

Packet Capture **Wireshark**

Fakrul Alam

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

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.

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
```

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

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

-c = number of packets

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**.

About Wireshark

- Formerly known as “Ethereal”
 - Author, Gerald Combs quit Network Integration Services
 - Free
- Requirement
 - Need to install winpcap
 - Latest wireshark installer contains winpcap, don't worry
 - (On Windows Vista) Need Administrator Privilege to capture
- GUI
 - Dramatically improved

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

The image displays the Wireshark interface for configuring network capture. It features a list of network interfaces, a 'Capture Interfaces' dialog box, and various configuration panels for capture files and display options.

Interface List:

| Device | Description | IP | Packets | Packets/s |
|---|---------------------------|---------------------------|---------|-----------|
| <input checked="" type="checkbox"/> en0 | fe80::bae8:56ff:fe25:b916 | fe80::bae8:56ff:fe25:b916 | 30 | 12 |
| <input type="checkbox"/> bridge0 | | none | 0 | 0 |
| <input type="checkbox"/> en2 | | none | 0 | 0 |
| <input type="checkbox"/> p2p0 | | none | 0 | 0 |
| <input type="checkbox"/> lo0 | | ::1 | 0 | 0 |

Capture Filter: [Empty]

Capture Files:

- File: [Empty] [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
- Use external network name resolver

Dashboard

The image shows a screenshot of the Wireshark network traffic analysis tool. The interface is divided into several sections:

- Menu:** Located at the top, containing standard application menus: File, Edit, View, Go, Capture, Analyze, Statistics, Telephony, Tools, Internals, and Help.
- Filter:** A search bar below the menu with a dropdown arrow and the text "Expression... Clear Apply Save".
- Capture Data:** A table listing captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The table contains 19 rows of data, including various protocols like TLSv1.2, TCP, and BJNP.
- Raw Data:** A section at the bottom showing the hexadecimal and ASCII representation of the selected packet's raw data.

| 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 |

```
0000 e0 3f 0e a5 25 96 b8 e8 56 25 b9 16 08 00 45 10  .7.%.V%...E
0010 00 3a 33 b8 40 00 40 06 b7 6e ac 10 01 06 42 c3  .3.@.@.n...B
0020 5f ae c1 27 00 17 47 46 90 1f 8c 0e 0c 7c 80 18  [...].GF.....|
0030 20 00 9d 70 00 00 01 01 08 0a 2d a5 d2 59 74 36  .p.....Yt6
0040 71 03 72 76 69 65 77 73  g.rviews
```

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

The screenshot shows the Wireshark network protocol analyzer interface. The main window displays a list of captured packets. Packet 118 is selected, and a context menu is open over it, with the 'Follow TCP Stream' option highlighted in blue. The menu options include: Mark Packet (toggle), Ignore Packet (toggle), Set Time Reference (toggle), Time Shift..., Packet Comment..., Manually Resolve Address, Apply as Filter, Prepare a Filter, Conversation Filter, Colorize Conversation, SCTP, Follow TCP Stream (highlighted), Follow UDP Stream, Follow SSL Stream, Copy, Protocol Preferences, Decode As..., Print..., and Show Packet in New Window.

| No. | Time | Source | Destination | Protocol | Length | Info |
|-----|--------------|----------------|----------------|-----------|--------|--|
| 111 | 14.339156000 | 172.16.1.3 | 224.0.0.1 | BJNP | 58 | Scanner Command: Unknown code (2) |
| 112 | 15.352153000 | 172.16.1.6 | 202.4.97.11 | SIP | 767 | Request: PUBLISH sip:09611033085@202.4.97.11;transport=UDP |
| 113 | 15.352381000 | 172.16.1.6 | 82.129.27.63 | CLASSIC-S | 70 | Message: Binding Request |
| 114 | 15.352412000 | 172.16.1.6 | 202.4.97.11 | SIP | 996 | Request: REGISTER sip:202.4.97.11;transport=UDP |
| 115 | 15.352436000 | 172.16.1.6 | 202.4.97.11 | UDP | 46 | Source port: 52696 Destination port: sip |
| 116 | 15.359213000 | 202.4.97.11 | 172.16.1.6 | SIP | 573 | Status: 200 OK (1 bindings) |
| 117 | 15.773121000 | 82.129.27.63 | 172.16.1.6 | CLASSIC-S | 130 | Message: Binding Response |
| 118 | 16.275298000 | 172.16.1.6 | 66.195.95.174 | TELNET | 72 | ... |
| 119 | 16.806218000 | 66.195.95.174 | 172.16.1.6 | TELNET | ... | |
| 120 | 16.806322000 | 172.16.1.6 | 66.195.95.174 | TCP | ... | =1277 Win=131056 Len=0 TSval=765842538 TSecr=19497 |
| 121 | 17.112570000 | 172.16.1.6 | 66.195.95.174 | TELNET | ... | |
| 122 | 17.616299000 | 66.195.95.174 | 172.16.1.6 | TELNET | ... | |
| 123 | 17.616389000 | 172.16.1.6 | 66.195.95.174 | TCP | ... | =1279 Win=131056 Len=0 TSval=765843345 TSecr=19497 |
| 124 | 18.025688000 | 66.195.95.174 | 172.16.1.6 | TELNET | ... | |
| 125 | 18.025773000 | 172.16.1.6 | 66.195.95.174 | TCP | ... | =1288 Win=131056 Len=0 TSval=765843753 TSecr=19497 |
| 126 | 19.709711000 | 172.16.1.6 | 66.195.95.174 | TELNET | ... | |
| 127 | 19.711165000 | 173.194.38.150 | 172.16.1.6 | TLSv1.2 | ... | |
| 128 | 19.711240000 | 172.16.1.6 | 173.194.38.150 | TCP | ... | =1486 Win=16380 Len=0 |
| 129 | 20.278535000 | 66.195.95.174 | 172.16.1.6 | TCP | ... | k=290 Win=57920 Len=0 TSval=1949725785 TSecr=76584 |

Frame 118: 72 bytes on wire (576 bits), 72 bytes captured (576 bits) on interface 0
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)
Transmission Control Protocol, Src Port: 49447 (49447), Dst Port: telnet (23), Seq: 273, Len: 72
Telnet

```
0000 c0 3f 0e a5 25 96 b8 e8 56 25 b9 16 08 00 45 10  .?..%...V%...E.
0010 00 3a 33 b8 40 00 40 06 b7 6e ac 10 01 06 42 c3  .:3.@.@.n...B.
0020 5f ae c1 27 00 17 47 46 90 1f 8c 0e 0c 7c 80 18  _...GF.....].
0030 20 00 9d 70 00 00 01 01 08 0a 2d a5 d2 59 74 36  .p.....Yt6
0040 71 03 72 76 69 65 77 73  q.rviews
```


Follow TCP Stream

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



```
Stream Content
168.215.52.9.Chicago, IL
..168.215.52.32.Dallas, TX
..168.215.52.192.Denver, CO
..168.215.53.186.Los Angeles, CA
..168.215.52.197.Oakland, CA
..168.215.52.203.Seattle, WA

This route-server should not be used to measure network performance.
High CPU utilization on this device causes unreliable results from
ping and traceroute.

For questions about this route-server, email: support@twtelecom.net

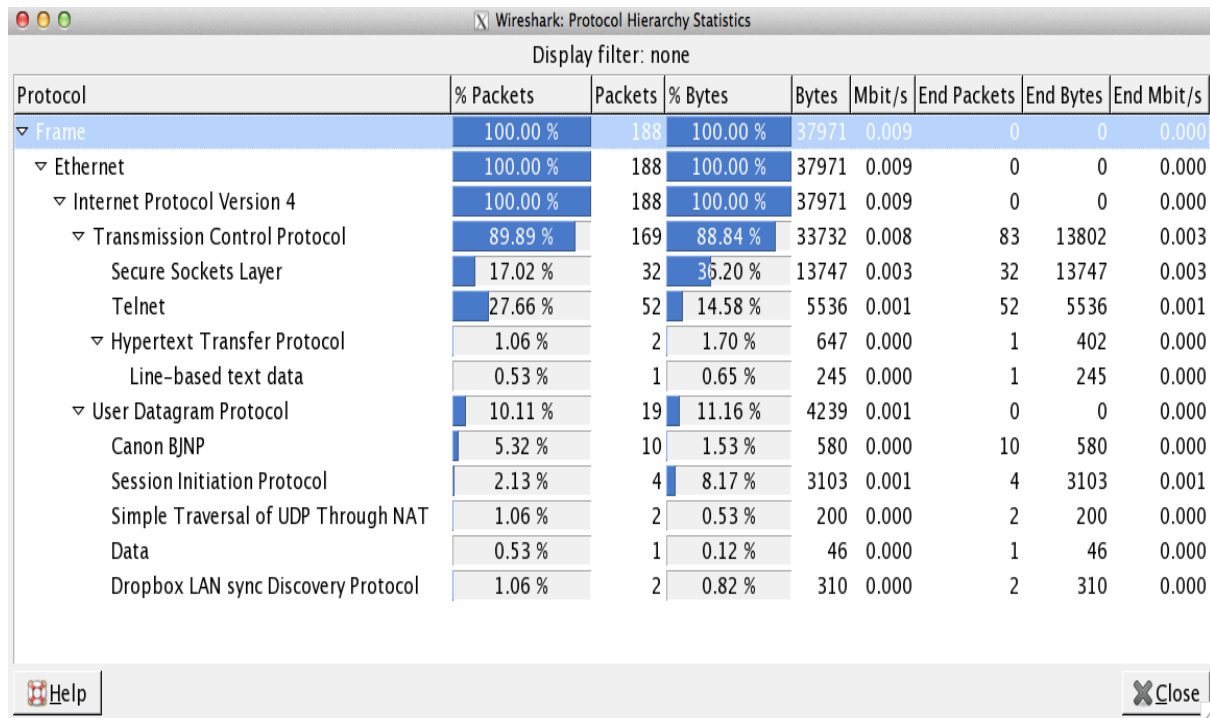
Login with username 'rviews' and password 'rviews123'

***** route-server.twtelecom.net *****

.
route-server (tty1)
.
Password: rviews123
.
Login incorrect
login: rviewsrviews
.
Password: rviews123
.
--- JUNOS 8.3R4.3 built 2008-02-24 20:35:04 UTC
```

Use “Statistics”

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



Wireshark: Protocol Hierarchy Statistics

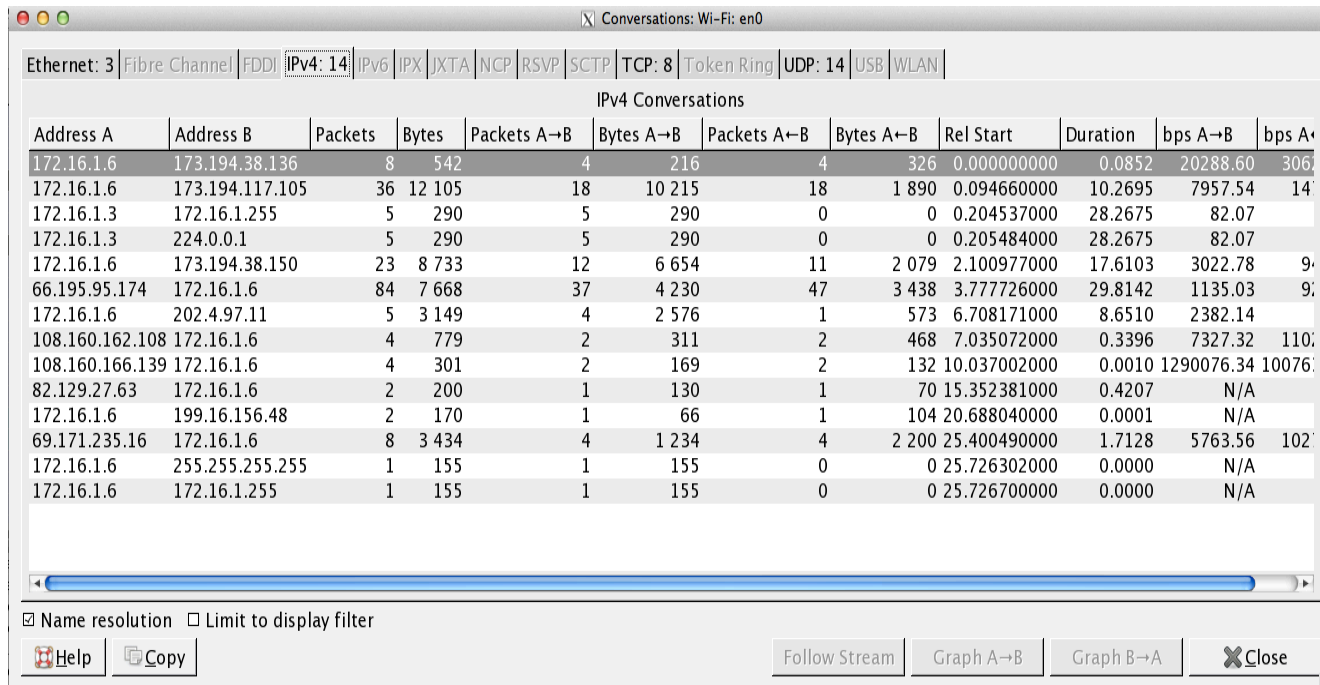
Display filter: none

| 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 |

Help Close

Use “Statistics”

- Which host most chatty
 - Statistics -> Conversations



The screenshot shows a window titled "Conversations: Wi-Fi: en0" with a tab for "IPv4: 14". The main content is a table of IPv4 Conversations. The table has columns for Address A, Address B, Packets, Bytes, Packets A→B, Bytes A→B, Packets A←B, Bytes A←B, Rel Start, Duration, bps A→B, and bps A←B. The data is as follows:

| Address A | Address B | Packets | Bytes | Packets A→B | Bytes A→B | Packets A←B | Bytes A←B | Rel Start | Duration | bps A→B | bps A←B |
|-----------------|-----------------|---------|--------|-------------|-----------|-------------|-----------|--------------|----------|------------|-----------|
| 172.16.1.6 | 173.194.38.136 | 8 | 542 | 4 | 216 | 4 | 326 | 0.000000000 | 0.0852 | 20288.60 | 306.0 |
| 172.16.1.6 | 173.194.117.105 | 36 | 12 105 | 18 | 10 215 | 18 | 1 890 | 0.094660000 | 10.2695 | 7957.54 | 14 000 |
| 172.16.1.3 | 172.16.1.255 | 5 | 290 | 5 | 290 | 0 | 0 | 0.204537000 | 28.2675 | 82.07 | |
| 172.16.1.3 | 224.0.0.1 | 5 | 290 | 5 | 290 | 0 | 0 | 0.205484000 | 28.2675 | 82.07 | |
| 172.16.1.6 | 173.194.38.150 | 23 | 8 733 | 12 | 6 654 | 11 | 2 079 | 2.100977000 | 17.6103 | 3022.78 | 9 000 |
| 66.195.95.174 | 172.16.1.6 | 84 | 7 668 | 37 | 4 230 | 47 | 3 438 | 3.777726000 | 29.8142 | 1135.03 | 9 000 |
| 172.16.1.6 | 202.4.97.11 | 5 | 3 149 | 4 | 2 576 | 1 | 573 | 6.708171000 | 8.6510 | 2382.14 | |
| 108.160.162.108 | 172.16.1.6 | 4 | 779 | 2 | 311 | 2 | 468 | 7.035072000 | 0.3396 | 7327.32 | 110 000 |
| 108.160.166.139 | 172.16.1.6 | 4 | 301 | 2 | 169 | 2 | 132 | 10.037002000 | 0.0010 | 1290076.34 | 10076 000 |
| 82.129.27.63 | 172.16.1.6 | 2 | 200 | 1 | 130 | 1 | 70 | 15.352381000 | 0.4207 | N/A | |
| 172.16.1.6 | 199.16.156.48 | 2 | 170 | 1 | 66 | 1 | 104 | 20.688040000 | 0.0001 | N/A | |
| 69.171.235.16 | 172.16.1.6 | 8 | 3 434 | 4 | 1 234 | 4 | 2 200 | 25.400490000 | 1.7128 | 5763.56 | 102 000 |
| 172.16.1.6 | 255.255.255.255 | 1 | 155 | 1 | 155 | 0 | 0 | 25.726302000 | 0.0000 | N/A | |
| 172.16.1.6 | 172.16.1.255 | 1 | 155 | 1 | 155 | 0 | 0 | 25.726700000 | 0.0000 | N/A | |

At the bottom of the window, there are checkboxes for "Name resolution" (checked) and "Limit to display filter" (unchecked). There are also buttons for "Help", "Copy", "Follow Stream", "Graph A→B", "Graph B→A", and "Close".

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
- Run wireshark and Capture inbound/outbound traffic
- Download capture files from
 - Follow the instructor's guide.

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!!

LAB