

# Lightweight Real-time Display Tool

Using Open Source

Paul Ferrill – [paul.ferrill@avtest.com](mailto:paul.ferrill@avtest.com)

CTO – Avionics Test and Analysis Corporation

# Problem Statement

- A preflight display tool is required for confirmation of a small set of key data items before takeoff.
- Test aircraft will be equipped with a Chapter 10 recorder capable of streaming data over Ethernet.
- The software should be able to display simple text and moving strip-chart type graphics.
- Data should be recorded locally for documentation purposes.

# Why Python?

- Libraries
  - Chapter 10
  - Scapy
  - Sqlite3
  - Standard library
    - Web server
    - Data mangling (bit twiddling)
  - WSGI (Web Server Gateway Interface)
    - PEP 333 - <https://www.python.org/dev/peps/pep-0333/>
- Programmer experience

# Application with Matplotlib

- Leverage work done on weather radar display
- Add real-time update capability
- Combine packet reading / parsing plus graphing into single tool
- Distribute in source form for easy installation
- Compile using PY2EXE for executable
- Create video using ffmpeg if needed

Time: 18:03:49.702837

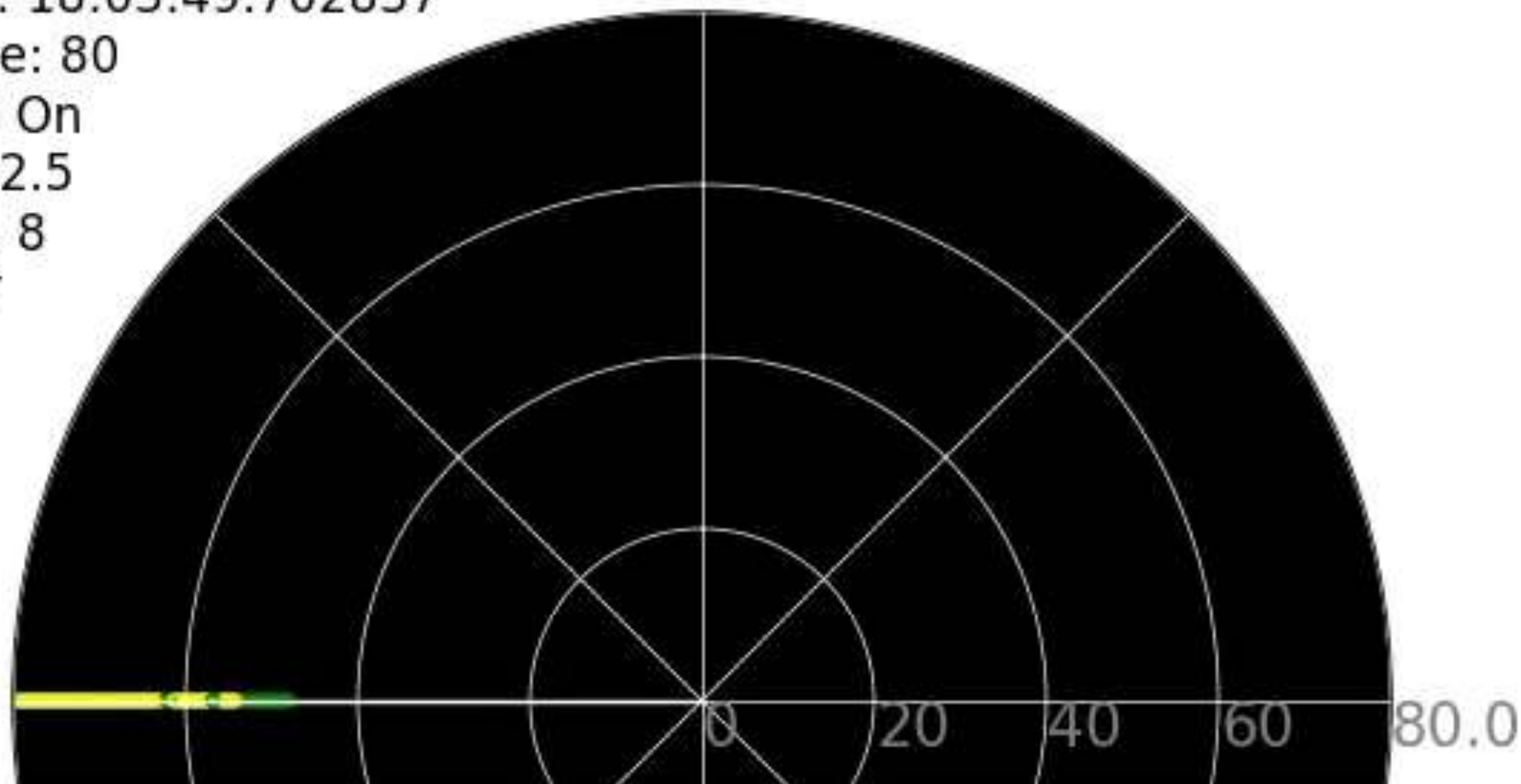
Range: 80

STAB On

Tilt: -2.5

Gain: 8

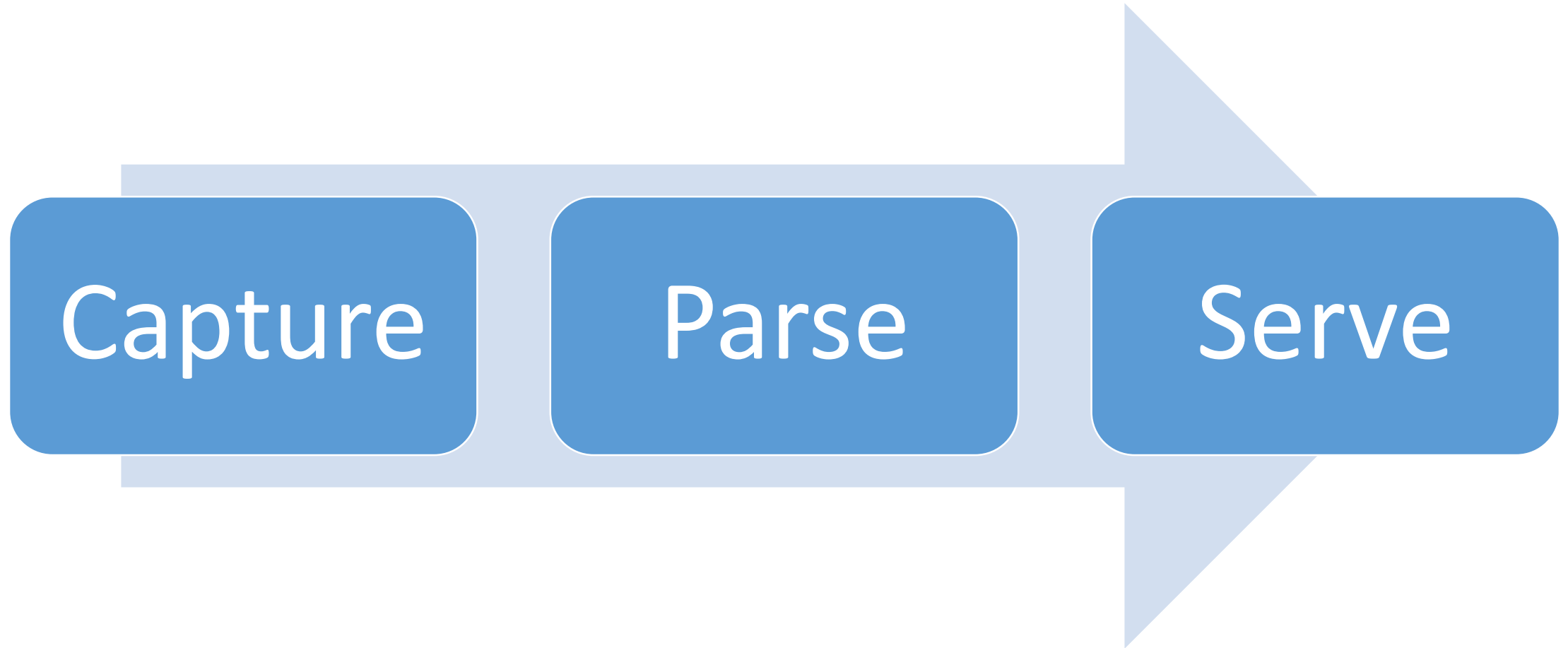
WX/T



# Issues with single application

- Does not scale well
- Requirement to use privileged code for packet capture
- Potential performance issues not easily overcome with single monolithic code

# Packet Processing



# General Architecture

- Sniffer loads Chapter 10 data from Ethernet or PCAP file into a database.
- Frontend web application displays data from database in a browser-based frontend using javascript.
- Open source tools used for graphics, display and database.
- Leverage existing toolsets for maximum software reuse.
- Single proprietary element in the Python library for parsing Chapter 10 data.



# Sniffing Chapter 10 over Ethernet

- Scapy – Python based network testing tool
- Takes a PCAP file or a wireshark-like filter to get a set of network packets
- Searches network data for chapter 10 packets (with some light validation) and stores in a SQLite database
- listen.py = 63 lines of executable code (104 with blank lines and comments)

# Why jQuery

- Mature JavaScript library for implementing Ajax application.
- Large number of jQuery plugins for implementing new functionality as in Flot.
- Plenty of examples and tutorials on getting started with jQuery.
- Cross-browser support and cross-platform

# Frontend Webapp

- Python-based webserver (using Python standard library tools) provides data upon request from the front-end application.
- jQuery based front end application configures itself and requests data to plot, playback, etc.
- Flot extension for jQuery support a variety of charts with the ability to update in real time.
- server.py = 74 lines of executable code (104 with blank lines and comments)

# jQuery plotting

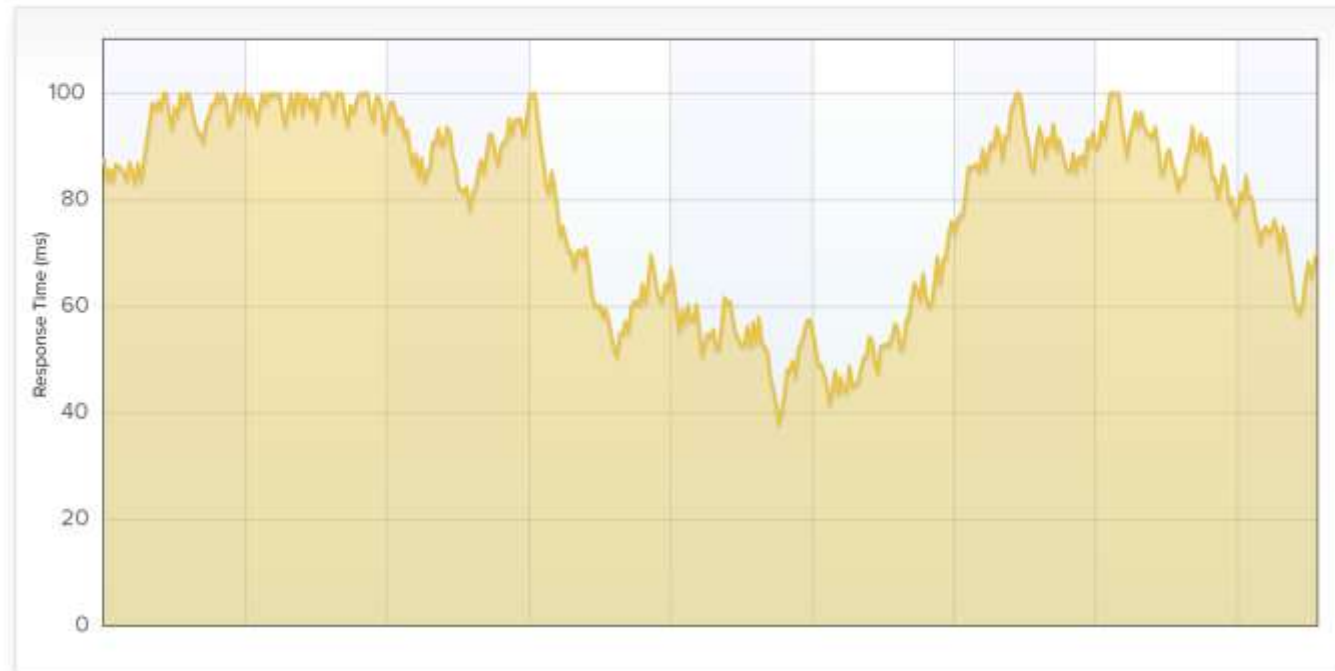
**Flot** [Documentation](#) [Examples](#) [Plugins](#) [Blog](#) [Forum](#) [Bugs](#) [f](#) [t](#) [w](#)

**Attractive JavaScript plotting for jQuery**

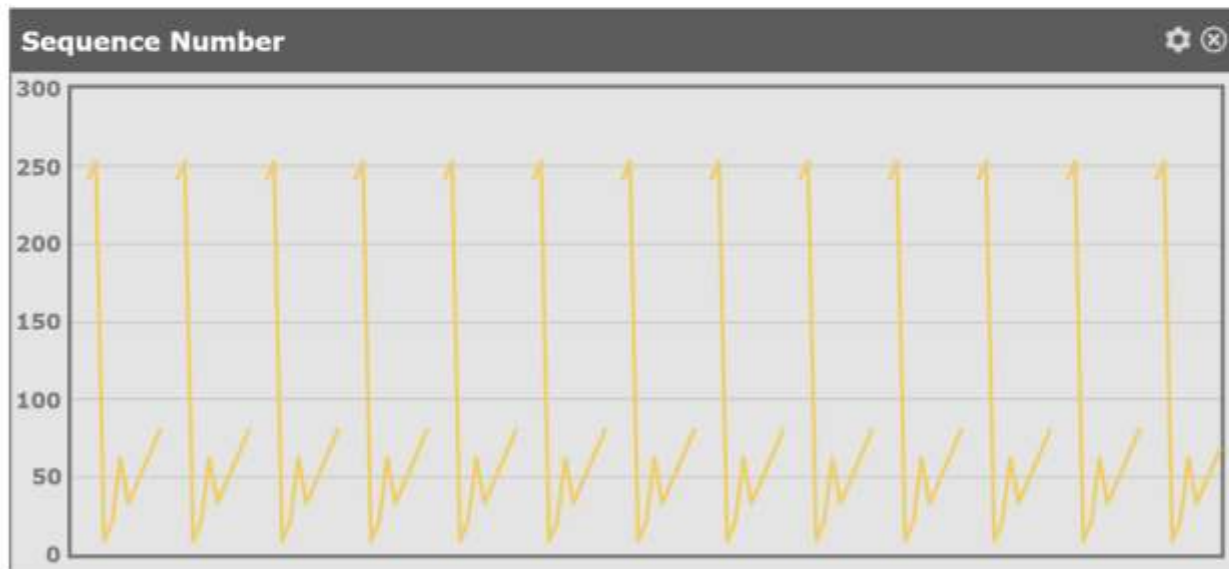
Flot is a pure JavaScript plotting library for jQuery, with a focus on simple usage, attractive looks and interactive features.

Works with Internet Explorer 6+, Chrome, Firefox 2+, Safari 3+ and Opera 9.5+

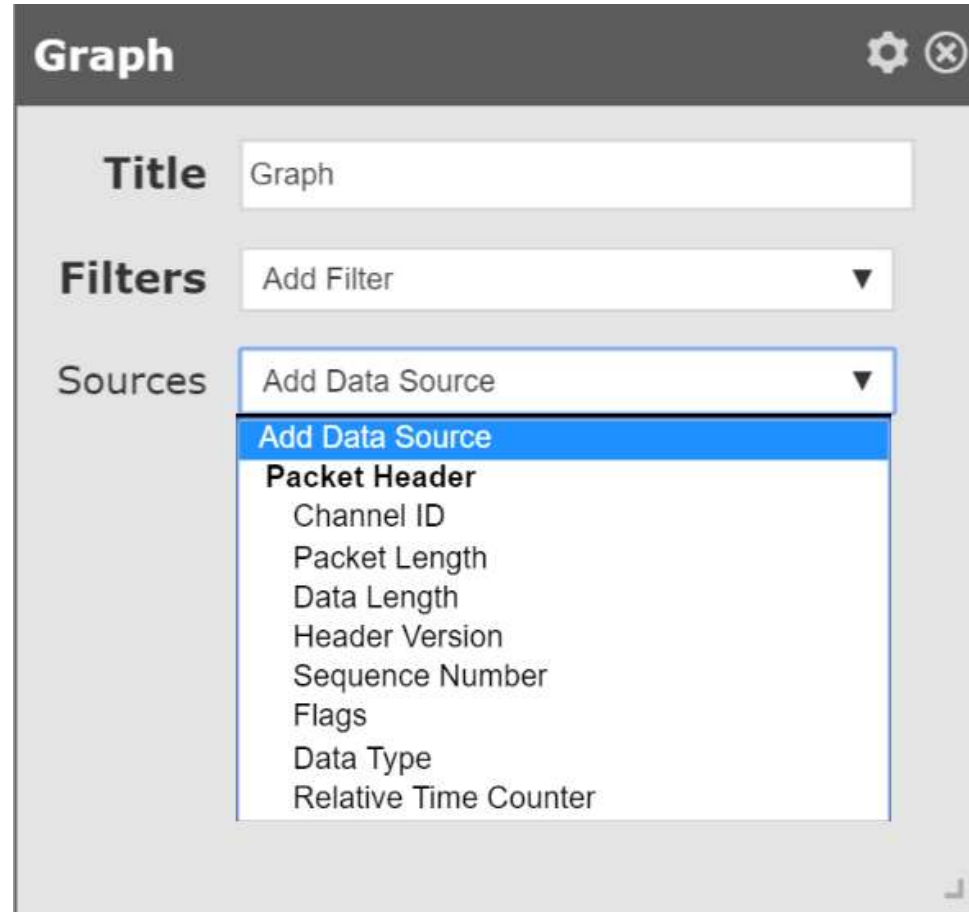
 **Version 0.8.3**  **Development**



Autogenerated Widget <span>⚙️</span> <span>✕</span>	
Channel ID	6
Data Type	9
Packet Length	65564
Sequence Number	69
Flags	3
Relative Time Counter	158791472529506
<input type="text" value="Relative Time Counter"/> ▾	



# Add a Widget



# Futures

- Parameter editor / TMATS definitions
- Layout editor
- Python-based derived processing
- Compiled packet processor
- Containers
- Distributed processing
- Move WSGI into recorder
- Low-cost packet processor hardware (Think Raspberry Pi)

# Resources / Links

- Flot
  - <http://www.flotcharts.org/>
  - <https://github.com/flot/flot/blob/master/README.md>
- jQuery
  - <https://jquery.com/>
- Scapy
  - <http://www.secdev.org/projects/scapy/>



Questions?