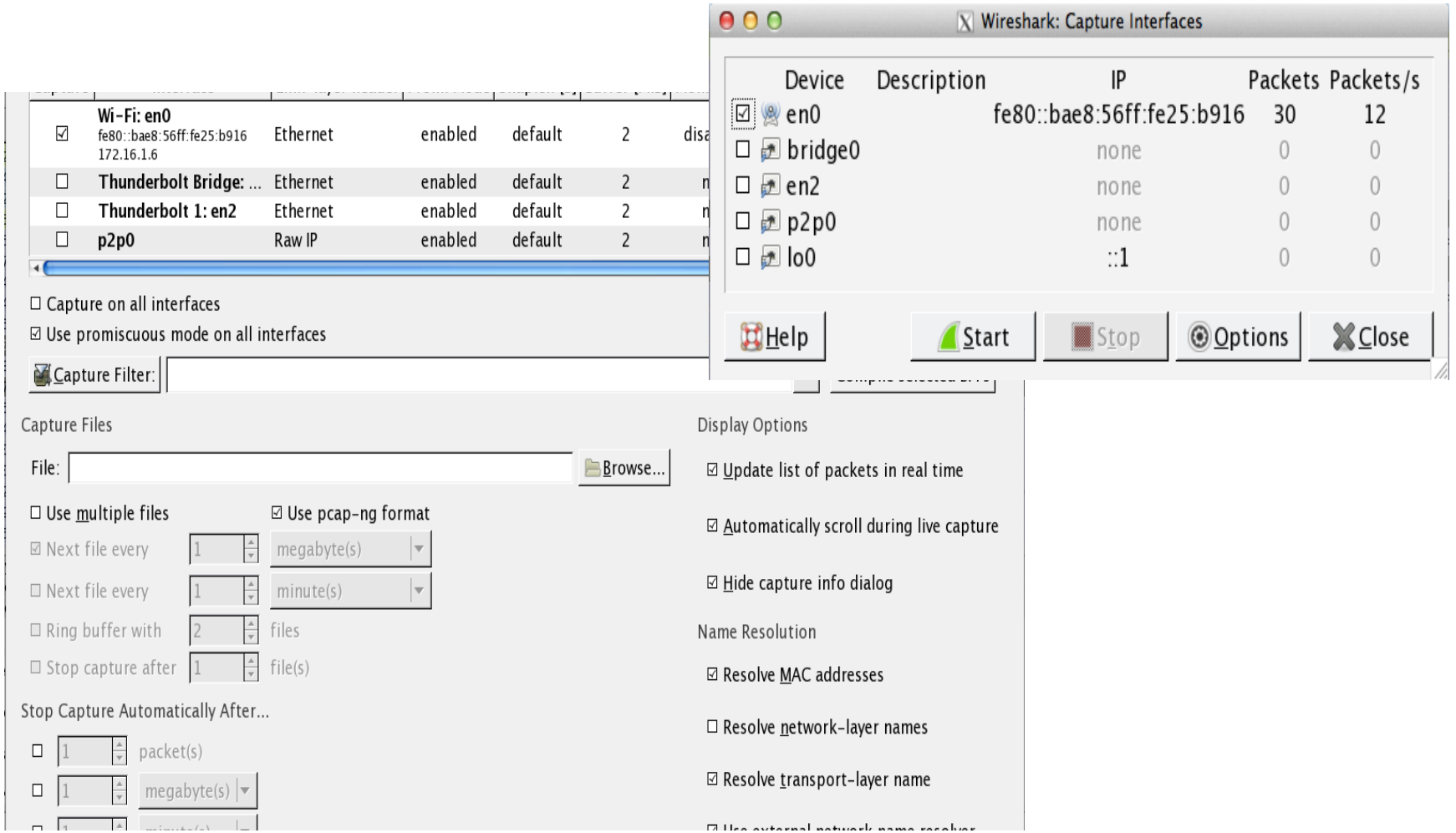
Why Wireshark

- network administrators use it to **troubleshoot network problems**
- network security engineers use it to **examine security problems**
- developers use it to **debug protocol implementations**
- people use it to **learn network protocol** internals
- Wireshark isn't an intrusion detection system.
- Wireshark will not manipulate things on the network, it will only "measure" things from it.

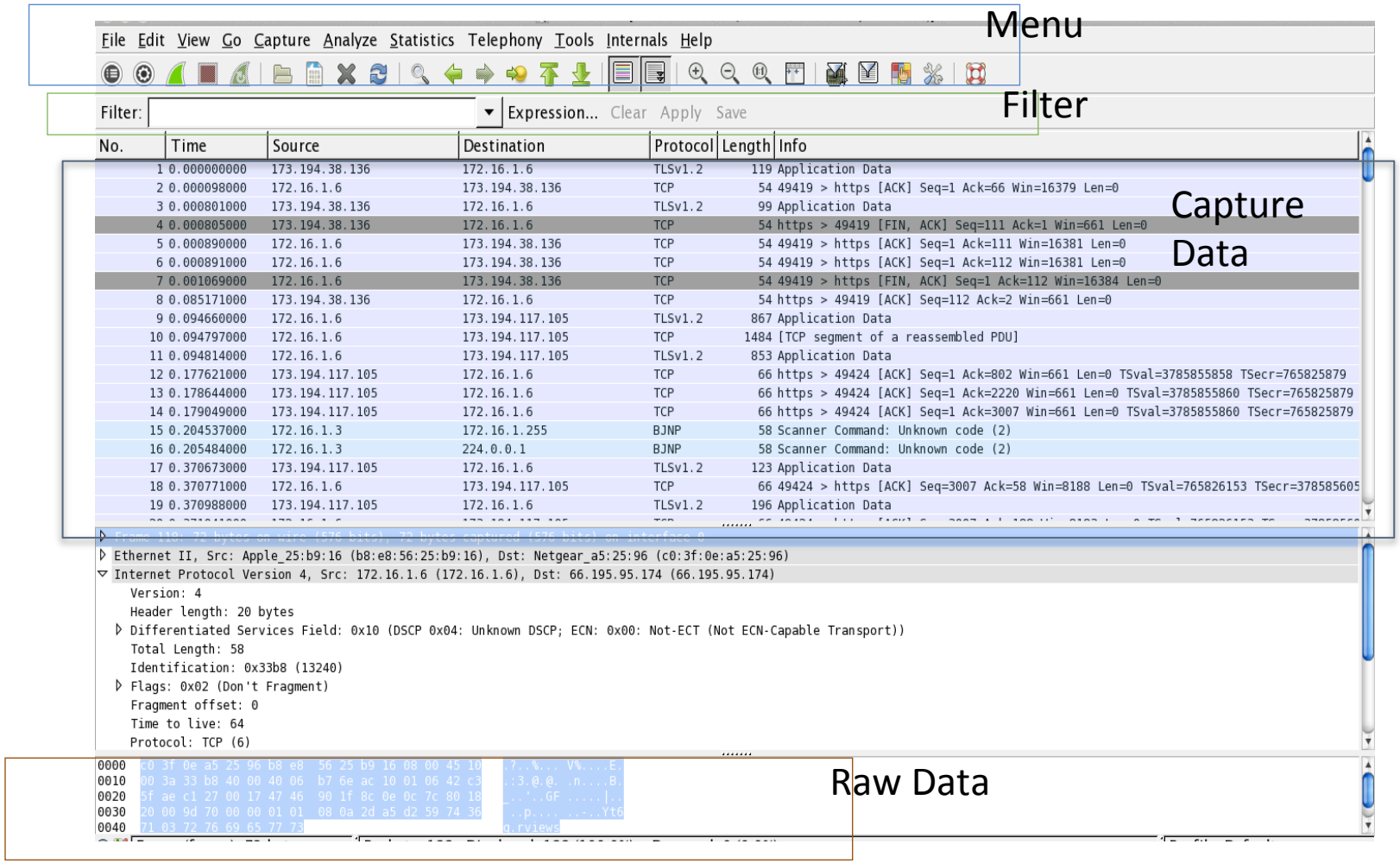
How to Install

- Very straight forward
- Just double-click and follow the instructions.

Capture


Wireshark: Capture Interfaces
Device Description IP Packets Packets/s
en0 fe80::bae8:56ff:fe25:b916 30 12
bridge0 none 0 0
en2 none 0 0
p2p0 none 0 0
lo0 ::1 0 0
Help Start Stop Options Close
Wi-Fi: en0
fe80::bae8:56ff:fe25:b916
172.16.1.6
Ethernet enabled default 2
Thunderbolt Bridge: ... Ethernet enabled default 2
Thunderbolt 1: en2 Ethernet enabled default 2
p2p0 Raw IP enabled default 2
Capture on all interfaces
Use promiscuous mode on all interfaces
Capture Filter:
Capture Files
File:
Browse...
Use multiple files
Use pcap-ng format
Next file every 1 megabyte(s)
Next file every 1 minute(s)
Ring buffer with 2 files
Stop capture after 1 file(s)
Stop Capture Automatically After...
1 packet(s)
1 megabyte(s)
Display Options
Update list of packets in real time
Automatically scroll during live capture
Hide capture info dialog
Name Resolution
Resolve MAC addresses
Resolve network-layer names
Resolve transport-layer name

Dashboard


Menu
Filter
Capture Data
Raw Data
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter: Expression... Clear Apply Save
No. Time Source Destination Protocol Length Info
1 0.000000000 173.194.38.136 172.16.1.6 TLSv1.2 119 Application Data
2 0.000098000 172.16.1.6 173.194.38.136 TCP 54 49419 > https [ACK] Seq=1 Ack=66 Win=16379 Len=0
3 0.000801000 173.194.38.136 172.16.1.6 TLSv1.2 99 Application Data
4 0.000805000 173.194.38.136 172.16.1.6 TCP 54 https > 49419 [FIN, ACK] Seq=111 Ack=1 Win=661 Len=0
5 0.000890000 172.16.1.6 173.194.38.136 TCP 54 49419 > https [ACK] Seq=1 Ack=111 Win=16381 Len=0
6 0.000891000 172.16.1.6 173.194.38.136 TCP 54 49419 > https [ACK] Seq=1 Ack=112 Win=16381 Len=0
7 0.001069000 172.16.1.6 173.194.38.136 TCP 54 49419 > https [FIN, ACK] Seq=1 Ack=112 Win=16384 Len=0
8 0.085171000 173.194.38.136 172.16.1.6 TCP 54 https > 49419 [ACK] Seq=112 Ack=2 Win=661 Len=0
9 0.094660000 172.16.1.6 173.194.117.105 TLSv1.2 867 Application Data
10 0.094797000 172.16.1.6 173.194.117.105 TCP 1484 [TCP segment of a reassembled PDU]
11 0.094814000 172.16.1.6 173.194.117.105 TLSv1.2 853 Application Data
12 0.177621000 173.194.117.105 172.16.1.6 TCP 66 https > 49424 [ACK] Seq=1 Ack=802 Win=661 Len=0 TSval=3785855858 TSecr=765825879
13 0.178644000 173.194.117.105 172.16.1.6 TCP 66 https > 49424 [ACK] Seq=1 Ack=2220 Win=661 Len=0 TSval=3785855860 TSecr=765825879
14 0.179049000 173.194.117.105 172.16.1.6 TCP 66 https > 49424 [ACK] Seq=1 Ack=3007 Win=661 Len=0 TSval=3785855860 TSecr=765825879
15 0.204537000 172.16.1.3 172.16.1.255 BJNP 58 Scanner Command: Unknown code (2)
16 0.205484000 172.16.1.3 224.0.0.1 BJNP 58 Scanner Command: Unknown code (2)
17 0.370673000 173.194.117.105 172.16.1.6 TLSv1.2 123 Application Data
18 0.370771000 172.16.1.6 173.194.117.105 TCP 66 49424 > https [ACK] Seq=3007 Ack=58 Win=8188 Len=0 TSval=765826153 TSecr=378585605
19 0.370988000 173.194.117.105 172.16.1.6 TLSv1.2 196 Application Data
Ethernet II, Src: Apple_25:b9:16 (b8:e8:56:25:b9:16), Dst: Netgear_a5:25:96 (c0:3f:0e:a5:25:96)
Internet Protocol Version 4, Src: 172.16.1.6 (172.16.1.6), Dst: 66.195.95.174 (66.195.95.174)
Version: 4
Header length: 20 bytes
Differentiated Services Field: 0x10 (DSCP 0x04: Unknown DSCP; ECN: 0x00: Not-ECT (Not ECN-Capable Transport))
Total Length: 58
Identification: 0x33b8 (13240)
Flags: 0x02 (Don't Fragment)
Fragment offset: 0
Time to live: 64
Protocol: TCP (6)
0000 c0 3f 0e a5 25 96 b8 e8 56 25 b9 16 08 00 45 10
0010 00 3a 33 b8 40 00 40 06 b7 6e ac 10 01 06 42 c3
0020 5f ae c1 27 00 17 47 46 90 1f 8c 0e 0c 7c 80 18
0030 20 00 9d 70 00 00 01 01 08 0a 2d a5 d2 59 74 36
0040 71 03 72 76 69 65 77 73

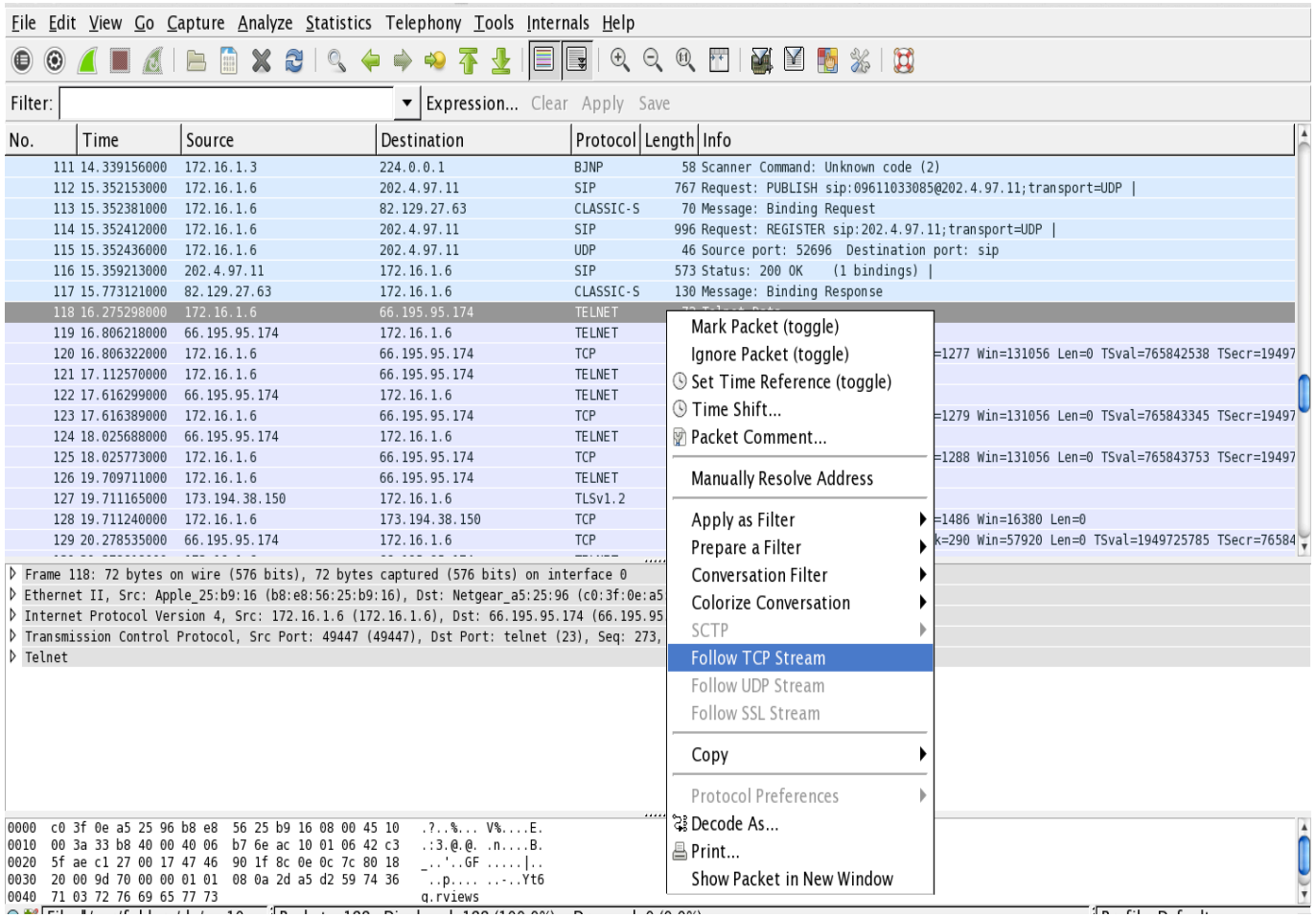
Filters

- Capture filter
  - Capture Traffic that match capture filter rule
  - save disk space
  - prevent packet loss
- Display filter
- Tweak appearance

Apply Filters

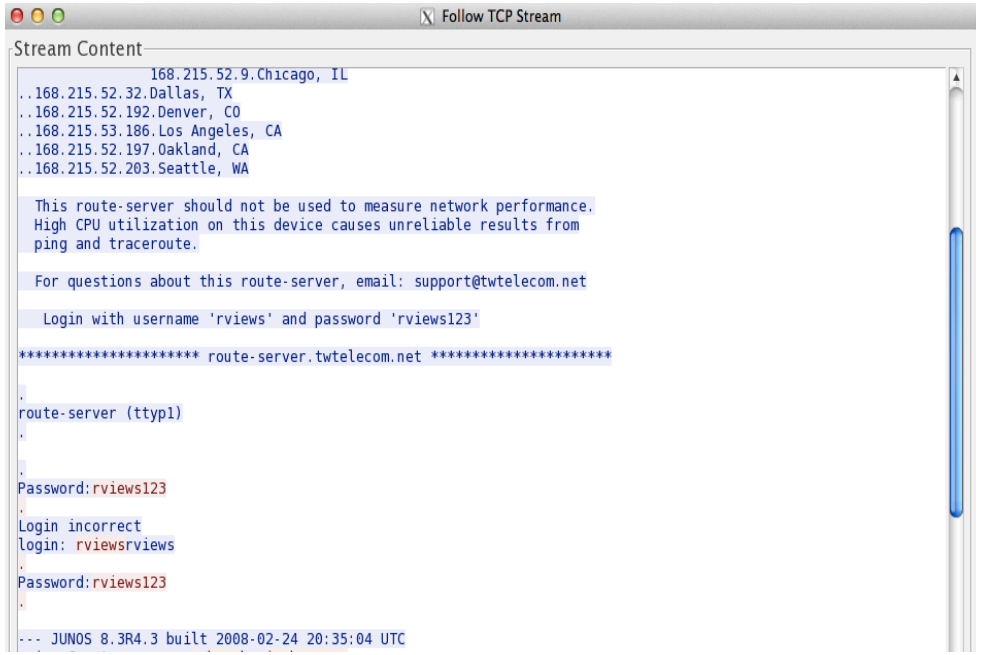
- ip.addr == 10.0.0.1 [Sets a filter for any packet with 10.0.0.1, as either the source or dest]
- ip.addr==10.0.0.1 && ip.addr==10.0.0.2 [sets a conversation filter between the two defined IP addresses]
- http or dns [sets a filter to display all http and dns]
- tcp.port==4000 [sets a filter for any TCP packet with 4000 as a source or dest port]
- tcp.flags.reset==1 [displays all TCP resets]
- http.request [displays all HTTP GET requests]
- tcp contains rviews [displays all TCP packets that contain the word 'rviews'. Excellent when searching on a specific string or user ID]
- !(arp or icmp or dns) [masks out arp, icmp, dns, or whatever other protocols may be background noise. Allowing you to focus on the traffic of interest]

Follow TCP Stream



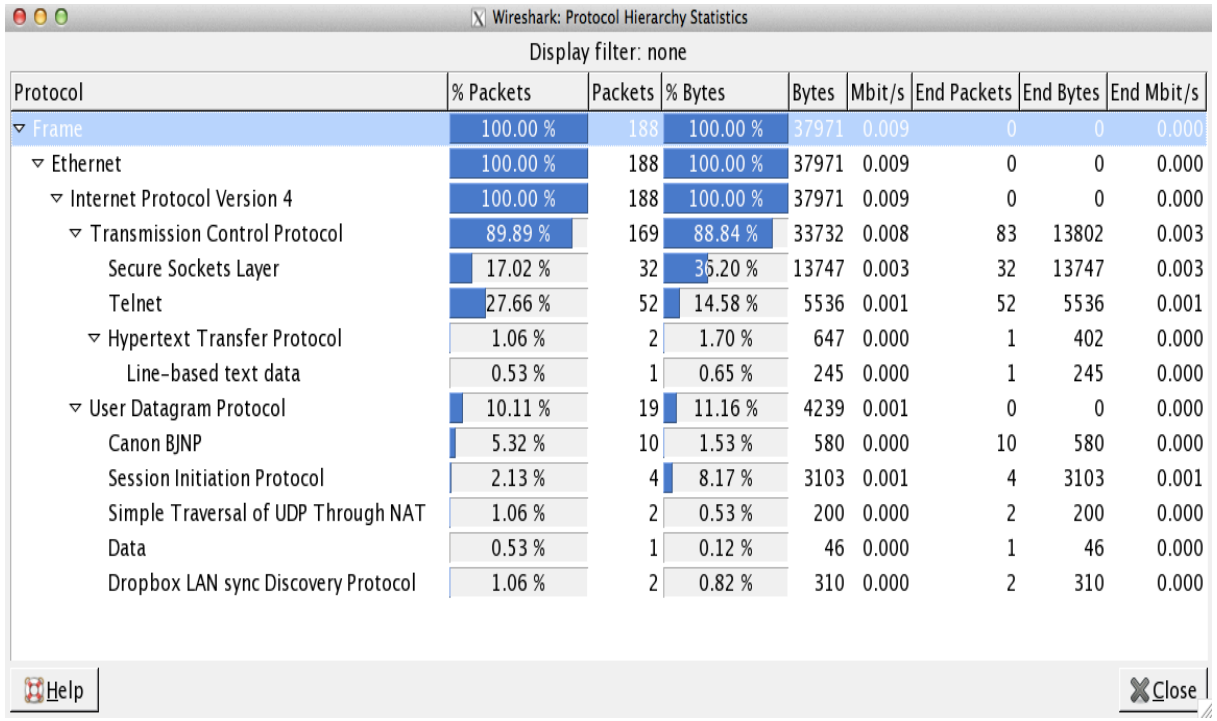
Follow TCP Stream

- Build TCP Stream
  - Select TCP Packet -> Follow TCP Stream



Use “Statistics”

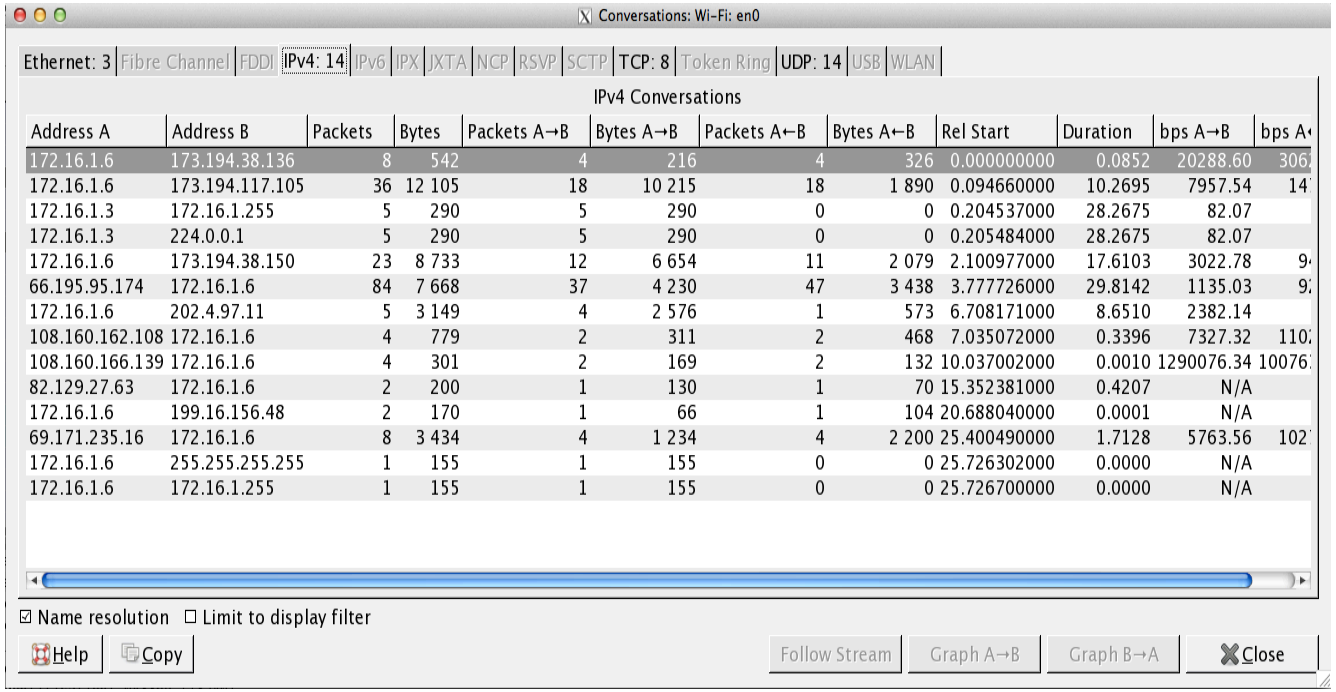
- What protocol is used in your network
  - Statistics -> Protocol Hierarchy



Protocol	% Packets	Packets	% Bytes	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
Frame	100.00 %	188	100.00 %	37971	0.009	0	0	0.000
Ethernet	100.00 %	188	100.00 %	37971	0.009	0	0	0.000
Internet Protocol Version 4	100.00 %	188	100.00 %	37971	0.009	0	0	0.000
Transmission Control Protocol	89.89 %	169	88.84 %	33732	0.008	83	13802	0.003
Secure Sockets Layer	17.02 %	32	36.20 %	13747	0.003	32	13747	0.003
Telnet	27.66 %	52	14.58 %	5536	0.001	52	5536	0.001
Hypertext Transfer Protocol	1.06 %	2	1.70 %	647	0.000	1	402	0.000
Line-based text data	0.53 %	1	0.65 %	245	0.000	1	245	0.000
User Datagram Protocol	10.11 %	19	11.16 %	4239	0.001	0	0	0.000
Canon BJNP	5.32 %	10	1.53 %	580	0.000	10	580	0.000
Session Initiation Protocol	2.13 %	4	8.17 %	3103	0.001	4	3103	0.001
Simple Traversal of UDP Through NAT	1.06 %	2	0.53 %	200	0.000	2	200	0.000
Data	0.53 %	1	0.12 %	46	0.000	1	46	0.000
Dropbox LAN sync Discovery Protocol	1.06 %	2	0.82 %	310	0.000	2	310	0.000

Use “Statistics”

- Which host most chatty
  - Statistics -> Conversations



Need CLI?

- If you stick to character based interface, try tshark.exe
- C:\program files\wireshark\tshark.exe

Tcpdump & Wireshark

- tcpdump -i <interface> -s 65535 -w <some-file>
  - -s <snaplen>: amount of bytes captures for each packet
  - -w <some-file>: write the raw packets to a file

Exercise

- Install Wireshark into your PC
- Or, you can use VM (tcpdump)

  $ ssh workshop@10.0.0.x

  Note: x is your group#

  Note: password is iij/2497
- Run wireshark and Capture inbound/outbound traffic
- Download capture files from https://github.com/randyqx/apricot-sec2019
  - 2-2-2.WiresharkExerciseData.zip

Exercise 1: Good Old Telnet

- File
  - telnet.pcap
- Question
  - Reconstruct the telnet session.
- Q1: Who logged into 192.168.0.1
  - Username _____, Password _____ .
- Q2: After logged in what did the user do?
  - Tip
  - telnet traffic is not secure

Exercise 2: Massive TCP SYN

- File
  - massivesyn1.pcap and massivesyn2.pcap
- Question
  - Point the difference between them.
- Q1: massivesyn1.pcap is a _____ attempt.
- Q2: massivesyn2.pcap is a _____ attempt.
- Tip
  - Pay attention to Src IP

Exercise 3: Chatty Employees

- File
  - chat.dmp
- Question
- Q1: What kind of protocol is used? _____
- Q2: This is a conversation between _____@hotmail.com and _____@hotmail.com
- Q3: What do they say about you(sysadmin)?
- Tip
  - Your chat can be monitored by network admin.

Exercise 4: Suspicious FTP activity

- File
  - ftp1.pcap
- Question
  - Q1: 10.121.70.151 is FTP _____ .
  - Q2: 10.234.125.254 is FTP _____ .
  - Q3: FTP Err Code 530 means _____ .
  - Q4: 10.234.125.254 attempt _____.
- Tip
  - How many login error occur within a minute?

Exercise 5: Unidentified Traffic

- File
  - Foobar.pcap
- Question
  - Q1: see what's going on with wireshark gui
    - Statistics -> Conversation List -> TCP (*)
  - Q2: Which application use TCP/6346? Check the web.

Exercise 6: Covert channel

- File
  - covertinfo.pcap
- Question
  - Take a closer look! This is not a typical ICMP Echo/Reply…
  - Q1: What kind of tool do they use? Check the web.
  - Q2: Name other application which tunnel user traffic.

Exercise 7: Analyze Malware

- File
  - malware.pcap
- Questions:
  - Q1: Find the bad HTTP traffic
  - Q2: Is there any malware in the HTTP traffic?
  - Q3: Upload one sample malware to https://www.virustotal.com/
    - Does all antivirus detect the malware?
- Tips
  - Filter with **`http contains "in DOS mode”`**
  - Export all the files

Exercise 8: SIP

- File
  - sip_chat.pcap
- Questions:
  - Q1: Can we listen to SIP voice?
  - Q2: How!!

Exercise 9: Your telnet

- Capture your telnet session to VM
  - PuTTY can telnet
  - Host Name: 10.0.0.x
  - Select Telnet
- Login: workshop
- Password: iij/2497
- Look at the captured data

Exercise 10: Your ssh

- Capture your ssh session to VM
  - $ ssh workshop@10.0.0.x
  - password is iij/2497 (or key-based authentication)

- Look at the captured data