Web Implementation of the Spinning Cube of Potential Doom for Netflow Data Analysis

Jhensen Grullón Sanabria, Eric D. Santos Sosa, José R. Ortiz Ubarri

Department of Computer Science, University of Puerto Rico, Rio Piedras Campus

Abstract

The monitoring and analysis of computer networks to detect abnormal network behavior is a complex task for network administrators, which is assisted with the use of visualization analytics. In this work, we present a web based implementation of the three dimensional cube known as the Spinning Cube of Potential Doom using WebGL and the Three.js library. With this implementation our goal is to provide an easy to use and access cube interface, without the need to install and configure additional graphic software in a single machine. This application allows the system administrators to visualize distinct network events such as network and port scanning. In our current implementation we use data from NetFlows stored in a remote server, and provide the system administrator with a control panel that allows the selection of past data, and filtering by networks classes. The computing needed for the filtering and parsing of the data is done in the server side, the web browser is only used for the visualization.

Objectives

•The main objective is to create an interactive and suitable web interface to monitoring the network, in this way network administrators will be able to get a better comprehension and analysis of the data through flows records .

- Provide an easy way to access the application with controls to filter the data.
- Visualize and analyze the flow data through a single monitor without the needs of an additional computer software or hardware.

What is a flow?

NetFlow is a service introduced by Cisco that allow system administrators to have a detailed record of the events on their network. A flow contains aggregated information of a network connection. Some of the data provided by a flow are:

- Source IP address - It is an address assigned to a node (computer) that is sending a packet trough the network.
- Destination IP address - This address is used to know where is the data going to be send.
- Destination Port - The port is an integer value between 0 and 65535 used to specify in what point (service) a packet is going to arrive. Each well known service use a designated port value like 80, 23, 22 for the HTTP, Telnet, and SSH services respectively.

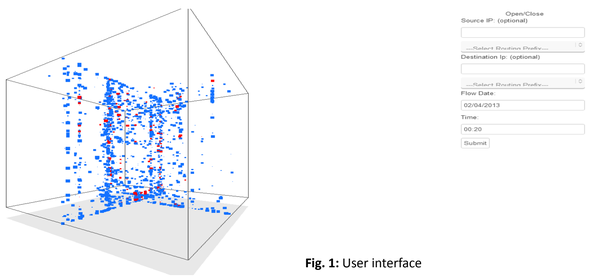


Fig. 1: User interface

As we can see in figure 1, we have the user interface where the cube is an additional component for a better comprehension of the data. The control panel at the right provide means to filter the data by source and/or destination IP address, or classes of networks, and by date and hour. Each point inside the cube represents a flow.

Problem

Given the complexity of monitoring network with large data sets, we are looking to provide efficient methods of analysis and visualization of network data through the web. This without the need of sophisticated hardware or software. In our method, we implement a web version of the Spinning Cube of Potential Doom.

Methodology

Graphic Interface

For the creation of the three-dimensional cube we have used the 3D JavaScript library three.js, this library allows you to create 3D objects, camera manipulation, lights, scenes and other features.

•The first step is to create a scene.
•Second, we create a cube geometry to represent the network data through its axis. At the *x* axis we have the destination address, in *y* axis we put the ports and at the *z* axis we represent the source address.
•Finally, we create a particle system inside the cube where each particles represents a network flow.

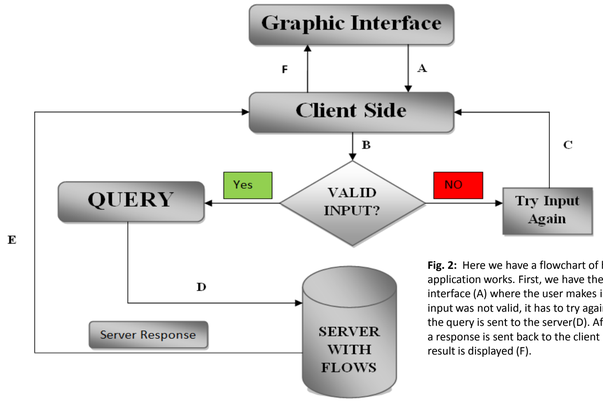


Fig. 2: Here we have a flowchart of how the application works. First, we have the graphic interface (A) where the user makes inputs (B), if input was not valid, it has to try again (C). If valid, the query is sent to the server(D). After execution, a response is sent back to the client (E) and the result is displayed (F).

The manipulations that can be made over the cube are rotate and zoom in/out. The zoom was implemented creating a camera object which allows the user to change the view perspective using the keyboard events provided by JavaScript. To rotate the cube, we use different mouse events such mousedown, mouseup and click. The control also allow you to select different flow files and apply some filters, and it was implemented using jQuery.

Controls

Client Side

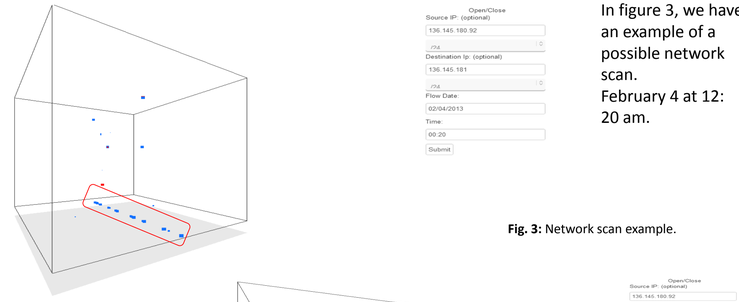
In addition to display, the web browser is also in charge to make input validation of the IP addresses , date and time, and to make the query to the server after the data is validated.

Server side

In the server side, we have a python script that verifies if the filename of the network flow requested is valid. If not valid, it returns a signal to notify the user, otherwise, it parses all the data requested by the user and returns it back to the client side.

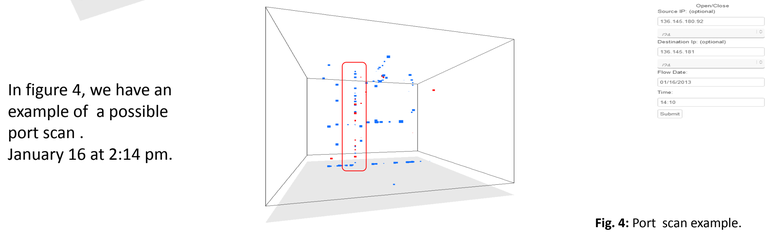
Results

As a result we have a working version of the desired application providing an interactive web interface in which we use different JavaScript libraries to display NetFlow records through the web browser. At the moment we do some manipulations over the cube such as rotating it, and zooming in or out. Also, it is possible to filter the flows by source and destination IP address and to select the division of the network classes, and the user is allowed to select date and time of the flow, providing a wide and structured way to search.



In figure 3, we have an example of a possible network scan.
February 4 at 12: 20 am.

Fig. 3: Network scan example.



In figure 4, we have an example of a possible port scan .
January 16 at 2:14 pm.

Fig. 4: Port scan example.

Conclusion

Given the structured implementation of this project and its successful results, we are contributing to innovation of applications directed to network monitoring through visual interfaces. In addition, we can easily manage and access the application. Based on the results that we have, we could use this application to monitor IP events to detect network or port scanning events, denial of services and other kinds of threats that often we see on a network. As a future project, this application will be attached as a network tool to the Toa Monitoring System.

Acknowledgements

First we would like to acknowledge to all the research lab members of Dr. José R. Ortiz Ubarri for their support. Also, we want to thank Dr. Humberto Ortiz Zuazaga for his comments and advices, and the HPCF for providing access to the University of Puerto Rico NetFlow data.

Bibliography

[1] *Thomas, J., Cook, K.: Illuminating the Path: Research and Development Agenda for Visual Analytics. IEEE-Press (2005)*
[2] Hwan Chang, Beom & Yoon Jeong, Chi.(2011, Feb 4). *An Efficient Network Attack Visualization Using Security Quad and Cube.*
[3] *Spinning Cube of Potential Doom*. Retrieved from www.kismetwireless.net/doomcube/
[4] Google, *Google Chrome experiments.* Retrieved from www.chromeexperiments.com
[5] Mr.doob, *Three.js Library.* Retrieved from https://github.com/mrdoob/three.js/
[6] W3Schools, *JavaScript reference*. Retrieved from www.w3schools.com/js/default.asp
[7] HPCF Version, Humberto Ortiz Zuazaga. Retrieved from http://plone.hpcf.upr.edu/Members/humberto/Wiki_Folder.2003-07-17.5848/TheSpinningCubeofImpendingDoom