Poster: *A network flows visualization framework and API for network forensics and analytics in the web*

Julio de la Cruz Natera
Department of Computer Science,
University of Puerto Rico,
Rio Piedras Campus
San Juan, PR 00936-8377
Email: jjcnatera@gmail.com

Ian Davila
Department of Computer Science,
University of Puerto Rico,
Rio Piedras Campus
San Juan, PR 00936-8377
Email: ian.davila@upr.edu

José R. Ortiz-Ubarri
Department of Computer Science,
University of Puerto Rico,
Rio Piedras Campus
San Juan, PR 00936-8377
Email: jose.ortiz23@upr.edu

Abstract— **High performance data networks such as Science DMZ networks are being deployed in research institutions all over the nation to provide high speed big data transfer among intra and inter institutional collaborations. The amount of network data generated by such networks is very costly to store and/or process to provide network security, network situational awareness, and network forensics. These research networks can not rely their security on traditional firewalls because firewalls tend deter the data transfer performance. The use of network flows have become more popular to aid provide network security to such networks. Silk, developed by CERT, has become the standard to store and manage network flows in the shell but the analysis of the data to find security anomalies becomes uphill because of the significant amount of line data results. Visualization analytics play a major role in the detection of events in big data as it has been in network visualizations. To help with the analysis, we present an API, that uses SILK as its base, with functions to filter network flows through a web interface and feed the output to web visualizations thereby (1) giving the power to non shell savvy system administrators to manage network flows data from the web, (2) providing a bridge between the processing of big network data and the visualization analytics researchers, (3) providing network analysis as a web service in the cloud.**

Keywords—visualization analytics; network monitoring; network forensics; network flows

I. Introduction

High performance data networks such as Science DMZ [1] networks are being deployed in research institutions all over the nation to provide high speed big data transfer among intra and inter institutional collaborations. The amount of network data generated by such networks is very costly to store and/or process to provide network security, network situational awareness, and network forensics. These research networks can not rely their security on traditional firewalls because firewalls tend deter the data transfer performance. The use of network flows have become more popular to aid provide network security to such networks.

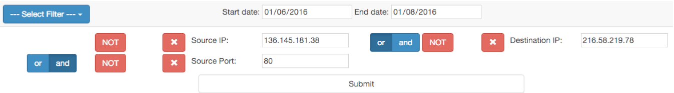
A network flow is a sequence of packets from a source computer to a destination, which may be another host, a multicast group, or a broadcast domain [2]. These network flows consist of a source and destination IP, source and destination port, the aggregated amount of packets sent, the aggregated amount of bytes sent, the input and output interfaces, among others. Network flows are used for network situational awareness, i.e. to keep track of what is happening in the network, and to detect network anomalies, and for network forensics [3].

The goal of this research is 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.

Our API uses AngularJS and Silk. AngularJS is a JavaScript framework which extends HTML attributes such that you can generate dynamic views in web-applications. SiLK, the System for Internet-Level Knowledge, is a collection of traffic analysis tools developed by the CERT Network Situational Awareness Team (CERT NetSA) to facilitate security analysis of large networks. The SiLK tool suite supports the efficient collection, storage, and analysis of network flow data, enabling network security analysts to rapidly query large historical traffic data sets [4].

II. Methodology

AngularJS was used to develop a function of the API that generates a dynamic GUI that allows the users to select among different network flow filters. The GUI permits to construct and remove the filters that they want to apply to the stored network flows data (See Figure 1). Once the filters are constructed by the user, the input of the user is translated into a query. The filters include the selection of a period of data that wants to be analyzed. Some simple examples of the filters that can be constructed are: all flows with source IP address 192.168.1.1, all flows with destination port 22, all flows with more than 20 packets, or a combination like: all flows with source IP address 192.168.1.1 and destination port 22 and more than 20 packets.



Start date: 01/06/2016 End date: 01/08/2016

Source IP: 136.145.181.38

Source Port: 80

Destination IP: 216.58.219.78

Figure 1. A simple filter constructed with the GUI.

The backbone server that receives the query, translates the query into the actual filters that are applied to the network data. The backbone utilizes the PySilk extension to retrieve the network flows stored in the file system, while the filters are applied to the data. The network flows that pass the filters are then constructed into an array of results in json format.

Finally, the results are returned to the GUI where the user can then access the results through the API and then either connect the results with a visualization for analytics, print them in plain text, or in an HTML. See Figure 2 for an illustration of the framework pipeline.

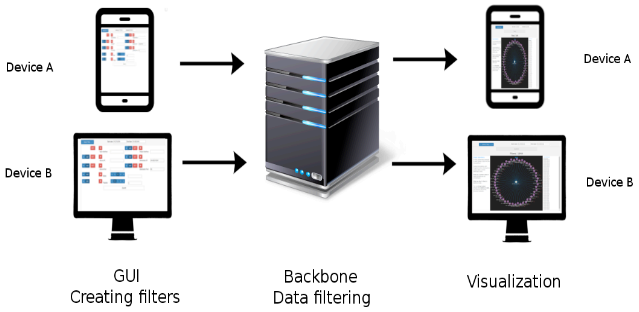


Figure 2. Illustration of the framework pipeline.

The results of this work is a network analysis framework with an API for Python developers that can be implemented with Python CGIs or Flask, and another for JavaScript developers. The API contains all the necessary functions to generate the GUI, to filter the network flows, and to access the results. Thus the visualization developers just need to create the visualizations and use the API functions to access the data results. Currently we implemented two visualizations [5]. The first visualization is a Force Directed graph that is used for finding botnets, and DDoS. The second visualization is a Tree Map graph that is used to find the top computers generating traffic.

III. Future work

Future work will focus on enhancing the API by adding more filters and more capabilities, like the ability to apply more filters to the already filtered data. Also, we will be working on getting new visualization tools that could be implemented and used with the API.

References

[1] 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.

[2] 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

[3] 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.

[4] SiLK. (n.d.). Retrieved January 28, 2016, from https://tools.netsa.cert.org/silk/

[5] Belmonte, N. "Javascript InfoVis Toolkit. ht tp." *thejit. org/about.*