



Michael Takeuchi

# Practical Packet Analysis for Network Incident Response with Mikrotik RouterOS

25 October 2019, Kuta Bali  
Mikrotik User Meeting Indonesia



# Hello, I am Michael Takeuchi

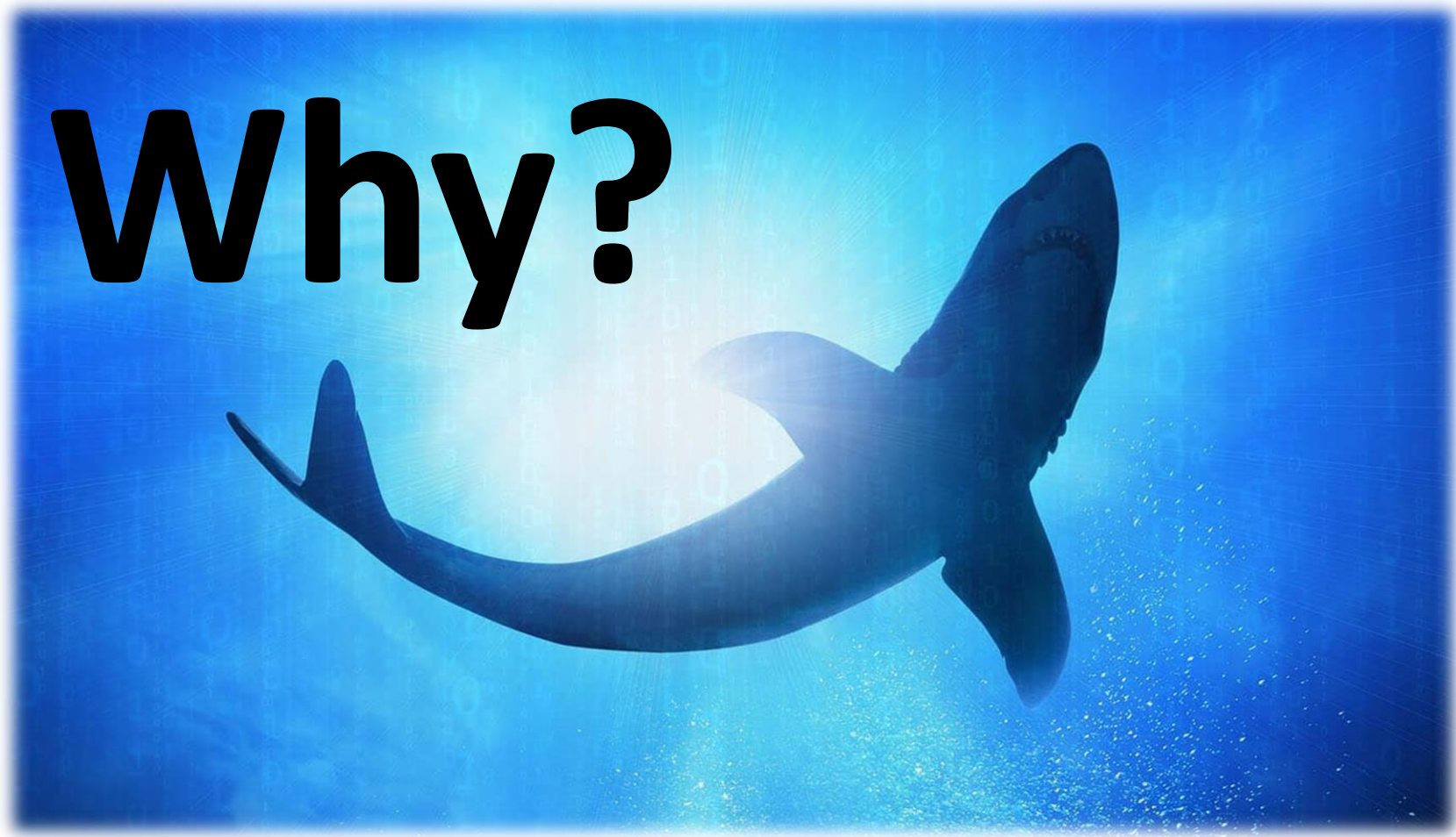
MikroTik Certified Engineer & Consultant from Jakarta, Indonesia

 <https://www.linkedin.com/in/michael-takeuchi>

 <https://www.facebook.com/mict404>

 [michael@takeuchi.id](mailto:michael@takeuchi.id)

# Why Packet Analysis?



# Why Packet Analysis?

- Information of 5W + 1H
  - What
    - DDoS? Spam? Flood?
  - Who
    - Router? PC? Server?
  - When
    - Now? Yesterday?
  - Where
    - AS? Network?
  - Why
    - Virus?
  - How
    - TCP? UDP?
- Action/Decision
  - Fix
  - Stop
  - Deny

# Who do Packet Analysis?

- Researchers: Access to RAW Data
- Administrator: Debugging Network Problems
- Analyst: Analyze the Traffic
- Incident Responders: Tracing the Incident

# How We Do Packet Analysis?

CAPTURE

&

ANALYZE

# Capturing Packets

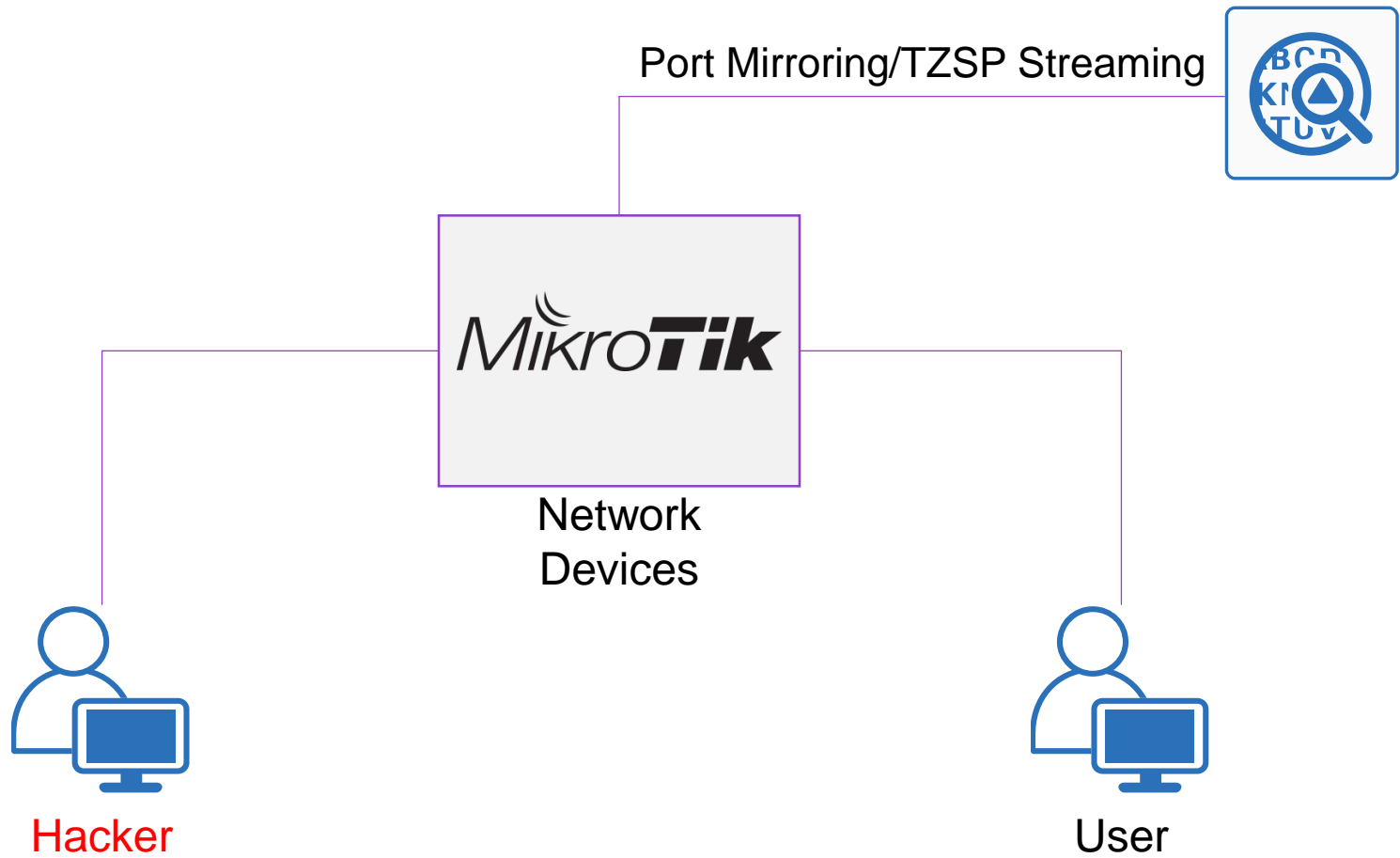
- Also known as **SNIFFING**
- PCAP is the common format of **Packet Capture**
- Perspective is Important
  - In-band
  - Out-band

# In-Band Capturing Packets/Sniffing

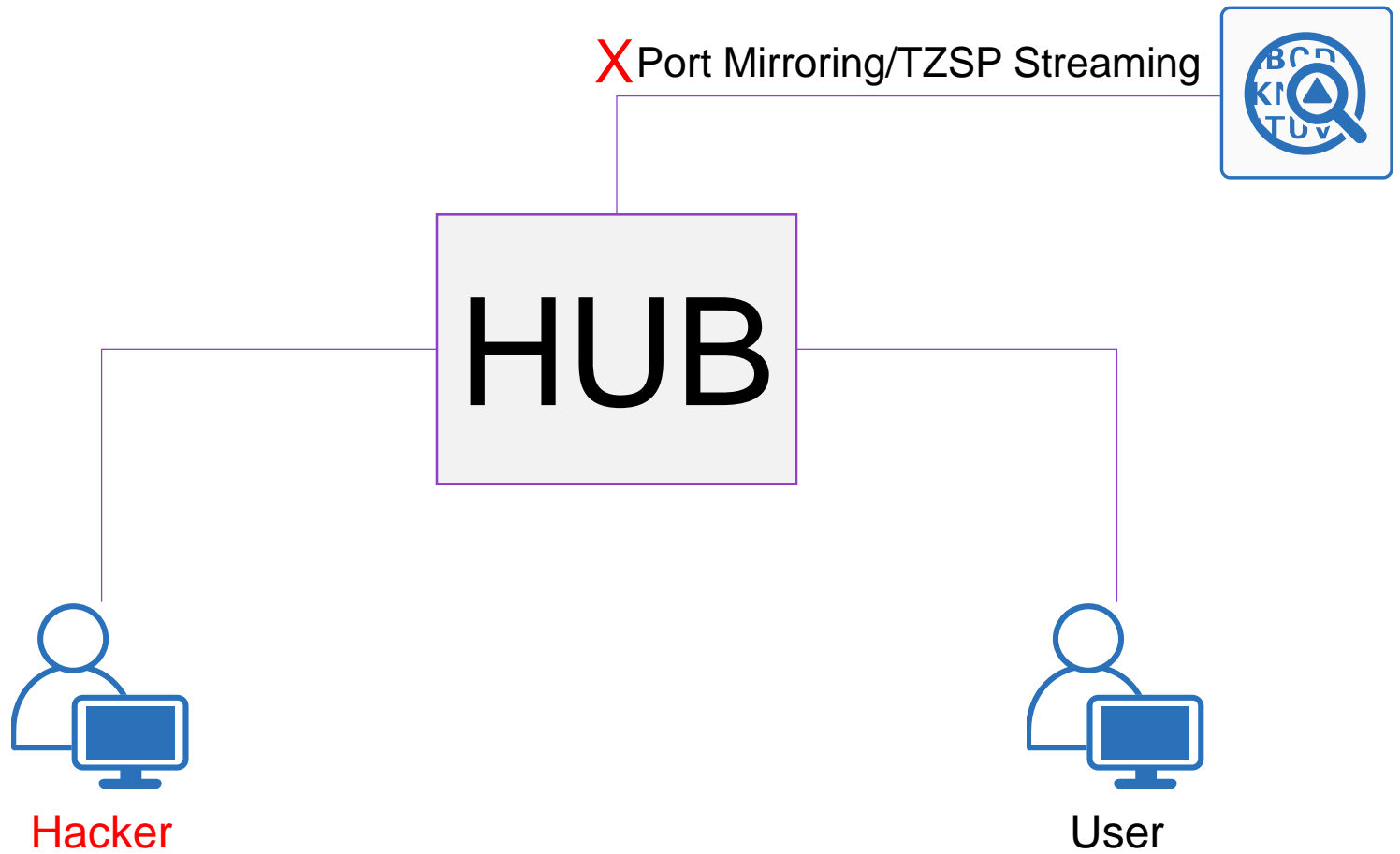




# Out-Band Capturing Packets/Sniffing



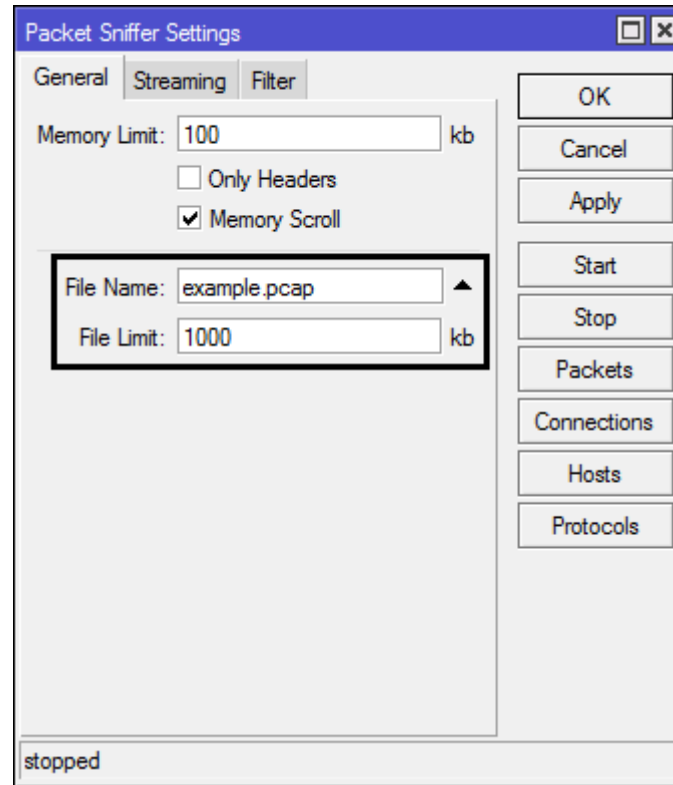
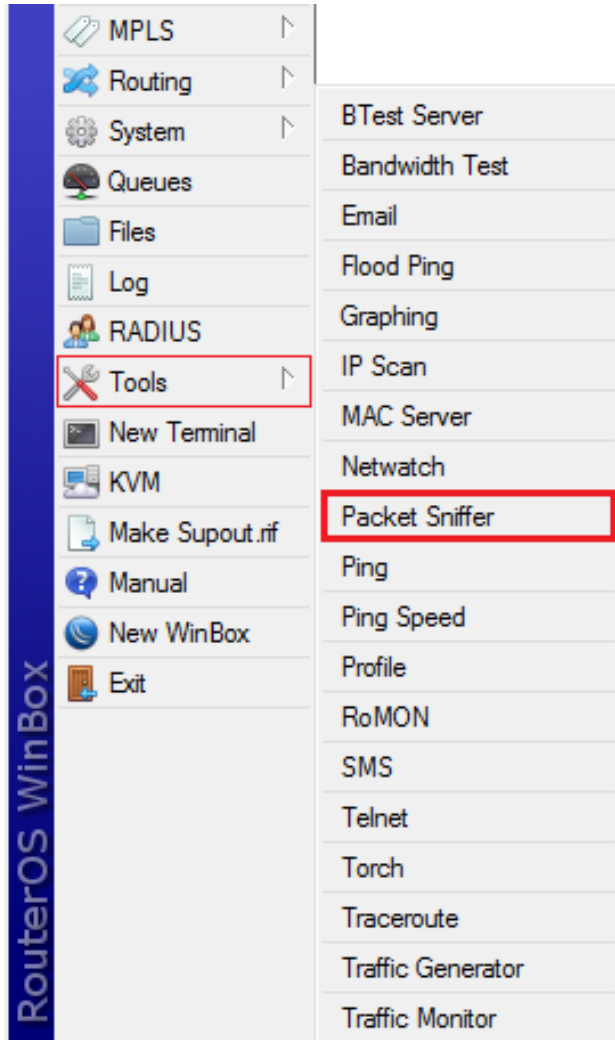
# Out-Band Capturing Packets/Sniffing



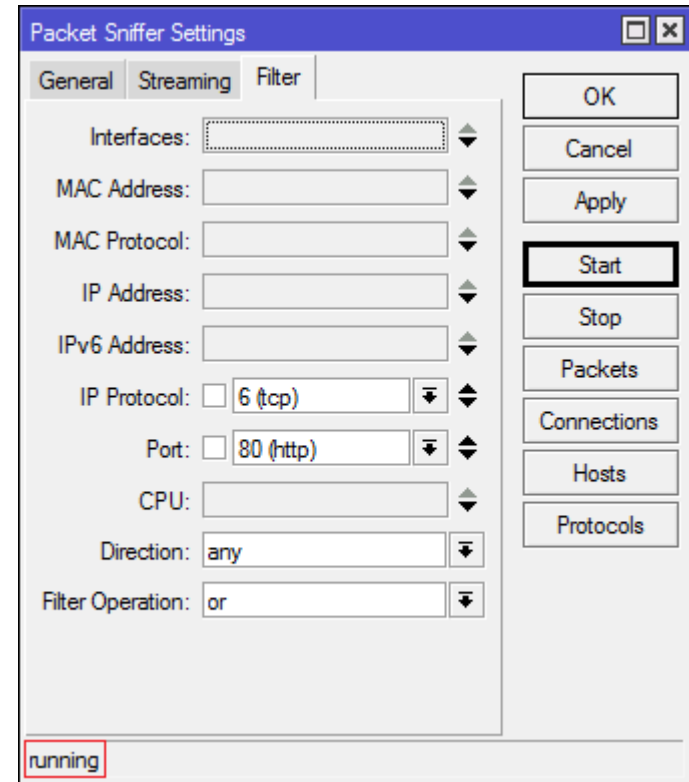
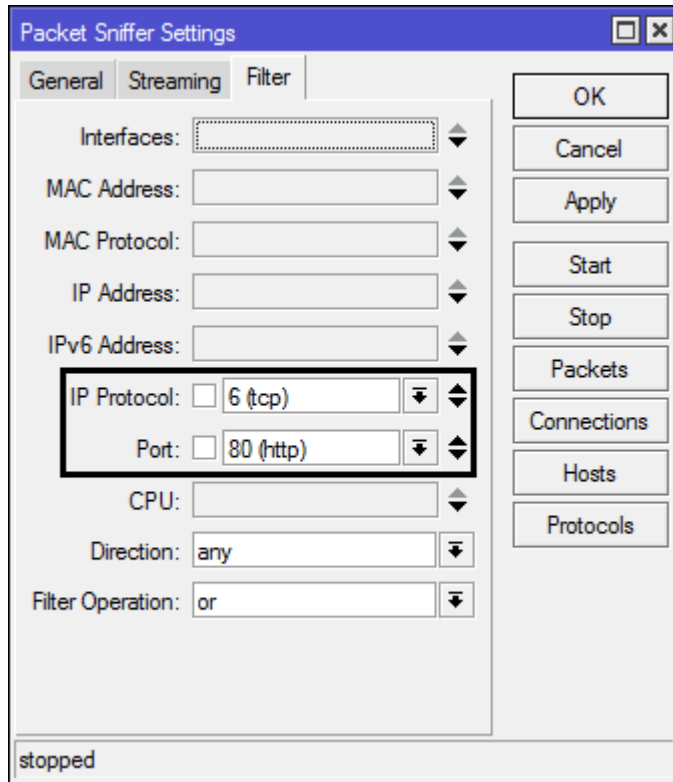
# Capturing Packets in MikroTik – HTTP

```
/tool sniffer  
set file-name="example.pcap"  
set file-limit="1000"  
set filter-ip-protocol="tcp"  
set filter-port="80"  
start  
/file print where name="example.pcap"
```

# Capturing Packets in MikroTik – HTTP



# Capturing Packets in MikroTik – HTTP



```
[takeuchi@MikroTik] > file print where name="example.pcap"
# NAME                TYPE                SIZE
0 example.pcap        .pcap file          1000.2KiB
```

# Capturing Packets in MikroTik – Storage Expense

**Expense storage quickly!!!**

○  $10\text{Mbps} * 3600 \text{ (second)} * 24 \text{ (hours)} = 864000\text{Mb}$

○  $864000\text{Mb} / 8 = 108000 \text{ Megabyte for 1 Day}$

10Mbps Bandwidth need 100+ Gigabyte storage for 1 Day

Double for full-duplex (200+ Gigabyte)

**How big is your storage?**

Solution? Use Out-Band Capturing Packets/Sniffing method with Port Mirroring, TZSP Streaming or use HUB

# Capturing Packets in MikroTik – Port Mirroring

- Port Mirroring is Switch Chip Feature
- MikroTik devices without switch chip can't do Port Mirroring

```
/interface ethernet switch  
set switch1 mirror-source=ether2  
set switch1 mirror-target=ether3
```

# Capturing Packets in MikroTik – TZSP Configuration

```
/tool sniffer  
streaming-server=ip.of.wireshark.box  
set streaming-enabled=yes  
start
```

## Capture

...using this filter:

Local Area Connection

Wireless Network Connection

TZSP is run on UDP/37008, you can listen on UDP/37008 with your sniffing tools like **wireshark** (will introduced more in analyze step)



# Capturing Packets in MikroTik – TZSP Configuration (Alt.)

```
/ip firewall mangle  
add action=sniff-tzsp chain=prerouting  
sniff-target=ip.of.wireshark.box  
sniff-target-port=port.of.wireshark.box
```

## Capture

...using this filter:

Local Area Connection

Wireless Network Connection

By default TZSP is run on UDP/37008, so you can listen on UDP/37008 with your sniffing tools like **wireshark** (will introduce wireshark more in analyze step)

# Capturing Packets in MikroTik – Done

Are you done?

Let's continue to analyze the PCAP!

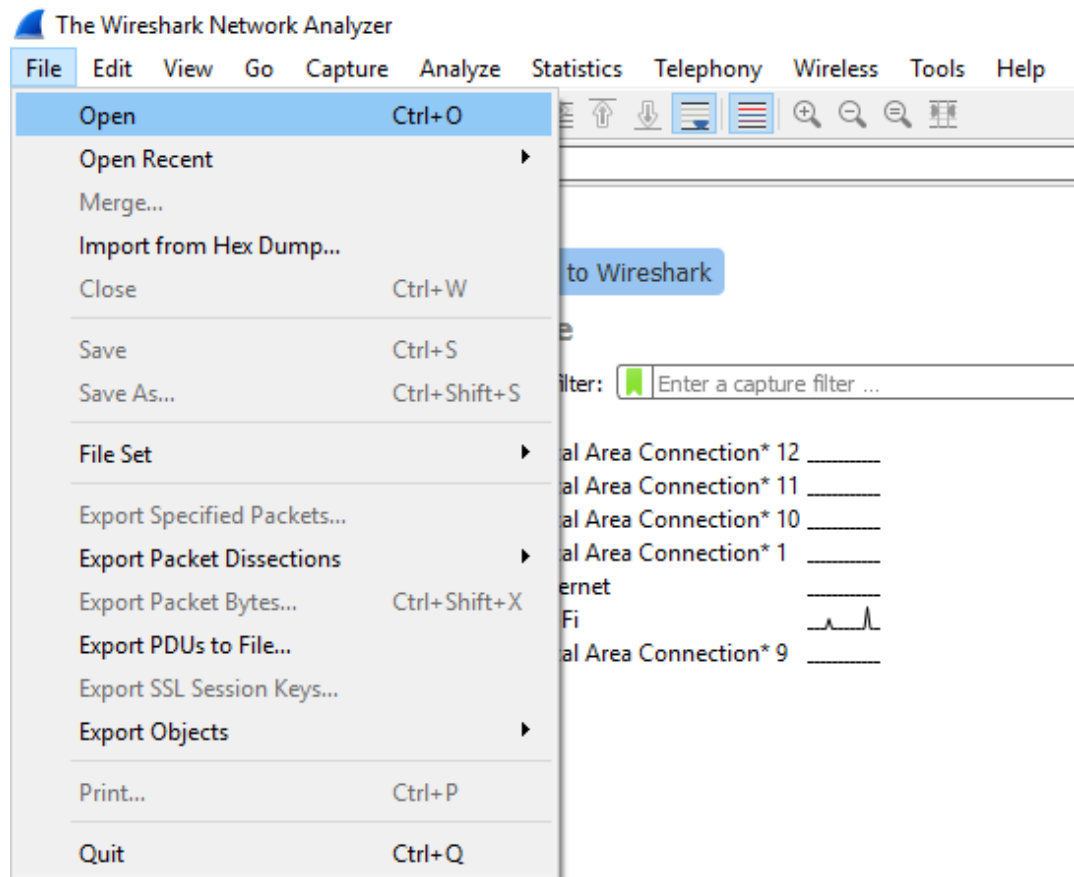
# Analyzing Packets – Fire On The Tools

Fire on your tools:

- **Wireshark**
  - Open Source (GNU Public License)
  - Multi-Platform (Windows, Linux, \*BSD & MacOS)
  - Advanced Filtering & Analyzing
  - Used for Live Sniffing & Packet Analysis
- Some people use **Wireshark** for:
  - Network Administrators: troubleshoot network problems
  - Network Security Engineers: examine security problems
  - Developers: debug protocol implementations
  - Peoples: learn network protocol internals

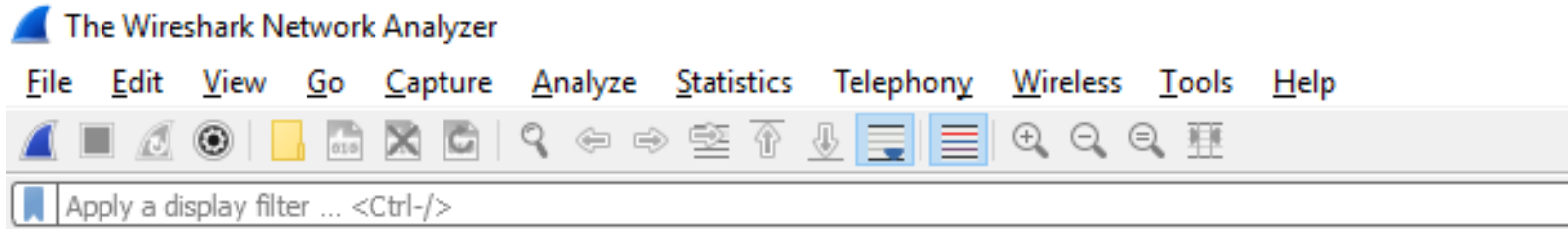
# Analyzing Packets – Getting Started with Wireshark

- To getting started with wireshark you can open the pcap file that you have from capturing packets



# Analyzing Packets – Getting Started with Wireshark

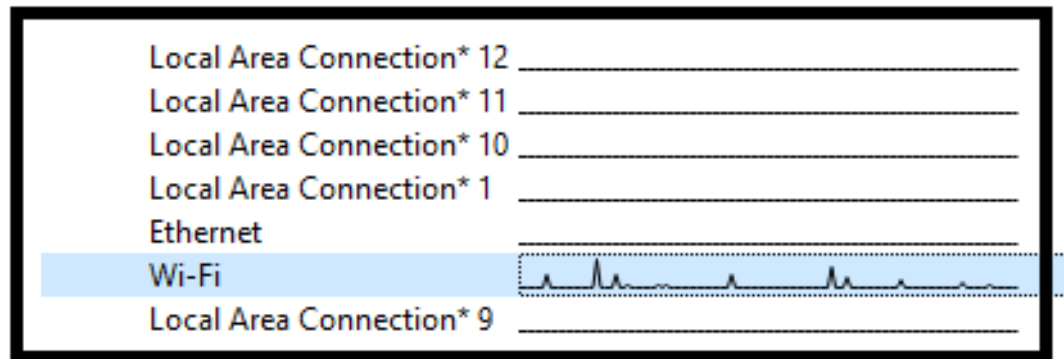
○ Or you can capture the new packets 😊



Welcome to Wireshark

## Capture

...using this filter:



# Analyzing Packets – Wireshark Interfaces

The screenshot displays the Wireshark network protocol analyzer interface. The top pane, 'Packet List', shows a list of captured packets with columns for No., Time, Source, Destination, Protocol, Length, and Info. The second pane, 'Packet Details', shows the hierarchical structure of the selected packet (No. 2171), including Ethernet II, Internet Protocol Version 4, and Transmission Control Protocol. The third pane, 'Packet Bytes', shows the raw hexadecimal and ASCII data of the selected packet. The status bar at the bottom indicates 'Packets: 2285 · Displayed: 945 (41.4%) · Dropped: 0 (0.0%)' and 'Profile: Default'.

No.	Time	Source	Destination	Protocol	Length	Info
2157	30.098742	172.16.50.60	172.16.50.1	TCP	118	38559 → 8291 [PSH, ACK] Seq=6946 Ack=342772 Win=4106 Len=64
2158	30.100370	172.16.50.1	172.16.50.60	TCP	137	8291 → 38559 [PSH, ACK] Seq=342772 Ack=7010 Win=501 Len=83
2159	30.112760	172.16.50.60	172.16.50.1	TCP	115	38559 → 8291 [PSH, ACK] Seq=7010 Ack=342855 Win=4105 Len=61
2160	30.114232	172.16.50.1	172.16.50.60	TCP	121	8291 → 38559 [PSH, ACK] Seq=342855 Ack=7071 Win=501 Len=67
2161	30.125736	172.16.50.60	172.16.50.1	TCP	115	38559 → 8291 [PSH, ACK] Seq=7071 Ack=342922 Win=4105 Len=61
2162	30.127310	172.16.50.1	172.16.50.60	TCP	121	8291 → 38559 [PSH, ACK] Seq=342922 Ack=7132 Win=501 Len=67
2163	30.167254	172.16.50.60	172.16.50.1	TCP	54	38559 → 8291 [ACK] Seq=7132 Ack=342989 Win=4105 Len=0
2164	30.231886	157.240.24.20	172.16.50.60	TLSv1.2	185	Application Data
2165	30.238707	172.16.50.60	157.240.24.20	TLSv1.2	676	Ignored Unknown Record
2166	30.254685	157.240.24.20	172.16.50.60	TCP	60	443 → 38611 [ACK] Seq=132 Ack=624 Win=362 Len=0
2167	30.255217	157.240.24.20	172.16.50.60	TLSv1.2	89	Application Data
2168	30.295106	172.16.50.60	157.240.24.20	TCP	54	38611 → 443 [ACK] Seq=624 Ack=167 Win=507 Len=0
2169	30.508962	172.16.50.1	172.16.50.60	TCP	922	8291 → 38559 [PSH, ACK] Seq=342989 Ack=7132 Win=501 Len=868
2170	30.620099	172.16.50.60	172.16.50.1	TCP	126	38559 → 8291 [PSH, ACK] Seq=7132 Ack=343857 Win=4102 Len=72
2171	30.632300	172.16.50.1	172.16.50.60	TCP	1514	8291 → 38559 [ACK] Seq=343857 Ack=7204 Win=501 Len=1460
2172	30.632793	172.16.50.1	172.16.50.60	TCP	1514	8291 → 38559 [ACK] Seq=345317 Ack=7204 Win=501 Len=1460
2173	30.632794	172.16.50.1	172.16.50.60	TCP	1514	8291 → 38559 [ACK] Seq=346777 Ack=7204 Win=501 Len=1460
2174	30.632794	172.16.50.1	172.16.50.60	TCP	1514	8291 → 38559 [ACK] Seq=348237 Ack=7204 Win=501 Len=1460
2175	30.632794	172.16.50.1	172.16.50.60	TCP	1514	8291 → 38559 [ACK] Seq=349697 Ack=7204 Win=501 Len=1460
2176	30.632795	172.16.50.1	172.16.50.60	TCP	1514	8291 → 38559 [ACK] Seq=351157 Ack=7204 Win=501 Len=1460
2177	30.632795	172.16.50.1	172.16.50.60	TCP	1514	8291 → 38559 [ACK] Seq=352617 Ack=7204 Win=501 Len=1460
2178	30.632795	172.16.50.1	172.16.50.60	TCP	207	8291 → 38559 [PSH, ACK] Seq=354077 Ack=7204 Win=501 Len=153
2179	30.632834	172.16.50.60	172.16.50.1	TCP	54	38559 → 8291 [ACK] Seq=7204 Ack=354230 Win=4106 Len=0
2180	30.655949	172.16.50.60	172.16.50.1	TCP	123	38559 → 8291 [PSH, ACK] Seq=7204 Ack=354230 Win=4106 Len=69

Frame 2171: 1514 bytes on wire (12112 bits), 1514 bytes captured (12112 bits) on interface 0  
Ethernet II, Src: Intel\_f0:34:bf (00:90:27:f0:34:bf), Dst: IntelCor\_a8:b3:76 (20:79:18:a8:b3:76)  
Internet Protocol Version 4, Src: 172.16.50.1, Dst: 172.16.50.60  
Transmission Control Protocol, Src Port: 8291, Dst Port: 38559, Seq: 343857, Ack: 7204, Len: 1460  
Data (1460 bytes)

```
0000  20 79 18 a8 b3 76 00 90 27 f0 34 bf 08 00 45 00  y...v...4...E-
0010  05 dc 18 16 40 00 40 06 60 a8 ac 10 32 01 ac 10  ...@...2...
0020  32 3c 20 63 96 9f 95 0d be a2 66 81 51 21 50 10  2<c...@...fQ!P...
0030  01 f5 1f 77 00 00 ff 05 28 27 bc da c9 a2 e0 c0  .....(.....
0040  b1 72 c7 72 04 f5 6b f2 26 79 9b 4b d5 db 5b 2a  ..m..k..8y..K..[*
0050  af 3b 21 b2 18 c3 b0 6f c0 b0 b4 e1 68 d9 0c 67  :!...o...h:g
0060  9a 38 21 01 78 51 53 7e 08 83 21 e8 48 6d ba 34  -8!xQ$w...!..Hm-4
0070  52 b4 5d ed bf 41 37 46 9e 3c a4 5b fd 22 59 3a  R.]..A7F<[...Y:
0080  d9 3f 82 3d c0 13 63 66 85 bc 24 7e ce d3 3e a5  ?...=cf...$->
0090  2c 01 0a 50 12 de 3d fc fc c9 3a 74 18 20 77 6d  :P...m...t:..vm
00a0  88 af 02 e9 31 f7 b6 98 4b 67 98 d5 23 f1 67 9b  ...1...Kg...#g:
00b0  5b d3 18 e6 e9 b9 4b 89 06 6a 46 04 b0 c8 06 ed  [......k...fjP...
00c0  24 3d db de 67 88 18 e9 27 fe ae 6b d8 f4 81 d2  $=g...<...f:G...
00d0  e7 64 2e bf 52 20 69 18 4a 20 0c 54 56 d3 37 aa  -d..R..j...TV-?
00e0  70 83 90 af 54 69 1f 31 83 5b c9 fb fe 28 b0  p...Ti:1:[...(:
00f0  3c 39 3f 5e b6 0b 2c 06 0b 03 65 cb 09 93 72 4e  <9?>...<...e...rN
0100  2f 06 f5 b4 5f b6 78 81 0a 60 08 00 4c 41 8a 5a  /...x...LA-Z
0110  df c3 a0 d3 d2 d4 6d 6b ff d8 22 ff 35 56 fe 19  ...mk...".SV...
0120  41 5c 39 e5 cf 27 f2 10 89 57 21 8f 3b bd 47 1f  A@>...W!;:6:
0130  a7 7f 24 ff db a9 a1 ff ff 04 c3 cf 9d 47 0c c3  $=g...<...f:G...
0140  6e ec 41 15 9e 5f 89 32 3c bc a5 71 f3 2c 53  n.A...<...l...S
0150  ba a1 ec c4 61 d5 f6 95 b3 1a 2b 66 cf 11 09 b6  $...<...+f...</pre>
```

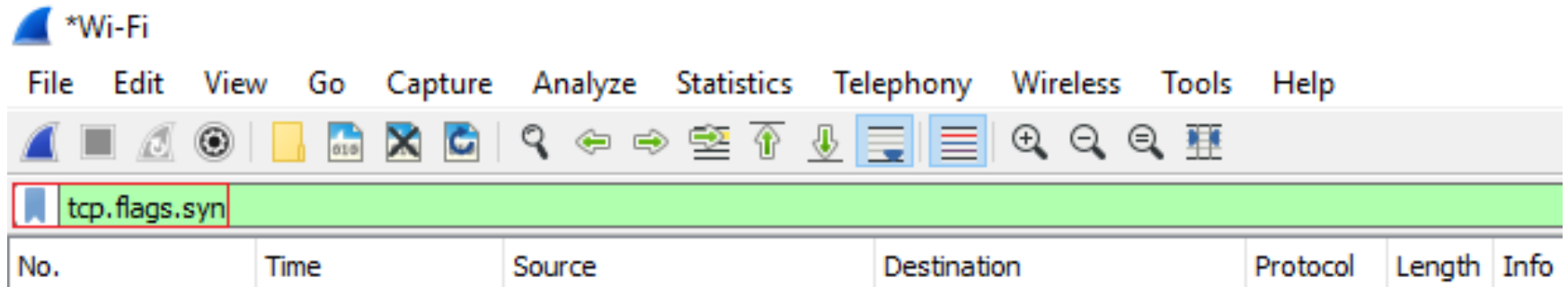
Packet List

Packet Details

Packet Bytes

# Analyzing Packets – Packet Filtering

- We can filter specific packet type in wireshark



- You can check the cheat sheet on

[http://packetlife.net/media/library/13/Wireshark\\_Display\\_Filters.pdf](http://packetlife.net/media/library/13/Wireshark_Display_Filters.pdf)

# Analyzing Packets – Fetching a Messages

The screenshot displays the Wireshark interface for analyzing network traffic. The main window shows a list of captured packets, with the selected packet (No. 249) highlighted in blue. A context menu is open over this packet, showing various actions such as 'Mark/Unmark Packet', 'Ignore/Unignore Packet', 'Set/Unset Time Reference', 'Time Shift...', 'Packet Comment...', 'Edit Resolved Name', 'Apply as Filter', 'Prepare a Filter', 'Conversation Filter', 'Colorize Conversation', 'SCTP', 'Follow', 'Copy', 'Protocol Preferences', 'Decode As...', and 'Show Packet in New Window'. The 'Follow' option is expanded, showing 'TCP Stream' (Ctrl+Alt+Shift+T), 'UDP Stream' (Ctrl+Alt+Shift+U), 'SSL Stream' (Ctrl+Alt+Shift+S), and 'HTTP Stream' (Ctrl+Alt+Shift+H).

The packet list table is as follows:

No.	Time	Source	Destination	Protocol	Length	Info
249	3.518943	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
250	3.511042	222.124.12.125	.96.138	SMTP	290	C: DATA fragment, 224 bytes
252	3.511254	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
263	3.513480	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
264	3.513671	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
265	3.513789	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
266	3.513898	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
267	3.514006	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
277	3.519622	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
279	3.519912	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
280	3.520058	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
283	3.520297	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
285	3.520546	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
286	3.520660	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
293	3.522530	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
297	3.523053	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
299	3.523323	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
300	3.523438	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
301	3.523550	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
302	3.523659	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
303	3.523768	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
311	3.529933	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
312	3.530173	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes
313	3.530306	222.124.12.125	.96.138	SMTP	1314	C: DATA fragment, 1248 bytes

The packet details pane shows the following information for the selected packet:

- > Frame 249: 1314 bytes on wire (10512 bits), 1314 bytes captured (10512 bits)
- > Ethernet II, Src: Routerbo\_15:2d:41 (4c:5e:0c:15:2d:41), Dst: Routerbo\_60:7d:d2 (4c:5e:0c:60:7d:d2)
- > Internet Protocol Version 4, Src: 222.124.12.125, Dst: .96.138
- > Transmission Control Protocol, Src Port: 47076, Dst Port: 25, Seq: 132743, Ack: 183, Len: 1248
- > Simple Mail Transfer Protocol

The packet bytes pane shows the raw data in hexadecimal and ASCII:

```
0000 4c 5e 0c 60 7d d2 4c 5e 0c 15 2d 41 08 00 45 00  L^...}.L^...A..E-
0010 05 14 e0 7e 40 00 36 06 81 1d de 7c 0c 7d 92 c4  ....@6...|}...
0020 60 8a b7 e4 00 19 0c e4  d4 b5 13 e6 44 db 80 10  ....D...
0030 00 63 6c 4a 00 00 01 01 08 0a d0 0b 3b de 00 70  ...c|].....p
0040 ff 2f 2f 33 71 74 75 76 35 55 39 31 37 2f 37 72  //3qtuv 5U9177r
0050 58 2f 76 36 70 2f 66 50 35 54 72 53 6e 2f 39 6c  X/v6p/fp 5Tr5n/9l
0060 51 75 2b 30 76 53 37 71 31 39 62 32 32 36 2f 0d  Qu0v57q 19b226//
0070 0a 36 64 6c 58 37 53 39 66 34 37 70 2f 2f 33 58  -6d1X759 f47p//3X
0080 37 37 57 4b 39 77 51 4c 72 6a 36 2f 59 2f 62 31  77mK9wQL rj6/v/b1
0090 72 58 78 57 33 73 56 78 57 78 73 56 58 54 37 73  rXwX3svXv WxsvX7Ts
00a0 6e 5a 4a 63 31 38 0b 50 32 70 4c 78 58 46 53 55  n2Zc18Kp 2pLXfSU
00b0 45 33 74 53 4e 39 2f 2f 4a 65 76 39 69 00 0a 6e  E3tSN9// Jev91-n
00c0 6d 6d 35 47 51 5a 56 39 39 53 33 37 2b 2b 6c 39  km5G1Z9v 9Q37+19
00d0 52 45 52 45 52 45 52 45 52 45 52 45 52 78 45 52  RERERERER RERERxER
00e0 45 52 45 52 45 52 45 52 48 45 52 6c 4f 45 77 68  RERERERER HERIOEwh
00f0 45 52 44 43 47 43 49 2f 73 56 46 62 39 50 2f 49  ERDCCGI/ sVFb9P/I
0100 33 33 79 4d 63 6a 63 66 78 45 52 0d 0a 45 63 52  33yMcjcf xER -EcR
0110 45 52 45 52 45 52 45 52 45 52 2f 2f 2f 2f 2f 2f  ERERERERER ER/////
0120 2f 2f 2f 2f 2f 2f 2f 2f 4a 41 4c 6e 5a 71 43 2f  /////// JALNzqC/
0130 2b 56 70 6c 61 66 2f 2f 2f 3b 2b 6a 69 4c 6b 59  +Vp1st// /e+LlLY
0140 50 7a 36 4f 49 32 2f 2f 32 73 71 5a 55 50 37 30  r260I// 2sqUP70
0150 56 5a 53 6e 2f 37 61 58 6a 0d 0a 39 36 56 2f 2f  VZ5n/7ax j-96V//
```



# Analyzing Packets – Fetching a Messages

```
220 mail. ....com ESMTTP
EHLO mail. ....com
250-mail. ....com
250-SIZE 2048000
250-AUTH LOGIN
250 HELP
MAIL From:<Dery@...com> SIZE=3087509
250 OK
RCPT To:<wijey@...com>
250 OK
RCPT To:<sari@...com>
250 OK
RCPT To:<operational_...@...com>
250 OK
RCPT To:<jenny@...com>
250 OK
RCPT To:<import4@...com>
250 OK
RCPT To:<import3@...com>
250 OK
RCPT To:<import@...com>
250 OK
DATA
354 OK, send.
Received: from BPGDOM01. ....COM (webmail. ....com [10.60.0.6])
by mail. ....com (8.14.4/8.14.4) with ESMTTP id w63Ahkw027114;
Tue, 3 Jul 2018 17:43:51 +0700
In-Reply-To: <CADuhDSCbu7R0k4LNRuhuXVpqb-GXVMjMX2pP0NifhyZwMy04LQ@mail.gmail.com>
References: <02e701d412ae596edbfce0$c4c93f40$@...com> <CADuhDSCbu7R0k4LNRuhuXVpqb-
GXVMjMX2pP0NifhyZwMy04LQ@mail.gmail.com>
To:
Cc:
```

○ Now we got a messages from email 😊 and now we can analyze the email

# Analyzing Packets – Exporting Object (PDF, JPG, PNG, etc.)

The screenshot shows the Wireshark interface with the File menu open. The 'Export Objects' option is selected, and a sub-menu is visible with options: DICOM..., HTTP..., IMF..., SMB..., and TFTP... The main packet list shows a series of SMTP data fragments. Below the list, the packet details pane shows the following information:

- > Frame 57: 1314 bytes on wire (10512 bits), 1314 bytes captured (10512 bits)
- > Ethernet II, Src: Routerbo\_15:2d:41 (4c:5e:0c:15:2d:41), Dst: Routerbo\_60:7d:d2 (4c:5e:0c:60:7d:d2)
- > Internet Protocol Version 4, Src: 222.124.12.125, Dst: [REDACTED].96.138
- > Transmission Control Protocol, Src Port: 47076, Dst Port: 25, Seq: 327, Ack: 183, Len: 1248
- > Simple Mail Transfer Protocol

# Analyzing Packets – Flood Example (DNS)

udp53.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

dns

No.	Time	Source	Destination	Protocol	Length	Info
6321	32.997895	[REDACTED]	.68.170	8.8.4.4	DNS	74 Standard query 0xb6d3 A m9.tthbbre.com
2336	9.379658	[REDACTED]	.68.170	8.8.4.4	DNS	74 Standard query 0xb6d3 A m6.rsxbjn.com
2335	9.379443	[REDACTED]	.68.170	8.8.4.4	DNS	74 Standard query 0xb6d3 A m6.rsxbjn.com
856	3.542306	[REDACTED]	.68.170	8.8.4.4	DNS	75 Standard query 0xb676 A m36.qjbyfio.com
855	3.542067	[REDACTED]	.68.170	8.8.4.4	DNS	75 Standard query 0xb676 A m36.qjbyfio.com
2827	11.392291	[REDACTED]	.68.170	8.8.4.4	DNS	75 Standard query 0xb669 A m10.qjwhpfe.net
2826	11.392077	[REDACTED]	.68.170	8.8.4.4	DNS	75 Standard query 0xb669 A m10.qjwhpfe.net
3397	13.700797	[REDACTED]	.68.170	8.8.4.4	DNS	75 Standard query 0xb607 A m28.pgngxmo.com
3396	13.700657	[REDACTED]	.68.170	8.8.4.4	DNS	75 Standard query 0xb607 A m28.pgngxmo.com
3173	13.030748	[REDACTED]	.68.170	8.8.4.4	DNS	75 Standard query 0xb603 A m38.ddjctyf.biz
3172	13.030574	[REDACTED]	.68.170	8.8.4.4	DNS	75 Standard query 0xb603 A m38.ddjctyf.biz
662	2.620854	[REDACTED]	.68.170	8.8.4.4	DNS	75 Standard query 0xb5c0 A m31.swwgocu.net
661	2.620628	[REDACTED]	.68.170	8.8.4.4	DNS	75 Standard query 0xb5c0 A m31.swwgocu.net
4996	26.914537	[REDACTED]	.68.170	8.8.4.4	DNS	74 Standard query 0xb5be A m7.brmgkod.com
4995	26.914357	[REDACTED]	.68.170	8.8.4.4	DNS	74 Standard query 0xb5be A m7.brmgkod.com
7936	40.211405	[REDACTED]	.68.170	8.8.4.4	DNS	74 Standard query 0xb596 A m9.bnwqcxl.com
7935	40.211193	[REDACTED]	.68.170	8.8.4.4	DNS	74 Standard query 0xb596 A m9.bnwqcxl.com
6961	35.918350	[REDACTED]	.68.170	8.8.4.4	DNS	75 Standard query 0xb588 A m30.atoifmo.com
6960	35.918120	[REDACTED]	.68.170	8.8.4.4	DNS	75 Standard query 0xb588 A m30.atoifmo.com
7059	36.405497	[REDACTED]	.68.170	8.8.4.4	DNS	74 Standard query 0xb54d A m16.plswhql.cc
7058	36.405310	[REDACTED]	.68.170	8.8.4.4	DNS	74 Standard query 0xb54d A m16.plswhql.cc
134	0.488968	[REDACTED]	.68.170	8.8.4.4	DNS	75 Standard query 0xb51b A m30.shbqnoe.net
133	0.488805	[REDACTED]	.68.170	8.8.4.4	DNS	75 Standard query 0xb51b A m30.shbqnoe.net
1256	5.029502	[REDACTED]	.68.170	8.8.4.4	DNS	75 Standard query 0xb51a A m31.gumqkle.net

# Analyzing Packets – Flood Example (TELNET)

hajime-np.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.stream eq 3

No.	Time	Source	Destination	Protocol	Length	Info
11	0.098194	.117.48	.69.50	TELNET	95	Telnet Data ...
12	0.098314	.117.48	.69.50	TCP	95	[TCP Retransmission] 23 → 54014 [PSH, ACK] Seq=1 Ack=1 Win=4078 Len=41
13	0.098625	.69.50	.117.48	TELNET	61	Telnet Data ...
14	0.098692	.69.50	.117.48	TCP	61	[TCP Retransmission] 54014 → 23 [PSH, ACK] Seq=1 Ack=42 Win=14600 Len=7
15	0.103057	.117.48	.69.50	TELNET	60	Telnet Data ...
16	0.103112	.117.48	.69.50	TCP	55	[TCP Keep-Alive] 23 → 54014 [PSH, ACK] Seq=42 Ack=8 Win=4071 Len=1
17	0.103438	.117.48	.69.50	TELNET	60	Telnet Data ...
18	0.103492	.117.48	.69.50	TCP	55	[TCP Keep-Alive] 23 → 54014 [PSH, ACK] Seq=43 Ack=8 Win=4071 Len=1
19	0.103521	.117.48	.69.50	TELNET	60	Telnet Data ...
20	0.103570	.117.48	.69.50	TCP	55	[TCP Keep-Alive] 23 → 54014 [PSH, ACK] Seq=44 Ack=8 Win=4071 Len=1
21	0.103597	.117.48	.69.50	TELNET	60	Telnet Data ...
22	0.103644	.117.48	.69.50	TCP	55	[TCP Keep-Alive] 23 → 54014 [PSH, ACK] Seq=45 Ack=8 Win=4071 Len=1
23	0.103657	.117.48	.69.50	TELNET	60	Telnet Data ...
24	0.103689	.117.48	.69.50	TCP	55	[TCP Keep-Alive] 23 → 54014 [PSH, ACK] Seq=46 Ack=8 Win=4071 Len=1
25	0.103711	.117.48	.69.50	TELNET	66	Telnet Data ...
26	0.103753	.117.48	.69.50	TCP	66	[TCP Retransmission] 23 → 54014 [PSH, ACK] Seq=47 Ack=8 Win=4071 Len=22
27	0.103911	.69.50	.117.48	TCP	69	54014 → 23 [ACK] Seq=8 Ack=59 Win=14600 Len=0
28	0.103967	.69.50	.117.48	TCP	54	[TCP Dup ACK 27#1] 54014 → 23 [ACK] Seq=8 Ack=59 Win=14600 Len=0
29	0.103994	.69.50	.117.48	TELNET	62	Telnet Data ...
30	0.104044	.69.50	.117.48	TCP	62	[TCP Retransmission] 54014 → 23 [PSH, ACK] Seq=8 Ack=59 Win=14600 Len=8
43	0.304985	.117.48	.69.50	TCP	60	23 → 54014 [ACK] Seq=59 Ack=16 Win=4063 Len=0
44	0.305059	.117.48	.69.50	TCP	54	[TCP Dup ACK 43#1] 23 → 54014 [ACK] Seq=59 Ack=16 Win=4063 Len=0
511	2.105686	.117.48	.69.50	TELNET	83	Telnet Data ...
512	2.105837	.117.48	.69.50	TCP	83	[TCP Retransmission] 23 → 54014 [PSH, ACK] Seq=59 Ack=16 Win=4063 Len=29

> Frame 12: 95 bytes on wire (760 bits), 95 bytes captured (760 bits)  
> Ethernet II, Src: Routerbo\_70:58:7a (6c:3b:6b:70:58:7a), Dst: Routerbo\_83:d7:1e (6c:3b:6b:83:d7:1e)  
> Internet Protocol Version 4, Src: .117.48, Dst: .69.50  
> Transmission Control Protocol, Src Port: 23, Dst Port: 54014, Seq: 1, Ack: 1, Len: 41

Wireshark - Follow TCP Stream (tcp.stream eq 3) - hajime-np.pcap

```
% Authentication failed
Username: admin
admin
Password: 123456
% Authentication failed
```

8 client pkt(s), 2 server pkt(s), 4 sum(s).

Entire conversation (102 bytes) Show and save data as ASCII Stream 3

Find: Find Next

Filter Out This Stream Print Save as... Back Close Help

```
0000  6c 3b 6b 83 d7 1e 6c 3b 6b 70 58 7a 08 00 45 c0 |;.k...l; kpXz..E-
0010  00 51 6e e8 40 00 fb 06 66 bf 77 6e 75 30 77 6e |Qn@...f wnu0vm
0020  45 32 00 17 d2 fe 46 bd 5e e8 fb 83 54 30 50 18 |E2...F...TOP-
0030  0f ee e4 c3 00 0d 0a 0d 0a 25 20 41 75 74 68 |.....% Auth
0040  65 6e 74 69 63 61 74 69 6f 6e 20 66 61 69 6c 65 |enticati on fail
0050  64 0d 0a 0d 0a 55 73 65 72 6e 61 6d 65 3a 20 |d...Use rname:
```

hajime-np.pcap | Packets: 68981 · Displayed: 32 (0.0%) | Profile: Default

# Analyzing Packets – Flood Example (WINBOX)

hajime-np.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port eq 8291 || tcp.flags.syn

No.	Time	Source	Destination	Protocol	Length	Info
68959	304.500312	.69.50	.123.140	TCP	60	28809 → 8291 [SYN] Seq=0 Win=14600 Len=0
68960	304.500396	.69.50	.110.130	TCP	60	19497 → 8291 [SYN] Seq=0 Win=14600 Len=0
68961	304.500438	.69.50	.123.140	TCP	54	[TCP Out-Of-Order] 28809 → 8291 [SYN] Seq=0 Win=14600 Len=0
68962	304.500439	.69.50	.246.195	TCP	54	[TCP Out-Of-Order] 11758 → 23 [SYN] Seq=0 Win=14600 Len=0
68963	304.500503	.69.50	.110.130	TCP	54	[TCP Out-Of-Order] 19497 → 8291 [SYN] Seq=0 Win=14600 Len=0
68964	304.500538	.69.50	.106.187	TCP	60	35890 → 8291 [SYN] Seq=0 Win=14600 Len=0
68965	304.500619	.69.50	.60.94	TCP	60	61649 → 8291 [SYN] Seq=0 Win=14600 Len=0
68966	304.500684	.69.50	.106.187	TCP	54	[TCP Out-Of-Order] 35890 → 8291 [SYN] Seq=0 Win=14600 Len=0
68967	304.500721	.69.50	.151.111	TCP	60	44140 → 8291 [SYN] Seq=0 Win=14600 Len=0
68968	304.500766	.69.50	.60.94	TCP	54	[TCP Out-Of-Order] 61649 → 8291 [SYN] Seq=0 Win=14600 Len=0
68969	304.500850	.69.50	.151.111	TCP	54	[TCP Out-Of-Order] 44140 → 8291 [SYN] Seq=0 Win=14600 Len=0
68970	304.500856	.69.50	.98.233	TCP	60	61327 → 8291 [SYN] Seq=0 Win=14600 Len=0
68971	304.500885	.69.50	.119.191	TCP	60	2865 → 8291 [SYN] Seq=0 Win=14600 Len=0
68972	304.500967	.69.50	.98.233	TCP	54	[TCP Out-Of-Order] 61327 → 8291 [SYN] Seq=0 Win=14600 Len=0
68973	304.500983	.69.50	.119.191	TCP	54	[TCP Out-Of-Order] 2865 → 8291 [SYN] Seq=0 Win=14600 Len=0
68974	304.501001	.69.50	.186.77	TCP	60	15023 → 8291 [SYN] Seq=0 Win=14600 Len=0
68975	304.501098	.69.50	.186.77	TCP	54	[TCP Out-Of-Order] 15023 → 8291 [SYN] Seq=0 Win=14600 Len=0
68976	304.501129	.69.50	.120.221	TCP	60	24557 → 8291 [SYN] Seq=0 Win=14600 Len=0
68977	304.501157	.69.50	.57.155	TCP	60	1057 → 8291 [SYN] Seq=0 Win=14600 Len=0
68978	304.501246	.69.50	.57.155	TCP	54	[TCP Out-Of-Order] 1057 → 8291 [SYN] Seq=0 Win=14600 Len=0
68979	304.501248	.69.50	.120.221	TCP	54	[TCP Out-Of-Order] 24557 → 8291 [SYN] Seq=0 Win=14600 Len=0
68980	304.502189	.69.50	.222.170	TCP	60	7020 → 23 [SYN] Seq=0 Win=14600 Len=0
68981	304.502296	.69.50	.18.124	TCP	60	58167 → 23 [SYN] Seq=0 Win=14600 Len=0

> Frame 6290: 60 bytes on wire (480 bits), 60 bytes captured (480 bits)

> Ethernet II, Src: Routerbo\_83:d7:1e (6c:3b:6b:83:d7:1e), Dst: Routerbo\_70:58:7a (6c:3b:6b:70:58:7a)

> Internet Protocol Version 4, Src: .69.50, Dst: .181.140

> Transmission Control Protocol, Src Port: 65207, Dst Port: 8291, Seq: 0, Len: 0

# Analyzing Packets – Flood Example (SMB)

tcp445.pcap

File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help

tcp.port eq 445 || tcp.flags.syn

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.1.46	157.201.73.159	TCP	66	56217 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
2	0.000014	192.168.1.46	157.201.73.159	TCP	66	[TCP Out-Of-Order] 56217 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
3	0.015284	192.168.1.46	155.81.49.177	TCP	66	56216 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
4	0.015308	192.168.1.46	155.81.49.177	TCP	66	[TCP Out-Of-Order] 56216 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
5	0.015363	192.168.1.46	212.182.121.241	TCP	66	56218 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
6	0.015373	192.168.1.46	212.182.121.241	TCP	66	[TCP Out-Of-Order] 56218 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
7	0.018396	192.168.1.46	219.78.8.88	TCP	66	56567 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
8	0.018414	192.168.1.46	219.78.8.88	TCP	66	[TCP Out-Of-Order] 56567 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
9	0.046451	192.168.1.46	124.37.75.228	TCP	66	56225 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
10	0.046465	192.168.1.46	124.37.75.228	TCP	66	[TCP Out-Of-Order] 56225 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
11	0.046523	192.168.1.46	217.46.218.120	TCP	62	55541 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
12	0.046533	192.168.1.46	217.46.218.120	TCP	62	[TCP Out-Of-Order] 55541 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 SACK_PERM=1
13	0.046555	192.168.1.46	91.38.86.182	TCP	66	56226 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
14	0.046562	192.168.1.46	91.38.86.182	TCP	66	[TCP Out-Of-Order] 56226 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
15	0.093386	192.168.1.46	183.175.120.189	TCP	66	56229 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
16	0.093404	192.168.1.46	183.175.120.189	TCP	66	[TCP Out-Of-Order] 56229 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
17	0.108902	192.168.1.46	189.220.119.135	TCP	66	56235 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
18	0.108915	192.168.1.46	189.220.119.135	TCP	66	[TCP Out-Of-Order] 56235 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
19	0.108953	192.168.1.46	190.64.238.86	TCP	66	56231 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
20	0.108961	192.168.1.46	190.64.238.86	TCP	66	[TCP Out-Of-Order] 56231 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
21	0.108977	192.168.1.46	197.87.120.30	TCP	66	56233 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
22	0.108984	192.168.1.46	197.87.120.30	TCP	66	[TCP Out-Of-Order] 56233 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
23	0.126188	192.168.1.46	115.16.221.99	TCP	66	56575 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1
24	0.126210	192.168.1.46	115.16.221.99	TCP	66	[TCP Out-Of-Order] 56575 → 445 [SYN] Seq=0 Win=8192 Len=0 MSS=1460 WS=4 SACK_PERM=1

> Frame 1: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)  
> Ethernet II, Src: HostEngi\_24:f6:2e (00:e0:62:24:f6:2e), Dst: Routerbo\_d4:5a:67 (d4:ca:6d:d4:5a:67)  
> Internet Protocol Version 4, Src: 192.168.1.46, Dst: 157.201.73.159  
> Transmission Control Protocol, Src Port: 56217, Dst Port: 445, Seq: 0, Len: 0

# Analyzing Packets – Wireshark Reference

- Wireshark Website

<http://www.wireshark.org>

- Wireshark Documentation

<http://www.wireshark.org/docs/>

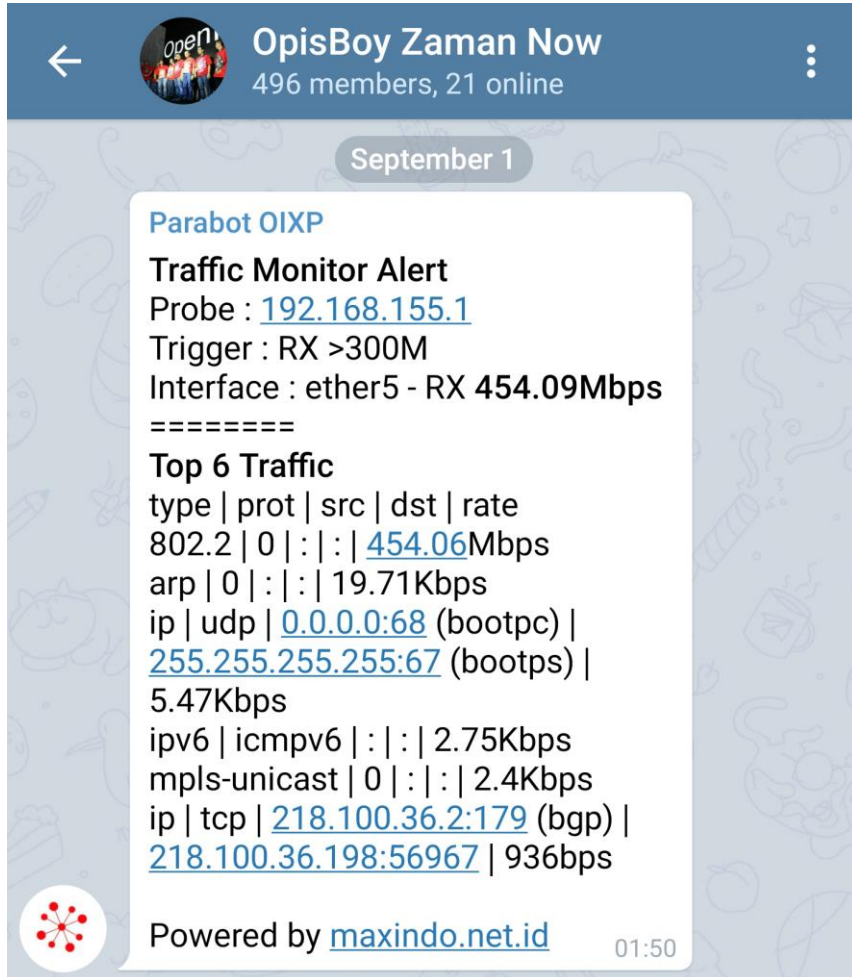
- Wireshark Wiki

<http://wiki.wireshark.org>

- Network analysis Using Wireshark Cookbook

<http://www.amazon.com/Network-Analysis-Using-WiresharkCookbook/dp/1849517649>

# Study Case – Parabot OpenIXP



The screenshot shows a Telegram chat interface. At the top, the chat is titled 'OpisBoy Zaman Now' with 496 members and 21 online. The message is dated 'September 1'. The message content is as follows:

**Parabot OIXP**  
**Traffic Monitor Alert**  
Probe : [192.168.155.1](#)  
Trigger : RX >300M  
Interface : ether5 - RX 454.09Mbps  
=====  
**Top 6 Traffic**  
type | prot | src | dst | rate  
802.2 | 0 | : | : | 454.06Mbps  
arp | 0 | : | : | 19.71Kbps  
ip | udp | [0.0.0.0:68](#) (bootpc) | [255.255.255.255:67](#) (bootps) | 5.47Kbps  
ipv6 | icmpv6 | : | : | 2.75Kbps  
mpls-unicast | 0 | : | : | 2.4Kbps  
ip | tcp | [218.100.36.2:179](#) (bgp) | [218.100.36.198:56967](#) | 936bps

Powered by [maxindo.net.id](#) 01:50

- **OpenIXP** is one of the biggest Internet Exchange in Indonesia
- and **Parabot**, a Bot in Telegram that brewed by **@ericksetiawan** and the Infrastructure was provided by **@mtakeuchi** using MikroTik RouterOS as a Probe & BGP router in OpenIXP, also Powered by **Maxindo Networks**
- Parabot help to notify us when the router **receiving** broadcast or flood on OpenIXP interface
- Parabot will do Torch and start **Packet Sniffer** on your Router



## Conclusion

Secure  $\neq$  Easy

Feel so hard to analyze?  
Let me help you!

[michael@takeuchi.id](mailto:michael@takeuchi.id)

<https://www.facebook.com/mict404>

<https://www.linkedin.com/in/michael-takeuchi/>

# Question & Answer



Slide is available in my GitHub repository

<https://github.com/mict404/slide/>

*Thank  
you*

