Understanding Wireshark’s Reassembly Features

Christian Landström, Airbus DS
Agenda

- Introduction to Reassembly Features
- Use cases where Reassembly is used
- Side effects of the feature stack
- Best practices and recommendations
Introduction to Reassembly Features

- Reassembly works within:
  - IP
  - TCP
  - SSL
- Can be toggled via different ways
- Default: All features turned ON
Hands-on time!

- Fire up your Wireshark and capture your traffic (highly recommended)
- Go to: www.packet-foo.com/SF15
- Alternatively click along using the sample captures: www.packet-foo.com/SF15/Talk20.zip
Focus: TCP Stream Reassembly

- Regularly used within network analysis
- Enables reconstruction of segmented payload
Let’s do some network analysis

- Use case: Application Server Analysis
  - To be analyzed: Application response times
  - Simple with HTTP: delta time Request <-> Response
Going from request to response

- Simple with delta displayed
  - Remember to filter for single TCP sessions before
  - Refer to Round-Trip-Time (RTT) for real application response time, depending where the capture was taken
How about our important data?

- Check webserver application response time

That's a fast one!!
Questions up to here?

- Everybody agrees on the timings? (roughly if captured by yourself)
- Anyone having strange behavior with his/her Wireshark version?
That’s where reassembly kicks in

- Watch the difference:
Side-Effects within TCP Reassembly

- Possible Re-Ordering of INFO-Column statements within the packet list
- Affects display filters too (e.g. http.response)
- Changes to the labelling of the „protocol“ column within Wireshark
  ➔ Also possibly affects display filters, statistics etc.
Side-Note: Wireshark Bugs #1?

- Filter for all HTTP request and HTTP responses
  → GUI export or tshark
- Save into new capture file and open for analysis --> Watch the number of packets!!
Side-Note: Wireshark Bugs #2?

Valid until some version between 1.10.x and 1.12.x

- Check the protocol hierarchy statistics
- Watch for HTTP percentage
- Try to explain the different results based on reassembly setting
No bugs of course!

- Yet more side-effects of reassembly
- Valid output, but strongly dependent on the question you ask:
  - Time until start OR end of data stream delivery
  - Statistics of ALL HTTP-related packets, meaning tcp.port==80
    OR
    All HTTP-related packets containing data (without ACKs, Handshake etc.)
  OR
  Just the Requests and Response packets
Best practices

- Watch carefully!
- Use separate Profiles
  - Turn off reassembly for any timing / statistics based analysis tasks
  - Turn on reassembly for content analysis / forensics
- Check your default profile, since it is the base setting for tshark on command line level
!! Thank you for attending !!

## Questions? ##

---------------------------
eMail: landi@packet-foo.com
Web: www.packet-foo.com
Twitter: @0x6C616E6469