An API to filter network flows in the web to use as plugin in web based network visualization apps

Julio de la Cruz, Ian Dávila, Dr. José Ortiz Ubarri
Computer Science
University of Puerto Rico

Outline

- Background
- API Web Interface
- API Backbone
- Visualizations

Previous work

- FlowScan
- NetFlow Sensor (tied to nfdump)
- NVisionIP (FloCon 2005)
- FloVis (FloCon 2009)
- Stager (FloCon 2010)
- FlowViewer (FloCon 2013)
- Rayon & Prism (FloCon 2014)

Previous work: Rayon & Prism

- Prism is a tool for quickly visualizing flow data as a time-series broken down into several configurable bins by SiLK's rwfilter tool.
- It uses Rayon to visualize the network flows.
- Rayon is a Python library and set of tools for generating basic two-dimensional statistical visualizations.
    - Scatterplots
    - Bar plots
    - Time series visualizations

Previous work: Flowbat

- “...analyst-focused graphical interface for analyzing flow data.”
- Features:
    - Multiple Deployment Scenarios
    - Quick Query Interface
    - Graphing and Statistical Capability
        - Generate bar, line, column, and pie charts

Goal

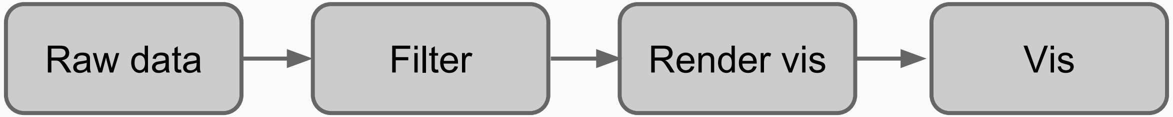
- To create an API that allow system administrators to manage network flows data in the web, to provide a bridge between the processing of big network data and visualization analytics researchers and provide network analysis as a web service in the cloud.

- The API will be part of Toa which was presented in
    - J. Ortiz-Ubarri, H. Ortiz-Zuazaga, A. Maldonado, E. Santos, J. Grullón. Toa: A Web-Based NetFlow Data Network Monitoring System. In Proceedings FloCon 2015, Portland Oregon. January 2015.

Features

- It can be implemented with Python CGI’s or Flask and with Javascript
- Web interface implementation based in bootstrap
  - Query interface
- Allows to pivot the data by applying filters to the already filtered results
- Graphing capability
- Visualization developers can create their own graphs to use with the API functions

Generic data preparation process



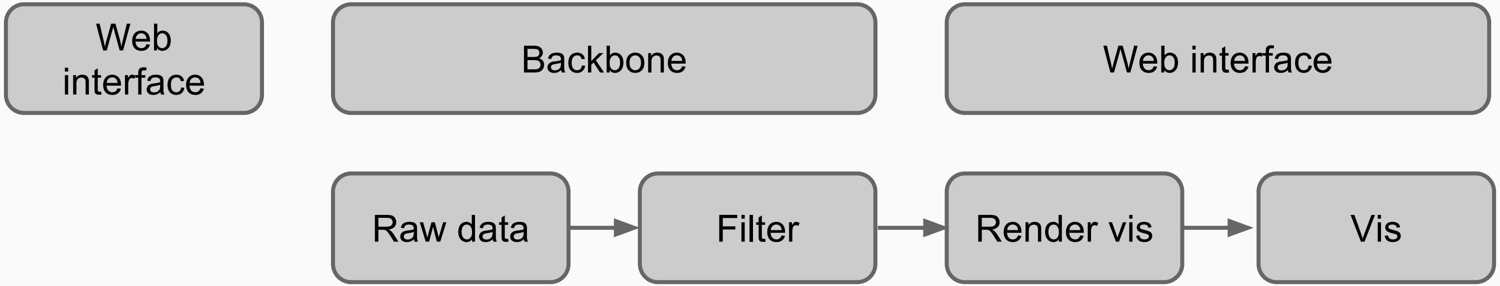
Raw data

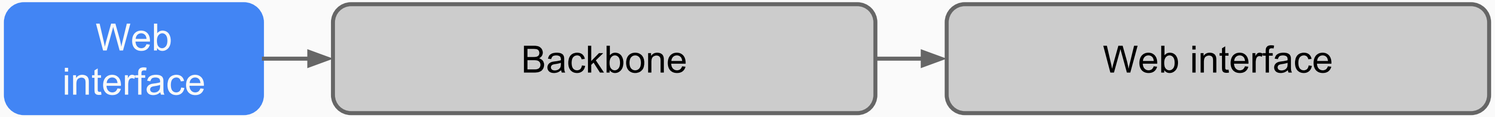
Filter

Render vis

Vis

Our API data preparation process

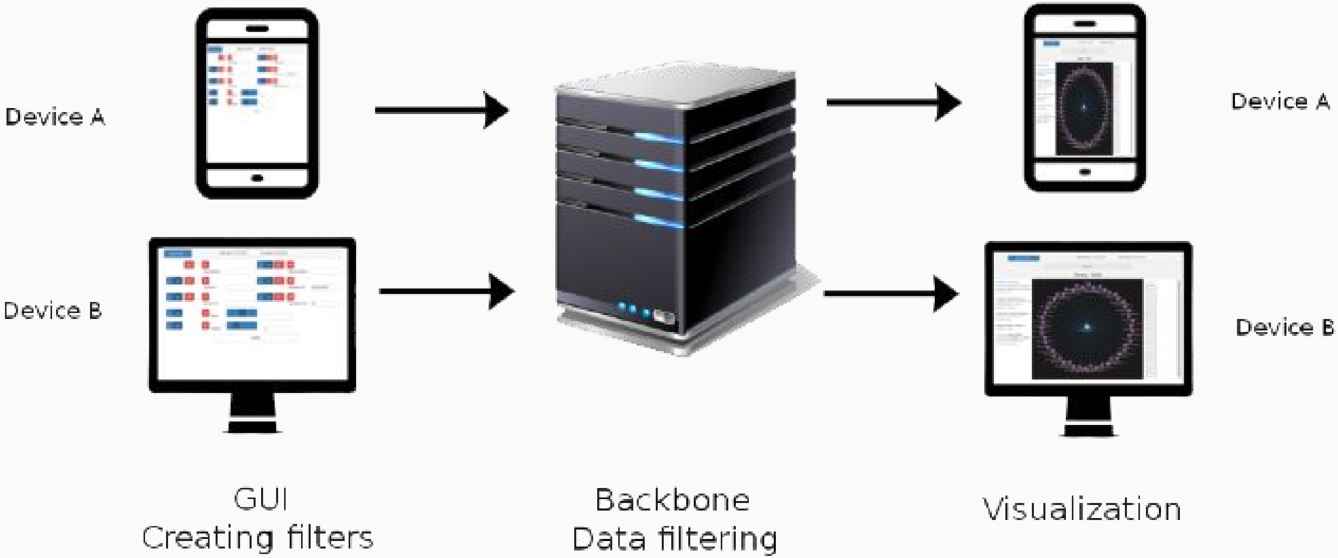




Web interface

Backbone

Web interface



Device A

Device B

GUI
Creating filters

Backbone
Data filtering

Device A

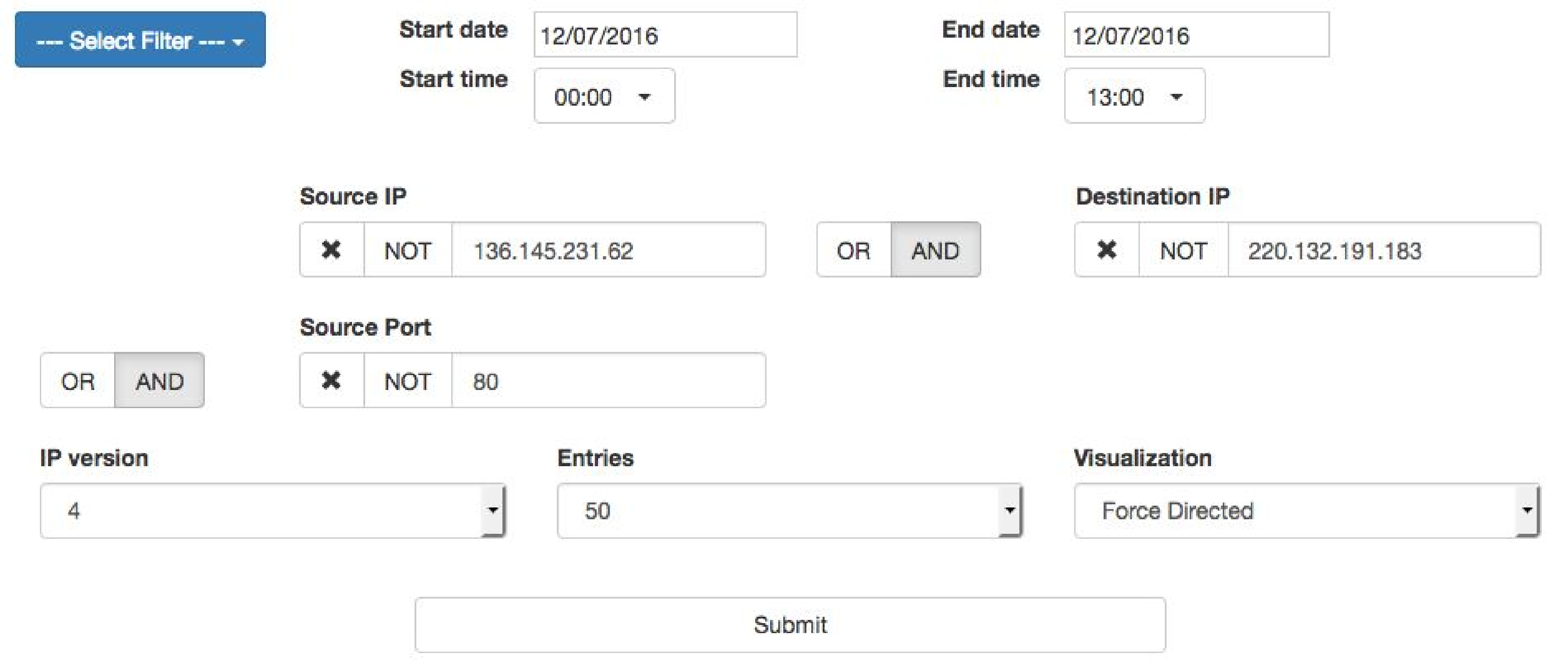
Device B

Visualization

API: Web interface

- AngularJS
  - “...is a structural framework for dynamic web apps. It lets you use HTML as your template language and lets you extend HTML's syntax to express your application's components clearly and succinctly.”
- Bootstrap
  - “...is the most popular HTML, CSS, and JavaScript framework for developing responsive, mobile-first web sites.”

API: Web interface



Start date 12/07/2016

Start time 00:00

End date 12/07/2016

End time 13:00

Source IP ✖ NOT 136.145.231.62

OR AND

Destination IP ✖ NOT 220.132.191.183

OR AND

Source Port ✖ NOT 80

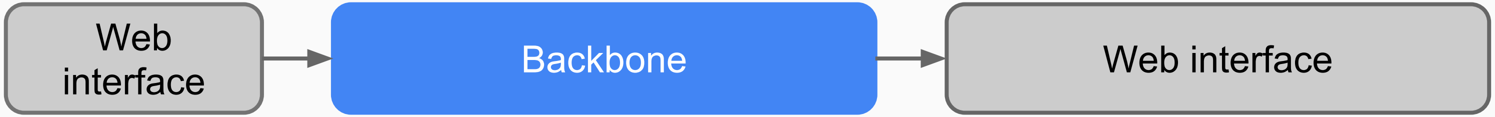
IP version 4

Entries 50

Visualization Force Directed

Filters

- Input and Output interface
- Source and destination IP
- Source and destination Port
- Bytes
- Packets

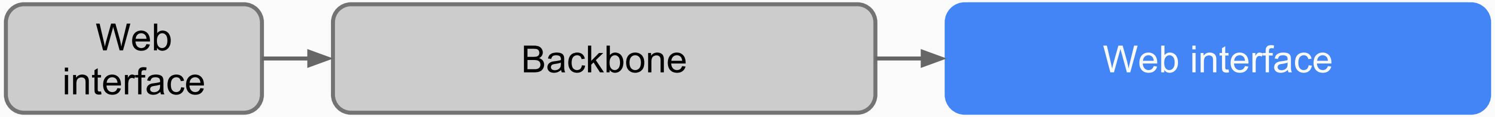

Web interface
Backbone
Web interface

API: Backbone

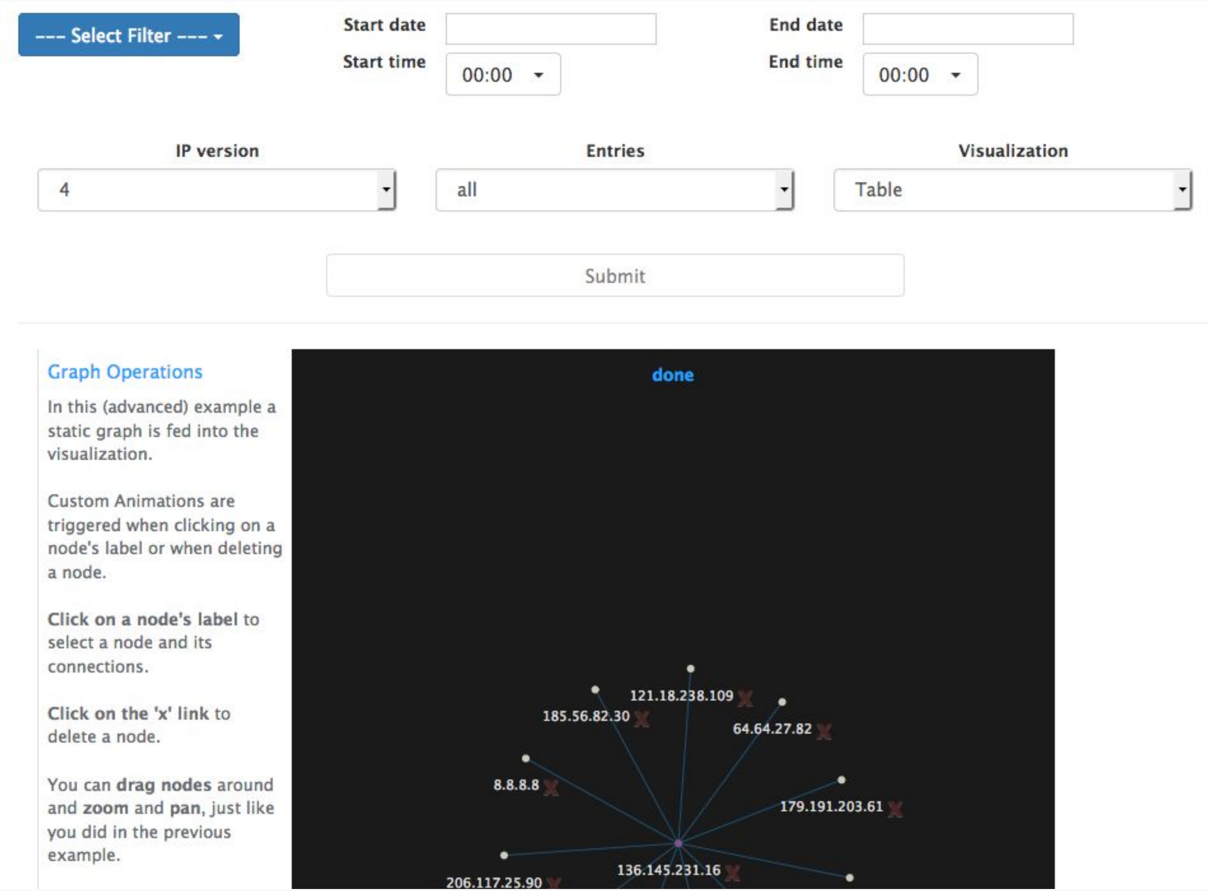
- Python CGI
- Flask
- Javascript

API: Backbone

- Receive and translates the query created by the user to the actual filters that are applied to the network data.
- Utilizes PySilk extension to retrieve the network flows stored in the file system generated in the given time frame.
- At the same time it applies the filters to the flows.
- Results are returned to the Web Interface


Web interface
Backbone
Web interface

API: Web interface



Start date

End date

Start time 00:00

End time 00:00

IP version 4

Entries all

Visualization Table

Graph Operations

In this (advanced) example a static graph is fed into the visualization.

Custom Animations are triggered when clicking on a node's label or when deleting a node.

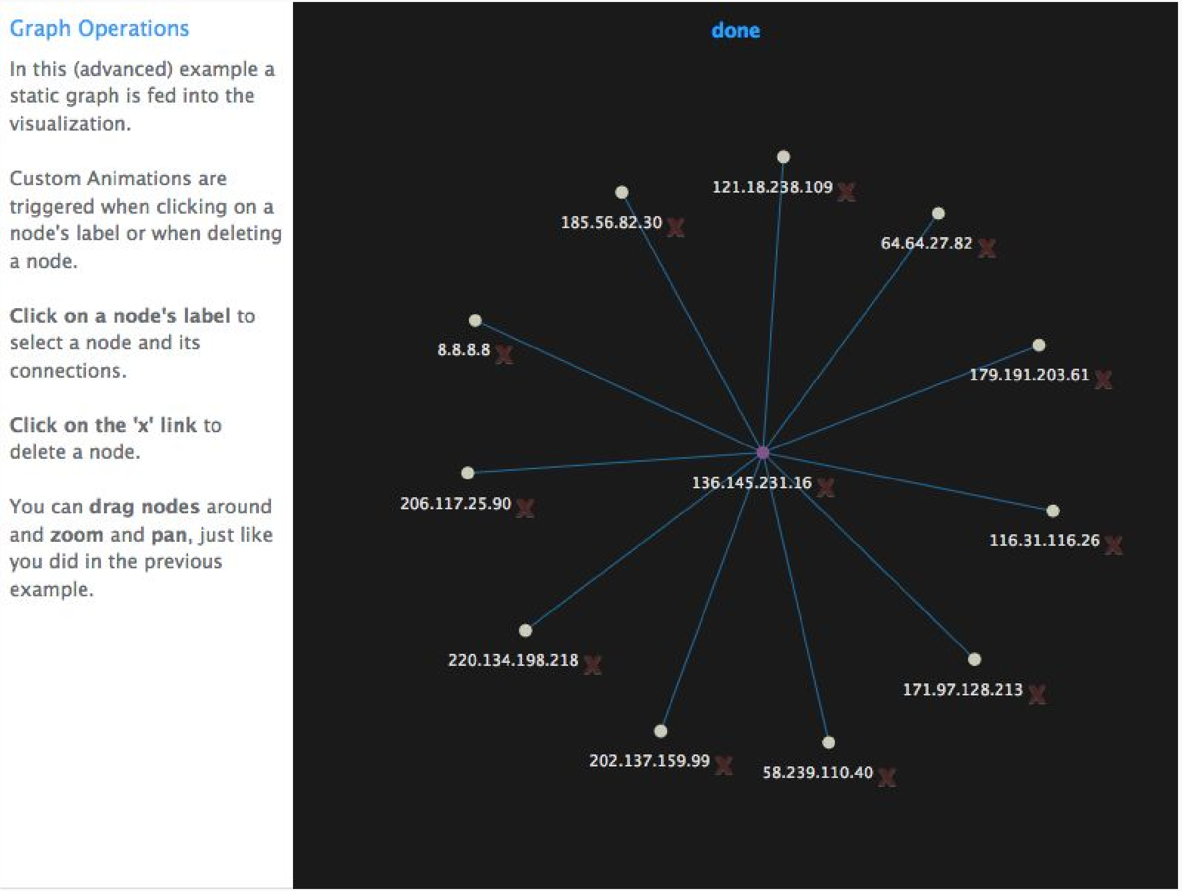
Click on a node's label to select a node and its connections.

Click on the 'x' link to delete a node.

You can **drag nodes** around and **zoom** and **pan**, just like you did in the previous example.

The user is able to either query the already filtered data or query the entire database again.

Visualizations: Force Directed



Graph Operations

In this (advanced) example a static graph is fed into the visualization.

Custom Animations are triggered when clicking on a node's label or when deleting a node.

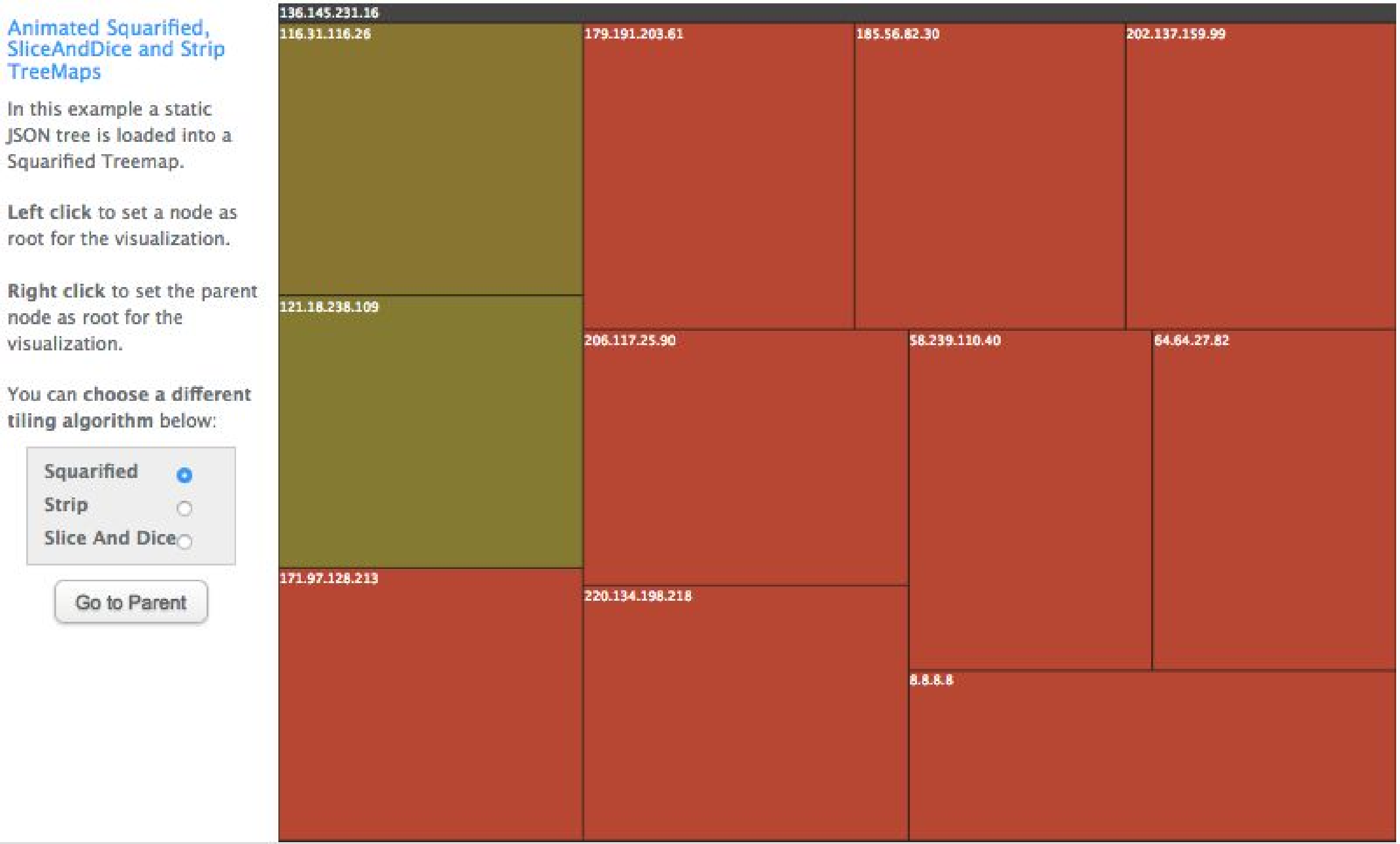
Click on a node's label to select a node and its connections.

Click on the 'x' link to delete a node.

You can **drag nodes** around and **zoom** and **pan**, just like you did in the previous example.

The Force Directed Graph is used for finding botnets, and DDoS.

Visualizations: Tree Map



Animated Squarified, SliceAndDice and Strip TreeMaps

In this example a static JSON tree is loaded into a Squarified Treemap.

Left click to set a node as root for the visualization.

Right click to set the parent node as root for the visualization.

You can **choose a different tiling algorithm** below:

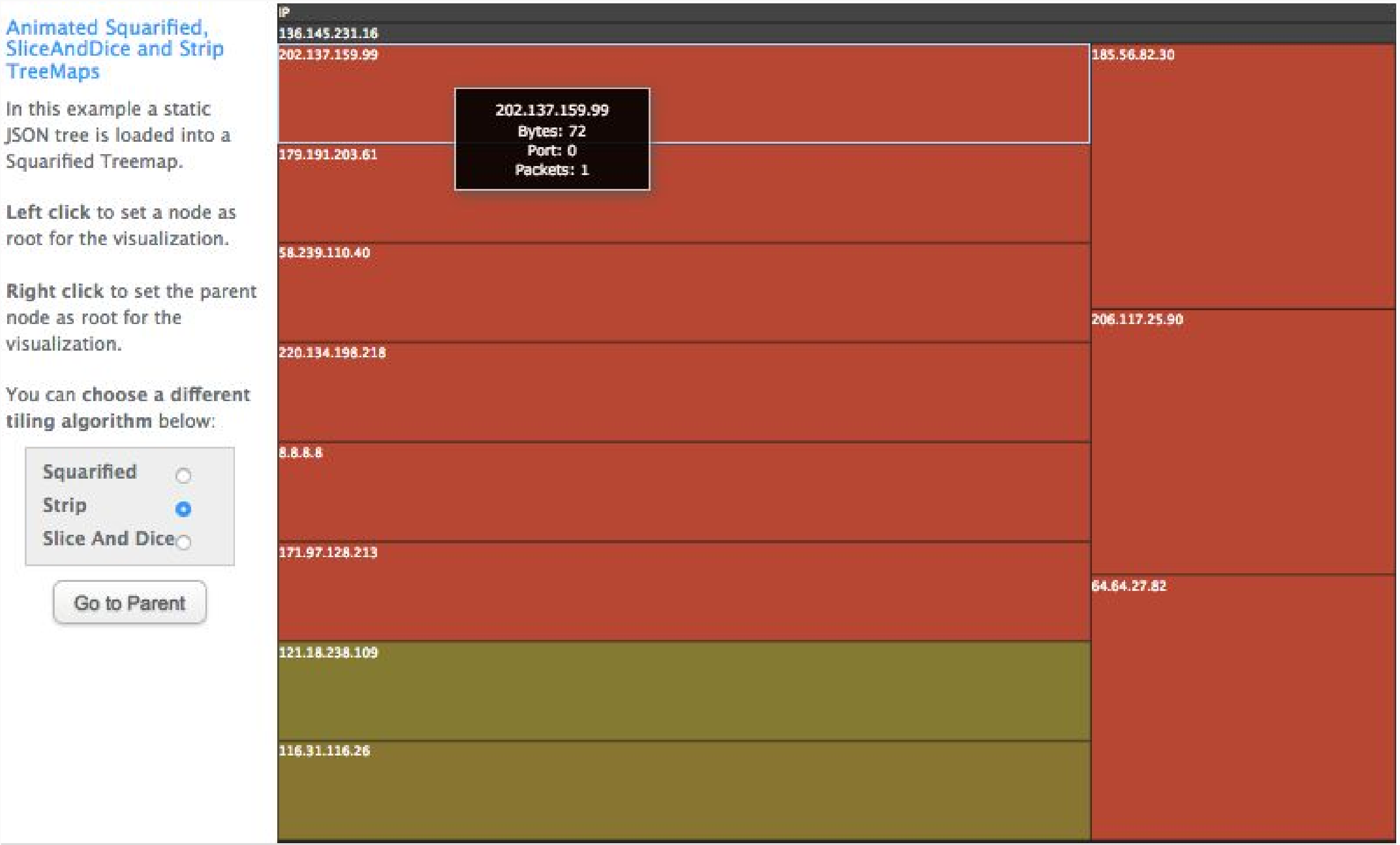
- Squarified ●
- Strip ○
- Slice And Dice ○

The Tree Map is used to find the top computers generating traffic

Options:

- Squarified

Visualizations: Tree Map



Animated Squarified, SliceAndDice and Strip TreeMaps

In this example a static JSON tree is loaded into a Squarified Treemap.

Left click to set a node as root for the visualization.

Right click to set the parent node as root for the visualization.

You can **choose a different tiling algorithm** below:

Squarified ○
Strip ●
Slice And Dice ○

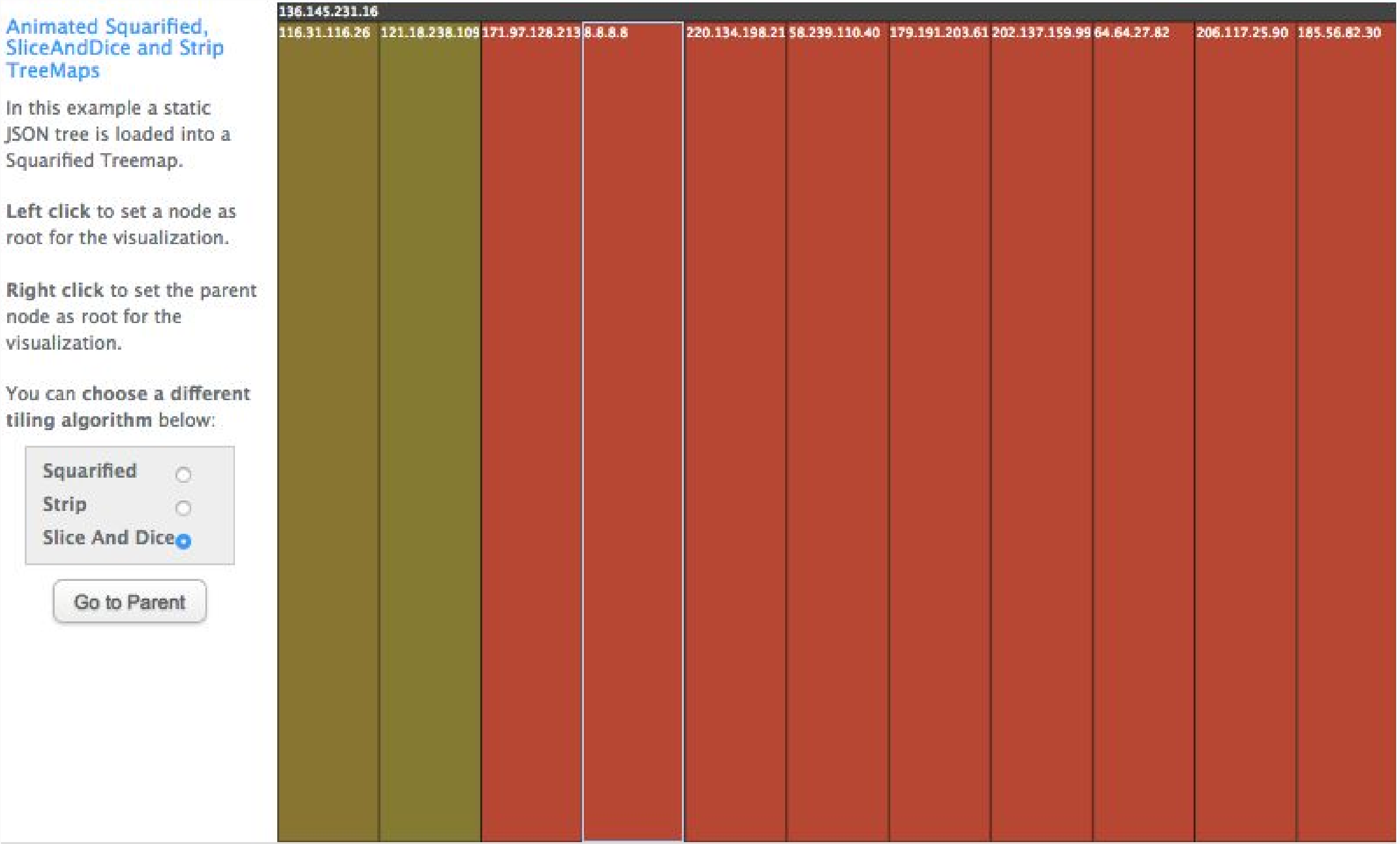
202.137.159.99
Bytes: 72
Port: 0
Packets: 1

The Tree Map is used to find the top computers generating traffic

Options:

- Squarified
- Strip

Visualizations: Tree Map



Animated Squarified, SliceAndDice and Strip TreeMaps

In this example a static JSON tree is loaded into a Squarified Treemap.

Left click to set a node as root for the visualization.

Right click to set the parent node as root for the visualization.

You can **choose a different tiling algorithm** below:

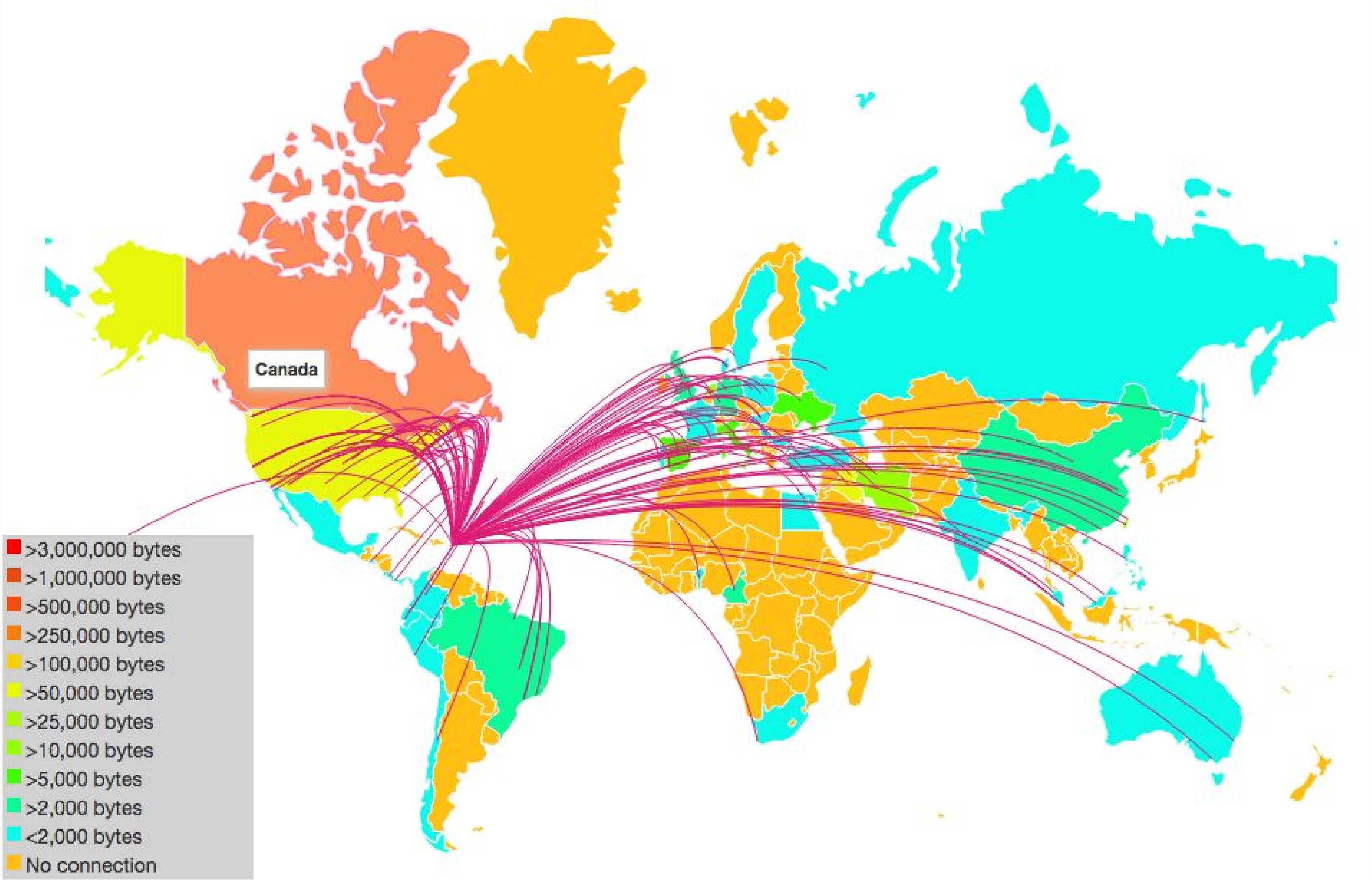
Squarified ○
Strip ○
Slice And Dice ●

The Tree Map is used to find the top computers generating traffic

Options:

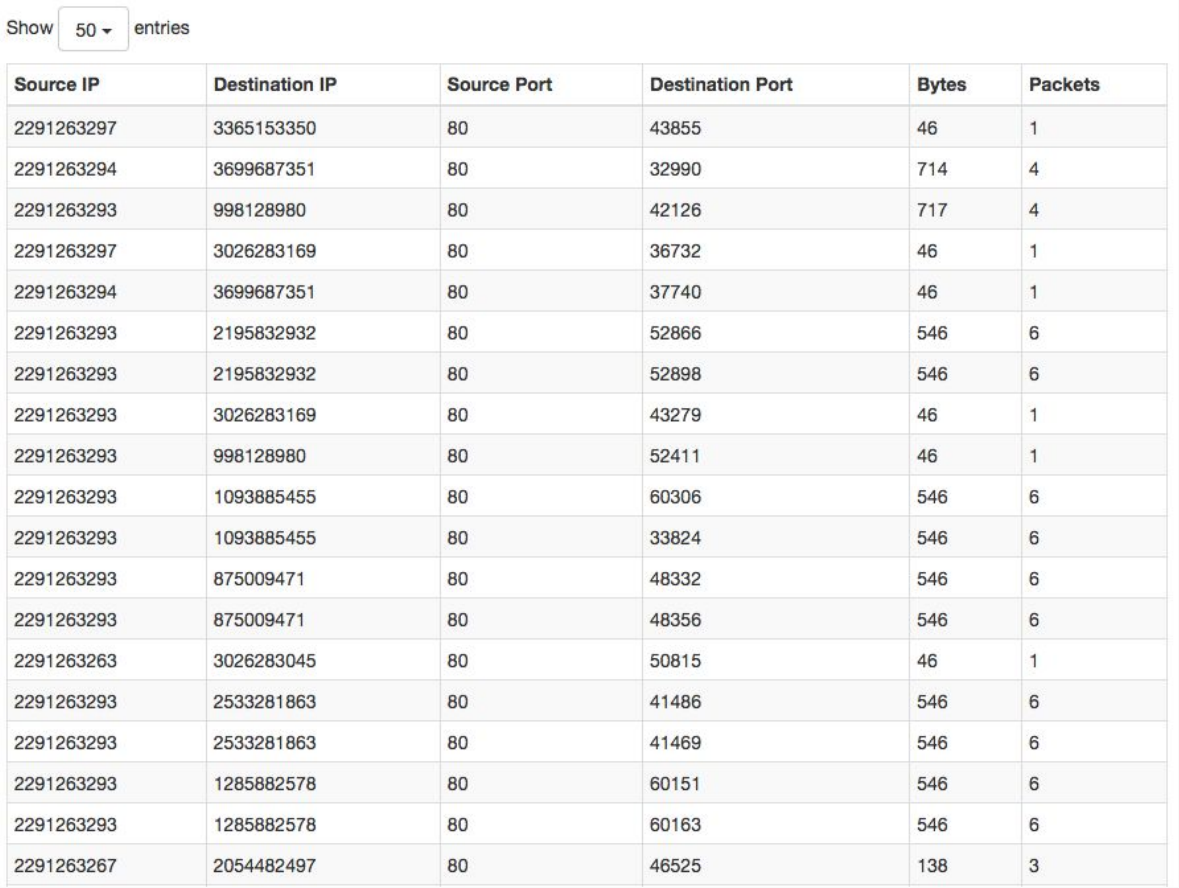
- Squarified
- Strip
- Slice and Dice

Visualizations: Data Map



The Data Map is used to find the top countries generating traffic and to detect unexpected connections

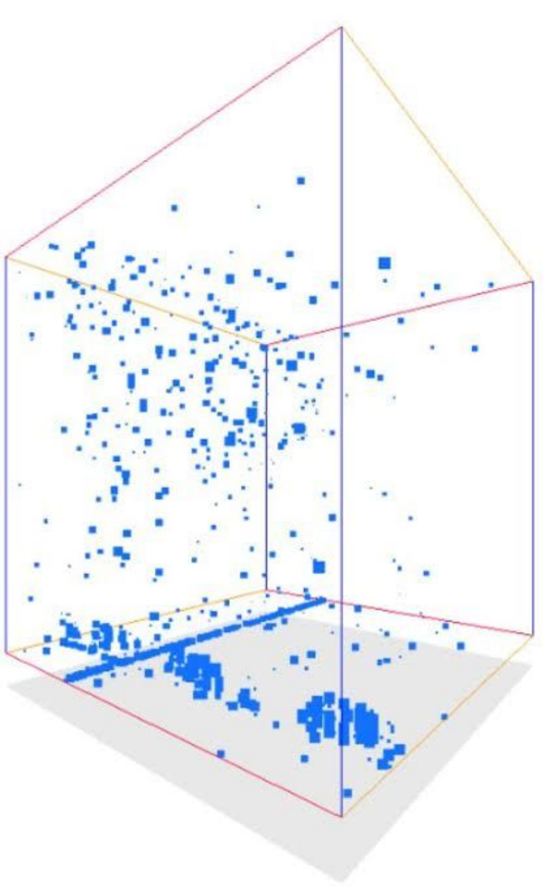
Visualizations: Table



Show 50 ▾ entries

Source IP	Destination IP	Source Port	Destination Port	Bytes	Packets
2291263297	3365153350	80	43855	46	1
2291263294	3699687351	80	32990	714	4
2291263293	998128980	80	42126	717	4
2291263297	3026283169	80	36732	46	1
2291263294	3699687351	80	37740	46	1
2291263293	2195832932	80	52866	546	6
2291263293	2195832932	80	52898	546	6
2291263293	3026283169	80	43279	46	1
2291263293	998128980	80	52411	46	1
2291263293	1093885455	80	60306	546	6
2291263293	1093885455	80	33824	546	6
2291263293	875009471	80	48332	546	6
2291263293	875009471	80	48356	546	6
2291263263	3026283045	80	50815	46	1
2291263293	2533281863	80	41486	546	6
2291263293	2533281863	80	41469	546	6
2291263293	1285882578	80	60151	546	6
2291263293	1285882578	80	60163	546	6
2291263267	2054482497	80	46525	138	3

Visualizations: Cube



The Cube uses WebGL and Three.js. Threats such as network and port scan can be detected.

References

- Dart, E., Rotman, L., Tierney, B., Hester, M., Zurawski, J. (2014). The science dmz: A network design pattern for data-intensive science. Scientific Programming, 22(2), 173-185.
- Reviews, C. (2012).e-study guide for cryptography and network security: Computer science, computer security. Cram101. Retrieved from https://books.google.com.pr/books?id=jr4ppkcEglUC
- Ortiz-Ubarri, J., Ortiz-Zuazaga, H., Maldonado, A., Santos, E., Grullon, J. (2015, June). Toa: A Web Based Network Flow Data Monitoring System at Scale. In Big Data (BigData Congress), 2015 IEEE International Congress on (pp. 438-443). IEEE.
- SiLK. (n.d.). Retrieved January 28, 2016, from https://tools.netsa.cert.org/silk/
- Belmonte, N. "Javascript InfoVis Toolkit. http." thejit. org/about.
- About FlowBAT. Retrieved December 08, 2016, from http://www.flowbat.com/about-flowbat.html
- AngularJS. Retrieved December 08, 2016, from https://docs.angularjs.org/guide/introduction
- Bootstrap 3 Tutorial. Retrieved December 08, 2016, from http://www.w3schools.com/bootstrap/

Questions?

Thank you!