Dell SonicWall Visualization Report

Wanzhang Sheng
Kaiming Yang
Jie Gao
Xi Han

December 11, 2014

Chapter 1

Introduction

1.1 Project Overview

In this project, we used D3 to build stand-alone prototype visualizations with Dell SonicWALL firewall data that could be integrated into their Dell SonicWALL Analyzer product. After we learnt SonicWALL data format, we decided to use wireshark data which is similar to the firewall data of Dell SonicWALL products, and we generate dashboard, datamaps, and realtime map plot.

Analyzer

Figure 1.1 is the current visualization of data usage of Dell SonicWALL Analyzer. The analyzer streamlines and summarizes web traffic and network data. We generated visualizations that could be interated into the Analyzer product to show near real-time analysis of traffic and network data.

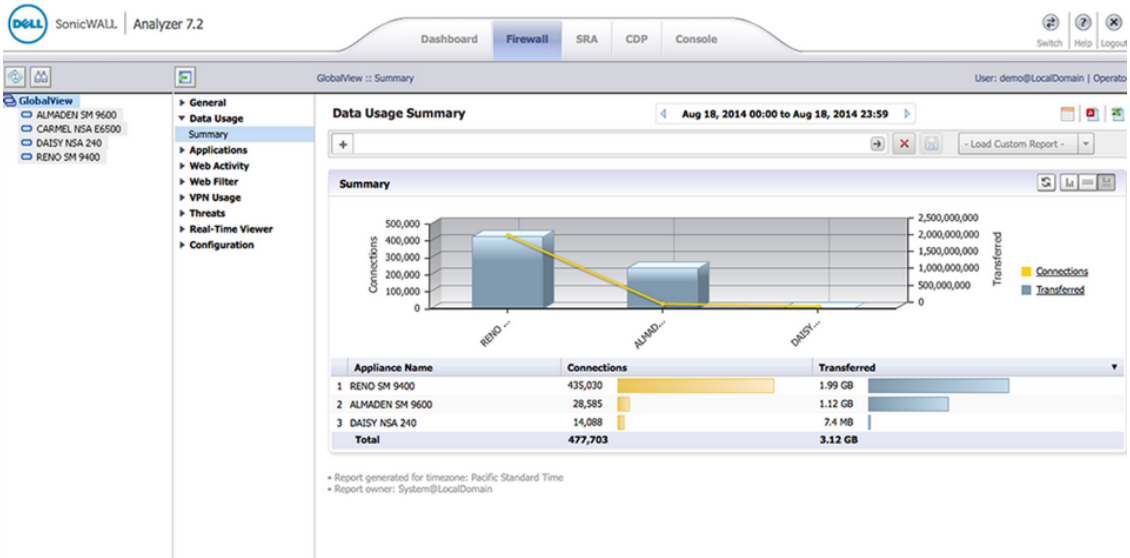

Figure 1.1: Analyzer

Transformation of Data

Figure 1.2 is the transformation of data, the data file size reduced from 1.4MB (left side) to 5.8KB (right side). This will significantly improve the network performance. The structure is just as what we used in D3, so much less pre-process in user's browser, which means it's faster.

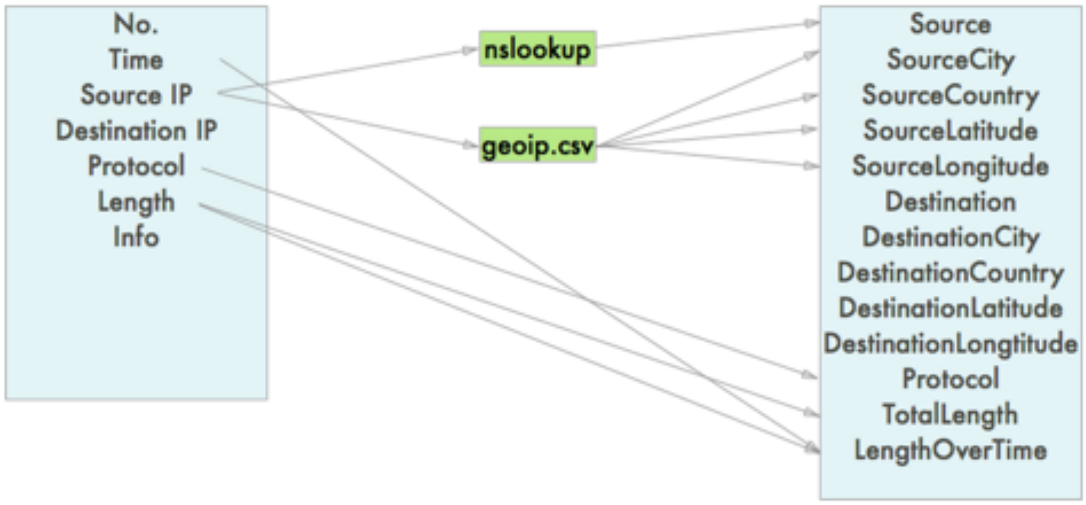

Figure 1.2: Transformation of Data Structure

Our Visualizations

Figure 1.3 lists our works, including the website we build for our projects and three visualizations. Using the dataset we generated from wireshark data, we created a datamap which shows the connection between source and destination nodes, a dashboard which contains an area chart and a relative sortable table, and a realtime map plot which visualizes network connections of realtime data received from our backend.

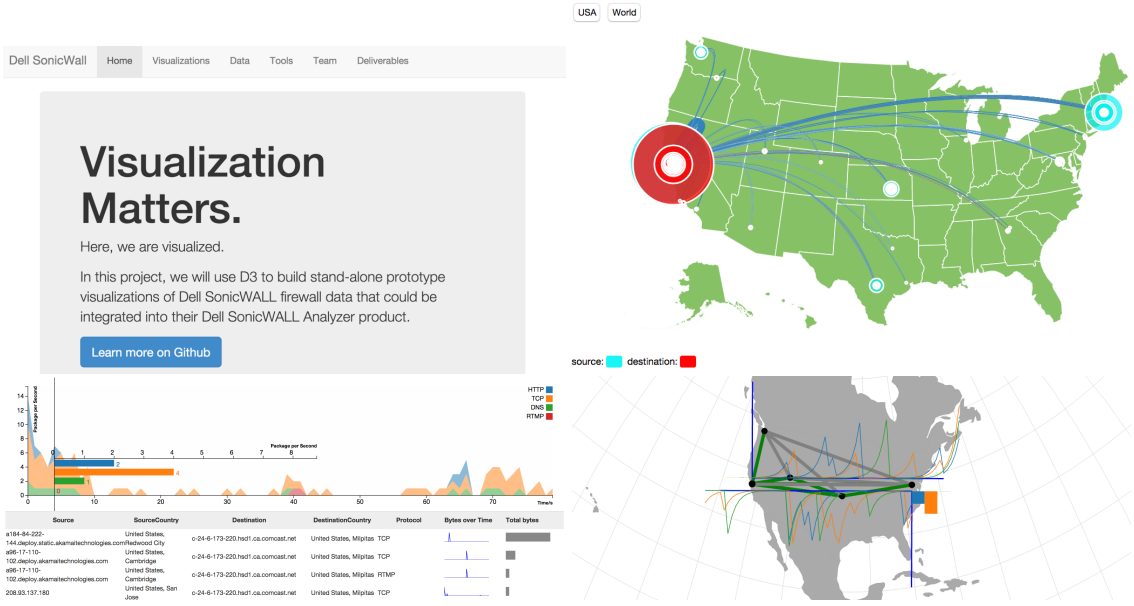

Figure 1.3: Our Website

1.2 Team Overview

1.2.1 Team Members

Our team members are Wanzhang Sheng, Kaiming Yang, Jie Gao and Xi Han.

Wanzhang Sheng is a Computer Science graduate student of University of San Francisco. He has worked on Ruby on Rails for over four years. Finished internship in Twitter working on Backbone.js.

Jie Gao is a Computer Science student at University of San Francisco. She has experiences in Java and C. She has built two-layer web service using Java.She used to study Electrical Engineering and has worked as a Information System administrator for over a year.

Xi Han is a Computer Science graduate student of University of San Francisco. He mainly focuses on data mining and data visualization on several fields using C#, python and javascript libraries. During his internship he was working on SQL error comparison visualization in Honeywell (China). And he also has experiences working on protein molecule visualization using OpenGL under his professor.

Kaiming Yang is a Computer Science student at University of San Francisco. He has experiences in many programming languages like C/C++, Java and Python. He used to study physics and worked a lot on scientific computation.

1.2.2 Sponsors

Our sponsors are Sophie Engle and Carrie GAtes.

Sophie Engle is an Assistant Professor in the Department of Computer Science at the University of San Francisco. Sophie Engle is also the Graduate Director for the Masters in Science in Computer Science (MSCS) program, and teach courses for the Masters in Science in Analytics (MSAN) program. She is the mentor and one of sponsors of this project.

Carrie Gates is the Chief Scientist at Dell Research. She received her PhD in May, 2006, and spent the last three years of her degree working at CERT, Carnegie Mellon University, where she performed research in network security. She have also worked on access control for WebSphere while at the IBM Toronto Centre for Advanced Studies (CAS). Previous to her PhD studies, she have six years of experience in systems administration and management. She is one of sponsors of this project.

Chapter 2

Specification

2.1 Requirements

Design and build a replacement of original

2.2 Design

The basic idea of our design is that visualizations begin with relativly simple and general information, the detailed information can be requested by some intuitive interactions. We came up with three visualizations, each of them is targeting a use case of firewall administration.

2.2.1 Dashboard

Dashboard is way for firewall administrators to grasp the status of a time span in the past. This visualization contains two views for same set of data: summary table and time plot.

The summary table shows the traffic over sources. Each row represents the traffic from a different source, with the domain of the source, geometry location of the source and the protocol it use. Each row also include a spark bar indicate the total traffic of the time span and a spark line indicate the traffic distribution over time. Only top few sources are shown to save screen space.

The time plot is a stacked area chart which shows the traffic of protocols overtime. Protocols are represented by colors, which indicated by a legend.

User can query more detailed information by some simple and intuitive interactions. For the time plot, clicking on an item of the legend should toggle the visibility of that item; reorder the legend items by drag and drop should also reorder the chart. When user move mouse over the area chart, the chart under the mouse should be high-lighted, and clicking mouse on area chart leads to a pop-up chart that shows the distribution and traffic of protocols at that time. For the summary table, click on title of each column should sort that column. User may also select one or more rows which would change the domain of the time plot.

2.2.2 Datamap

Datamap tries to visualize the traffic based on geometric locations. IPs are plotted on a map as a circle, where the size indicate the traffic volume and color distinguish the incoming and out coming data. Arcs on the map means the activities between IPs, where the thickness shows the traffic volume between the IPs.

2.2.3 Real-time Map Plot

Real-time map plot will help administrators to track events on real time, such as identifying DDOS attack. Like datamap, IPs and activities are visualized as circles and lines on a map, but size and thickness no longer carry information. Instead, the color of a edge would turn to green if there are any activities between two IPs and keep grey if there are no activities. When moving mouse over an edge, two real-time chart will pop-up and lined with the

edge to show the detailed traffic between two IPs in past few seconds. To get a better view of the real-time chart, user can click on the edge to rotate chart to right angle.

Chapter 3

Implementation

3.1 Results

Our main goal is to improve the sonic-wall product interface by making several responsive visualizations. The original visualization they use are not good enough, there are some problems like putting two charts in one coordinate, or using unnecessary 3D visualization. Our approaches are to fix the existing problems and to make it better.

In the final stage, we successfully make one combined area plot with bar chart attached showing selected details, an responsive table with sorting and filtering functions and a real-time geographical visualization to track the network traffic between different locations. These are all created using D3.

Besides, we did some extra works. Since there are functional and performance limitations using D3, we create another P5.js visualization library in order to make more flexible charts. Although those P5 charts we make are not in use currently, this library itself is worth further development.

We also create a tool for mapping IP address into longitude and latitude information, a data server to transfer any real-time firewall data to the visualization and a simulated data back-end to preprocess or cluster the data.

Current enhancement meets the need already, however, if we have time, we may refine our visualization to pursue more visual consistency by adding more design elements.

3.2 Obstacles

We face plenty of balks during the entire working period.

First of all, getting familiar with Javascript. Unlike web science students, we are basically writing projects using Java or Python, at the beginning we don't even know the very essential of this web language. We have to try hard, keep making new simple projects to get use to its syntax and remember the common APIs.

Then, since D3 is a mature visualization framework, we have to spend lots of time on learning its structure. On the contrary, P5 only provides us very limited graphical functions. If we want to make visualizations using P5, we almost have to start from nothing. These problems among two different tools are about trade-offs between efficiency and functionality.

Finally, turn the idea of real-time visualization into reality is incredibly hard. Finding a proper way to transfer the data between front and back end takes almost two weeks for us to achieve.

And we make introduction videos, we create website to put everything, every day we are learning new staffs.

3.3 Timeline

For the first several weeks, we are working on learning D3, making bar charts and scatter plots using sample datasets to enhance our skills on manipulating all the visualization tools. Then we start trying Tableau and Wireshark to explore the data which is similar to the sonic-wall data. After we understand the basic format of firewall data, we move on preprocessing the detailed information of it, for example, query the domain and geographical information and convert it into the format we can easily use.

Once we finish the data part, we break into four individual one to manage different tasks. Wanzhang Shen is working on the website, Kaiming Yang is making geographical visualization, Michelle Gao is refining the details and Xi Han is creating P5 components. These are the fundamental building blocks we are going to use in the following stages.

In the end, right after we finish the development of front end visualization, we move on generating a new back end to make our map visualization into real-time one. This is quite a new technique so it takes about three weeks to debug and test. Finally we get it work in the last week.

Chapter 4

Test Plan

Chapter 5

Deliverables

5.1 Primary Deliverables

Our main deliverables are our demo visualizations and the website we created for this project.

5.1.1 Demo Visualizations

Dashboard

Figure 5.1 aims to create an overview of the network traffic. We limited the result only shows a summary of top ten connections. It includes two parts. The top one is an area chart shows networking packages over time, separated by protocols. Users can click on it to show traffic details at the corresponding time. The bottom one is a sortable table. Each row indicates a different connection by source, destination and protocol, along with a spark-line to show the trend, and a bar to show the total bytes of the connection. By hovering mouse over a row, the area chart can change to the subset data correspondingly. And also, users are allowed to select one single row by a simple click or select multiple rows by holding command/option/shift key with a clicking to lock the dataset, then users can move their mouses to the area chart to inspect more details.

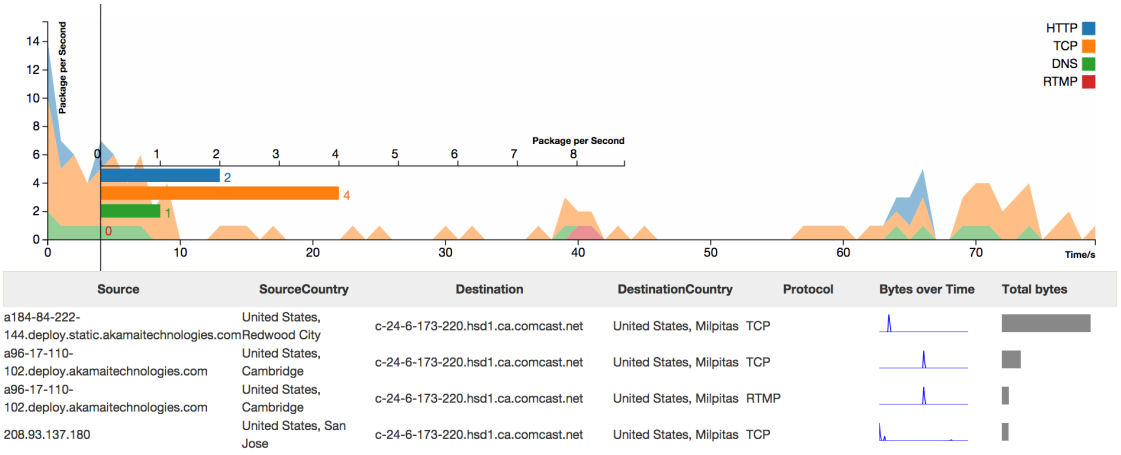

Figure 5.1: Dashboard

Datamaps

Figure 5.2 is a map visualization. It shows the traffic on a a map. User can switch map scope between USA and the world. Arcs between cities indicate connections and thicknesses indicate the volumes. Circles indicate the traffic volume of the city. Cyan for outgoing volume and red for incoming volume. Users can toggle circles by clicking on the legend.

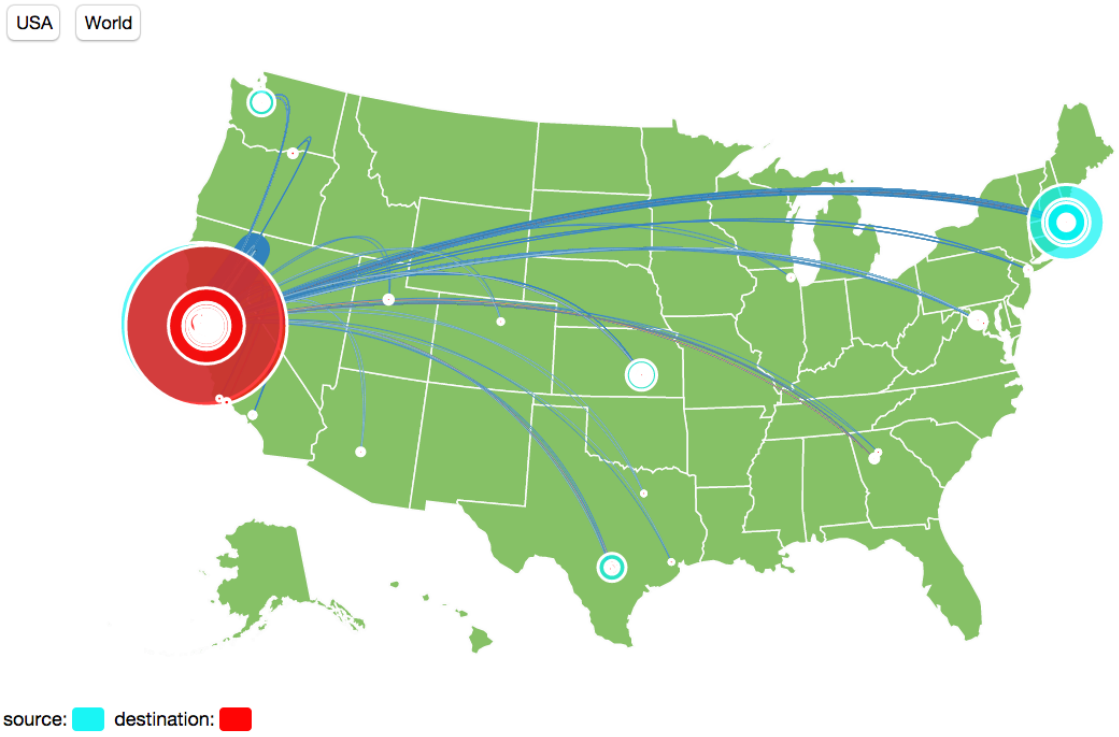

Figure 5.2: Data-maps

Realtime Map Plot

Figure 5.3 is a map visualization which keeps receiving realtime data from our back-end. Lines on the map indicate connections. By hovering on any line, users are allowed to inspect details of that connection with two line charts and two bar charts for both directions. And by clicking, to make it horizontal which is easier for users to watch the charts.

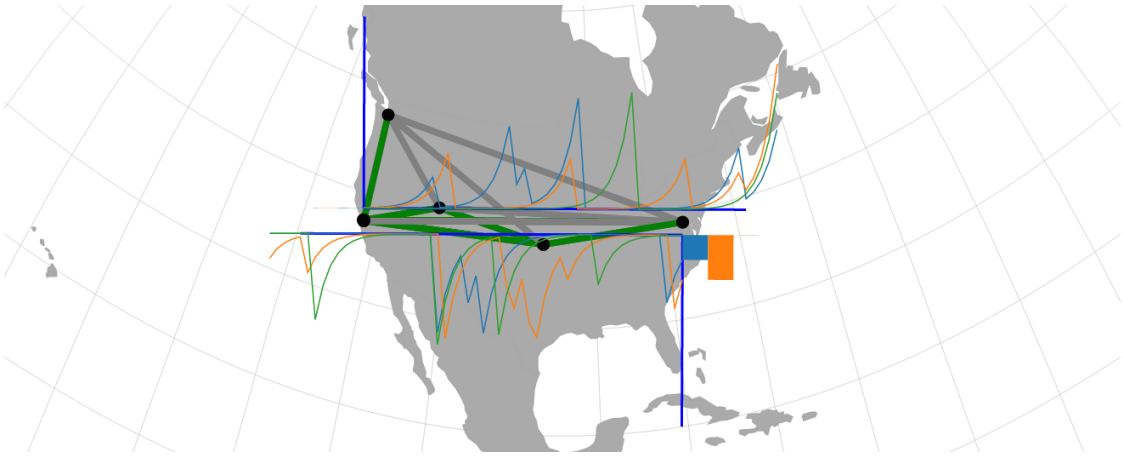

Figure 5.3: Realtime Map Plot

5.1.2 Website

Because our project is about visualizations, we created a website to show our work at http://sjengle.cs.usfca.edu/cs690-sonicwall/ with Bootstrap. There are several pages on it.

Home
This page shows a brief introduction to this project. Why we worked on this project and the basic timeline of the whole process.

Visualizations
This page lists all visualizations we created for this project, including all attempts before the final visualizations.

Data
This page shows where did we pick the data for our visualizations, and how we transformed it and mixed geography information to fit our needs.

Tools
This page lists main open-source libraries or visualization tools we used during the development.

Team
This page lists all team members and sponsors with avatars and biographies.

Deliverables
This page shows main deliverables including this report, usage videos and slides used in the final presentation.

Our works are mainly in page Visualizations. We have seven sections list different attempts. In most sections, every team member has their own attempts. By hovering mouse on any item, it will show a thumbnail of the visualization correspondingly. By this way, we can explore as wide as we could, to find the best way to visualize the data.

5.1.3 Code Files

```
/
├── index.html
├── visual.html
├── data.html
├── tools.html
├── team.html
├── deliverables.html
├── assets/ ..... Assets for pages above
│   ├── default.css
│   ├── *.png
│   └── *.jpg
└── public/
    ├── demo/ ..... Demo page
    │   ├── index.html
    │   ├── style.css
    │   ├── g1.js ..... Reusable area chart
    │   └── table.js ..... Reusable sortable table
    └── lib/
        └── sparkcharts/
            └── sparkcharts.js ..... Reusable spark chart
```

5.1.4 Source Lines of Code

We use a JavaScript tool SLOC to calculate source-line-of-code:

```
sloc . --exclude "(jquery|bootstrap|d3|p5|tip|modernizr|datamaps).*\.(js|css)|.*\.(csv|
    json)|nsa6600|xihan/lib"
```

This command means when it analytics files, it will exclude files whose filename or pathname matches the given regular expression. By this, we remove libraries or data we used or generated in the project to make the result more meaningful and convincing. And the result is in Table 5.1.

Table 5.1: SLOC Result

Physical	Source	Comment	Single-line comment	Block comment	Mixed	Empty	Number of files
9978	8226	621	384	238	62	1210	123

5.2 Other Deliverables

5.2.1 Data Files

```
/
└─ public/
   ├─ prepare.rb ..... Ruby script for preparing data
   ├─ geoip.csv (11K) ..... Geography information table
   ├─ sfgate.csv (1.2M) ..... Wireshark data
   ├─ sfgate_with_geoip.csv (1.6M) ..... Wireshark data with geography information mixed in
   ├─ sfgate_summary.csv (3.3K) ..... Wireshark data summary of top 10 connections in CSV
   ├─ sfgate_summary.json (12K) ..... Wireshark data summary of top 10 connections in JSON
   ├─ sfgate_subset.csv (8.3K) ..... Wireshark subset data with top 10 connections in CSV
   ├─ sfgate_subset.json (16K) ..... Wireshark subset data with top 10 connections in JSON
   ├─ nsa6600/ ..... SonicWall data
   │  └─ merged.csv (240K) ..... SonicWall data merged
```

5.3 Documentation

The usage videos in the Deliverables page are our user documentations, in which we descript our user interface, our main features, and how to use them. Those videos can also be found on YouTube:

Dashboard http://youtu.be/4Hakf4k9Lig

Realtime Map Plot http://youtu.be/KNbSRVzx_gY

Data Structure http://youtu.be/YU_TZmgiooQ

We also maintained README file for background description and resources links, and Wikis for development progress notes, on the Github repository.

Chapter 6

Conclusion

During this project, we have learned D3 and basic visualization implementation. We have built built some basic visualizations such as barchart and scatterplot before we implement geomap visualizations. We have learned that good visualization requires informative and interactive implementation based on good understanding of the format of dataset. We learn network data formats from Wireshark and SonicWALL data, and modify original dataset and generate useful and efficient dataset for visualization of D3. There are some improvements for this project. We have created visualizations of general network connection, and we could add some analysis in our visualization.There are lots of techniques we could learn in the future such as choosing color and generating creative and informative visualization.