Network Visualization Tool (NVT)

Overview and Background

When attempting to do forensics on a network pcap file, the analyst can easily be overwhelmed with the amount of data to examine. Tools like Wireshark and the command line tshark provide excellent methods to do in-depth examinations on network traffic. But, these tools come with a decent learning curve. Without understanding how to properly use Wireshark/tshark filters, it is difficult to get even a general understanding of the data.

The Network Visualization Tool provides a simple way to examine network traffic parsed from a pcap file to get a high-level overview of the types of traffic, the sources, and destination IPs involved, and traffic levels over time. The analyst can then use that information to delve further into the data.

Installation

The tool can be downloaded from the GitHub repository:

https://github.com/fredtheranger/network-visualizaton-tool

The following prerequisites must be met before running the tool:

- Python 2.6 or greater
- https://pypi.python.org/packages/source/i/ipcalc/ipcalc-1.0.0.tar.gz
    - Provides calculation of IP subnets
    - Extract tar.gz file
    - Python setup.py install
- http://webpy.org/
    - Provides a simple web framework to serve up the web application
    - On Ubuntu variants, use: sudo apt-get install python-webpy

Running the tool

- cd ${install}/python
- python –n 192.168.1.0/24 example.pcap
- python server.py

Browse to the web application

- http://localhost:8080/static/index.html

Code overview and methodologies

The code is divided into two primary modes: parsing the data and displaying the data. The parsing code uses the Scapy (http://www.secdev.org/projects/scapy/). Scapy is a python library that allows for manipulation of network packets. The code uses Scapy to parse the initial pcap file and examine the packet to determine the information of interest, including the protocol, the source and destination data, as well as the timestamp of the packet.

After using Scapy to parse out the items of interest, the tool inserts the data into a sqlite database. The sqlite database allows for more efficient serving of the data to the web application using JSON.

Sample PCAP Captures

The tool was tested with sample pcap captures from http://www.netresec.com/?page=PcapFiles

Visualizations

The visualizations were generated using the parsed pcap data, converted to JSON format, with the JavaScript visualization library D3.js. D3.js provides the foundation for creating interesting and dynamic visualizations in a variety of charts and layouts. Unfortunately, D3.js has a bit of learning curve to be able to go beyond simple line and bar charts and achieve more advanced charts such as treemaps and bubble charts.

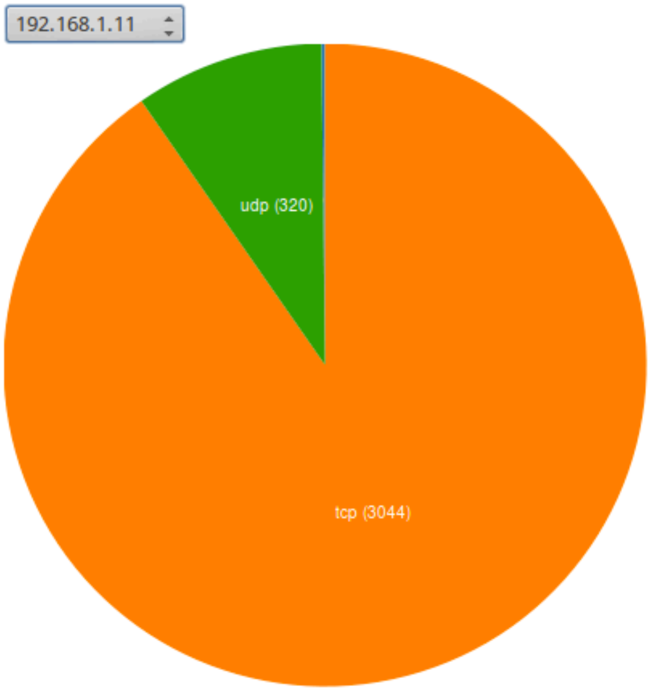

Figure 1: Pie graph of protocol usage

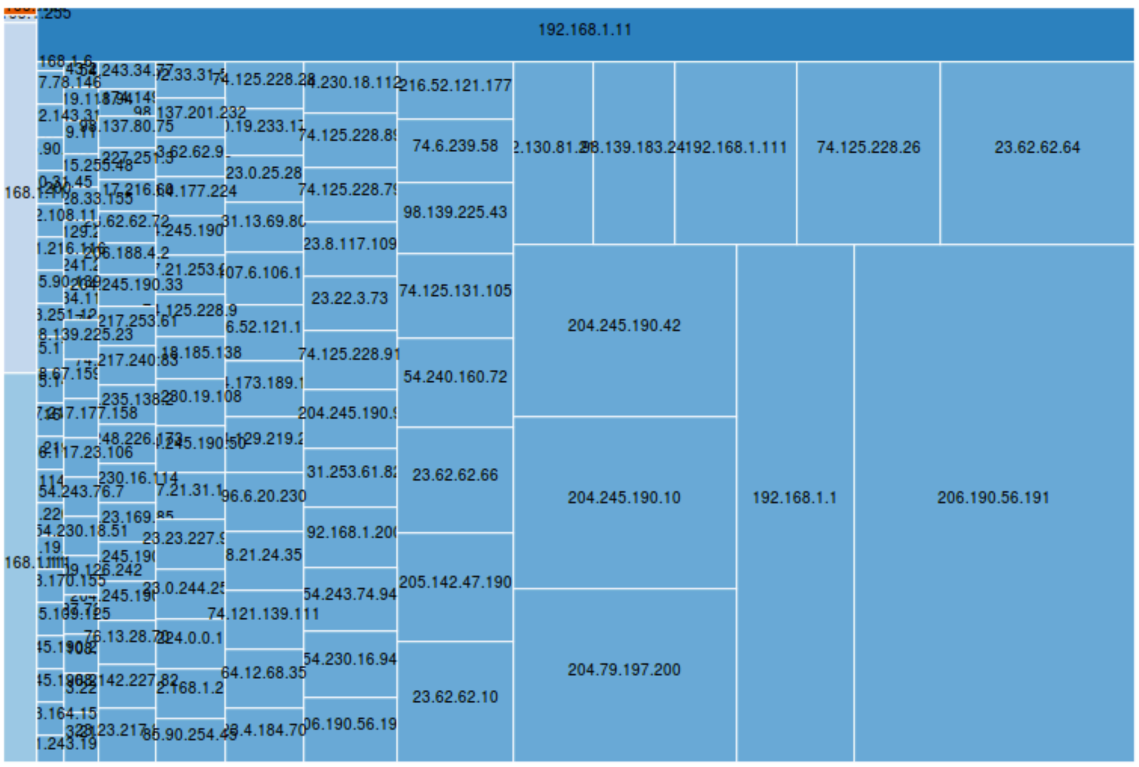

Figure 2: Treemap showing destination by source

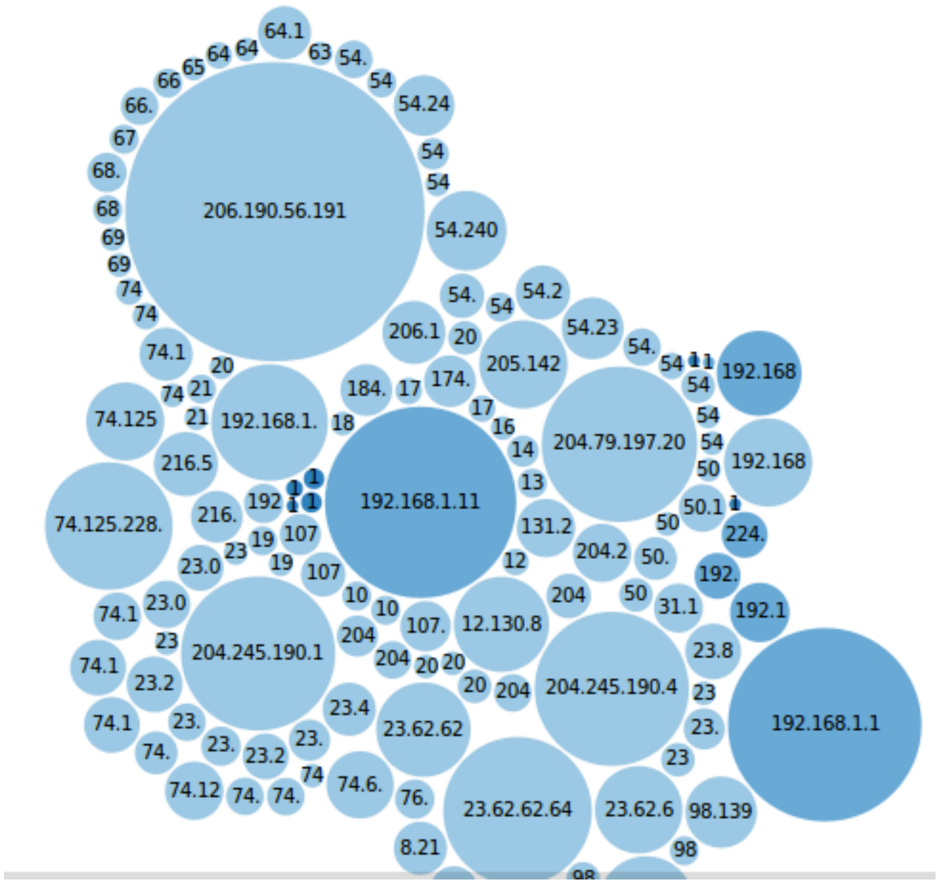

Figure 3: Bubble chart showing traffic by destination

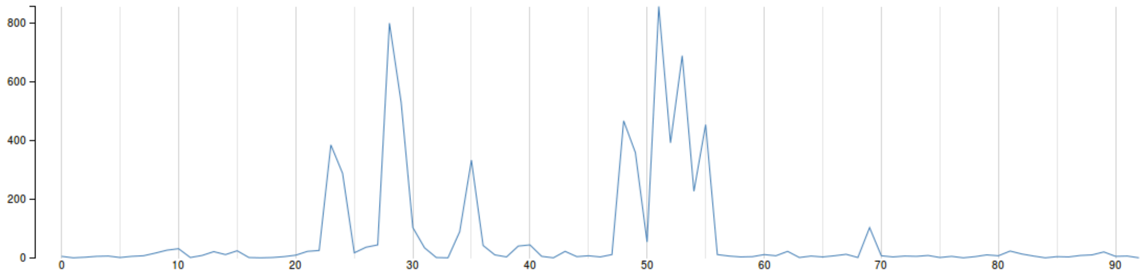

Figure 4: Traffic pattern over time