FloVis: Flow Visualization System

Teryl Taylor
Dalhousie University
Halifax, NS, Canada
teryl@cs.dal.ca

Diana Paterson
Dalhousie University
Halifax, NS, Canada
paterson@cs.dal.ca

Joel Glanfield
Dalhousie University
Halifax, NS, Canada
joel@joelglanfield.com

Carrie Gates
CA Labs
New York
carrie.gates@ca.com

Stephen Brooks
Dalhousie University
Halifax, NS, Canada
sbrooks@cs.dal.ca

John McHugh
Dalhousie University
Halifax, NS, Canada
mchugh@cs.dal.ca

Abstract

NetFlow data is routinely captured at the border of a many enterprise networks. Although not as rich as full packet capture data, Netflow provides a compact record of the interactions between host pairs on either side of the monitored border. Analysis of this data presents a challange to the security analyst due to its sheer volume. We report preliminary results on the development of a suite of visualization tools that are intended to complement the command line tools, such as those from the SiLK Tools, that are currently used by analysts to perform forensic analysis of NetFlow data. The current version of the tool set draws on three visual paradigms: Activity diagrams that display various aspects of multiple individual host behaviors as color coded time series, connection bundles that show the interactions among hosts and groups of hosts and the netbytes viewer that allows detailed examination of the port and volume behaviors of an individual host over a period of time. The system supports drill down for additional detail and pivoting that allows the analyst to examine the relationships among the displays. SiLK data is preprocessed into a relational database to drive the display modes, and the tools can interact with the SiLK system to extract additional data as necessary.

1. Introduction

A security administrator or anaylst has a difficult job. He or she must examine vast amounts of network data, host and IDS logs and information from other sources in order to understand what is occurring on the network. Buried in this mass of data are the signs of intrusive behaviour. The tasks associated with this analysis can be time consuming and cumbersome as it is difficult to find patterns in the data and trends over time.

Recent research has focused on using visualization techniques to aid security administrators in gaining a mental image of how data is flowing along the network. Visualization is powerful because it allows us to view a significant amount of data at once and utilize our cognitive and precognitive abilities to find patterns much more quickly than sifting through raw packet contents.

The real challenge for using visualization is creating images that are more than just pretty pictures. They must provide insight into the underlying data to be useful. It is possible to show too much information in a picture which could serve to confuse rather than help.

Some years ago, we developed a number of static visualization approaches[13] that provided insight into both network wide and host behaviors. The work reported here extends these preliminary ideas to an interactive tool that is integrated with the SiLK netflow tools to allow the analysist a rich and flexible visualization capability that complements the textual features of the usual SiLK analysis.

FloVis is a suite of interactive visualizations designed to show various aspects of network data flow. Each image complements the others and shows the data in a different way such that if one image does not show a specific pattern another might.

This paper describes the FloVis application features and some of the key visualizations offered in the suite. In Section 2 we discuss our overall philosophy and our approach the the visual representation of flow data. This is followed by the introduction of our three key visual paradigms in Section 3. The representations capture time series behavior for a group of hosts as color coded blocks representing categorical aspects of hourly behavior, as bundles of flows from related sources to related destinations, and as detailed,

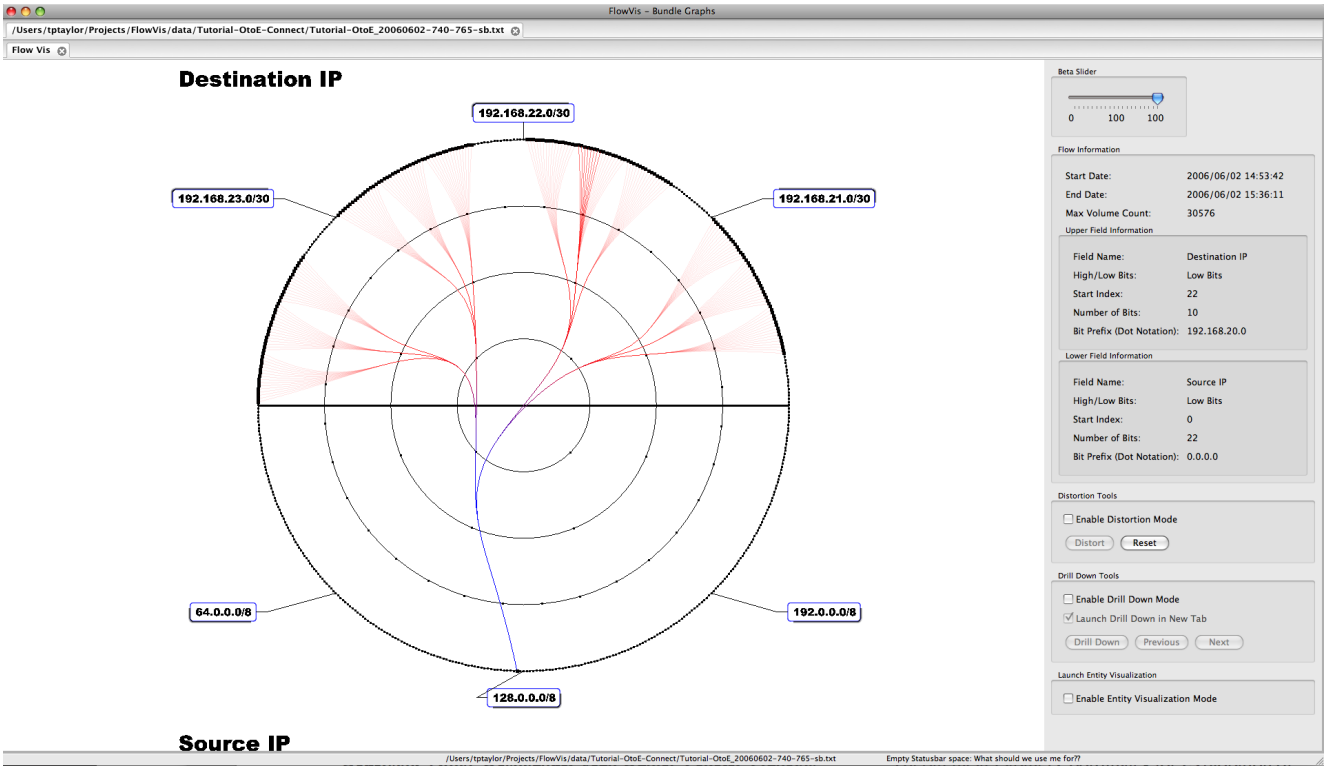

Figure 1. FloVis Bundle Diagram which shows data flows between IP entities.

pseudo three dimensional displays of per port volume activity over time. Using data available from a local network, we describe an analysis scenario in which the ability to visualize connection behavior leasd to the discovery of a compromised host in Section 4. Section 5 gives an overview of related work in visual analysis for security and the paper ends with our preliminary conclusions and out plans for future work in Section 6.

2 Approach

FloVis works along with the SiLK Toolkit[1] for network data analysis. The SiLK tools are used to filter the raw flow data and to aggregate the selected Netflow data into counted sets or SiLK bags. The bags contain counts of flows, packets, or bytes and can be indexed by IP addresses, ports, protocols and other scalar fields contained in flow data. They can also be indexed by composite vaules composed of up to 32 bits taken from any two scalar fields of the flow records. These can be used to indicate connections at the subnet to subnet (/16 to /16, for example), subnet to host (within a given subnet, /24 prefix to hosts in a given /24, for example) or host to host (within two prescribed subnets, hosts within a given /8 to hosts within a /24, for example). Bags are typically accumulated on a per hour or per day basis and contain a volume measure associated with the index. Volumes are usually counts of flow records or sums of the packet or byte counts contained in the records. In a production environment, hourly bags for various fields of interest would be created using command scripts that are scheduled for periodic execution. The resulting bags are then formatted into a data format readable by the various FloVis visualizations. Some visualizations utilize an SQL (MySQL at present) database for its quick and powerful queries while others read from the processed bag files directly. As the system progresses, we hope to use the database wherever possible because it buffers the visualization from direct dependence on the SiLK Tools, allowing data form other sources to be treated in similar ways.

FloVis itself is a client based application designed to run on a client workstation. It's built in OpenGL and utilizes a platform independent windowing system which should allow it to be built on most operating systems. FlowVis currently contains three different types of visualizations: Flow Bundle Diagrams, the NetBytes Entity Viewer and an Existence Graph. Each presents different aspects of host or network behavior, but they share linkages through the database and, ultimately, through an underlying SiLK repository. The approach allows explicit interactions among the differ-

[1]The SiLK Tools are available as open source under the GPL from `http://tools.netsa.cert.org/silk/index.html`.

ent views, allowing a interesting region of one visualization to trigger visualizations of related behaviors. The underlying philosophy of the system is to aid the analyst in developing insight into the behaviors that are seen on the network and hosts being monitored. We assume that the analyst is already familiar with most aspects of intrusive behavior and is familiar with the command line operation of the SiLK tools allowing us to translate drill down requests that go beyond the precomputed summaries into scripts to extract additional data from the SiLK (or a similar) repository and process it for either visualization or subsequent, manual or automated, analysis outside the FloVis system. The scripts will be parameterized and can be saved in a library for subsequent execution or modification as circumstances dictate.

3 Sample Visualizations

3.1 Bundle Diagrams

The Bundle Diagram as shown in Figure 1 focuses on displaying Netflow connections between entities (i.e. hosts or subnetworks) on a network. The knock against current applications that visualize network connections is that they can have significant visual occlusion when showing large datasets. The Bundle Diagram attempts to mitigate this by bundling connections [7] together and utilizing node aggregation to limit occlusion. Figure 1 shows a set of circular rings. The outer ring contains 512 points. Each point represents the highest 8 bits of an entity's address (could be an individual host or a subnet) on the network. The line dividing the circle represents a network border where netflow data crosses from (for instance) an internal network to an external network (or vice versa). Connections are represented by B-spline curves connecting points (entities) on the circle. Colour is used to represent flow direction (data flows from blue to red) while transparency is used to indicate flow volume. The more opaque a connection line is, the higher the percentage of the maximum volume traffic is running along it (for a specific time period).

The user can choose to loosen the bundles by using a slider control as shown in Figure 2. Bundle loosening straightens the bundles essentially separating them so the user can get a better look at the individual netflows.

The inner rings of the bundle diagram play an important part in the overall visualization. They create a radial tree layout for the underlying data source which facilitates the ability to drill down in the data. The bundle diagram supports a drill down mode as show in Figure 3. As seen from the Figure, the diagram activates a set of transparent green ovals overtop of points on the inner rings. This allows the user to select a specific branch on the diagram (highlighted in a purple ring in the figure). Clicking on a specific branch drills down in that branch launching another bundle visualization with that particular branch taking the full portion of the semi-circle. Drilling down essentially means to slide the highest 8 bits of the entity's address down the address by a number of bits (which is determined by how far along the branch the user clicks in drill down mode). This reduces the number of IP addresses in any specific node aggregation and provides more details about individual hosts (see Use Case section for an example). The drilled down visualizations can be launched in place over the existing diagram or in a separate tab to retain a historical context. Hierarchical tabbing is used so that the user can look at several different bag files while keeping the drilled down visualizations organized. Tabbing also facilitates a dragging feature where by two diagrams can be compared on the same screen for further analysis.

Another interactive feature of the Bundle Diagram is the ability to linearly distort [11] points on the circle as shown in Figure 4. This allows users to focus on key points on the circle which might show interesting behaviour while pushing aside points that are not that interesting.

One last point that should be touched on is bundle labeling. Labeling can be one of the most important yet difficult aspects of creating good visualizations. Without decent labeling, a great visualization can be meaningless. However, too many labels can create occlusion issues and confuse the user more than help them. In the Bundle Diagram, we took a more guided approach. We do not label every point around the circle, rather we label a set of strategic points around the circle. IP addresses are organized in order around the circle so the user can infer the pattern. However, the user also has the ability to add extra labels to the visualization by right clicking on a point on the circle. If there are many labels in a particular region, the visualization will push the other labels out of the way to create as much free space for the new label.

In terms of data input, the bundle diagram visualization takes a daily connection bag as its input. Connection bags contain information about the host to host connections occurring during a particular day and also can relay byte, flow or packet count information about the connection.

The real value of the bundle diagram is that it is able to show which entities on the network are talking to what other entities and gives a good idea of how data is flowing. This is important if, for example, we see some strange communication behaviour in which one machine is doing a scan of several IP addresses or an IP address which is not supposed to exist on the network is suddenly generating traffic.

3.2 NetBytes Viewer

The NetBytes Viewer [17] is a complementary visualization to the Bundle Diagram. While the Bundle Diagram focuses on interactions between entities, NetBytes Viewer

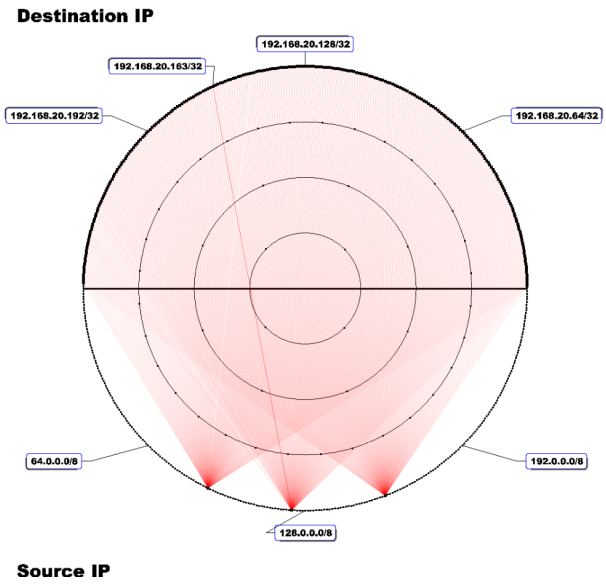

Figure 2. Loosening the bundle for a more detailed view.

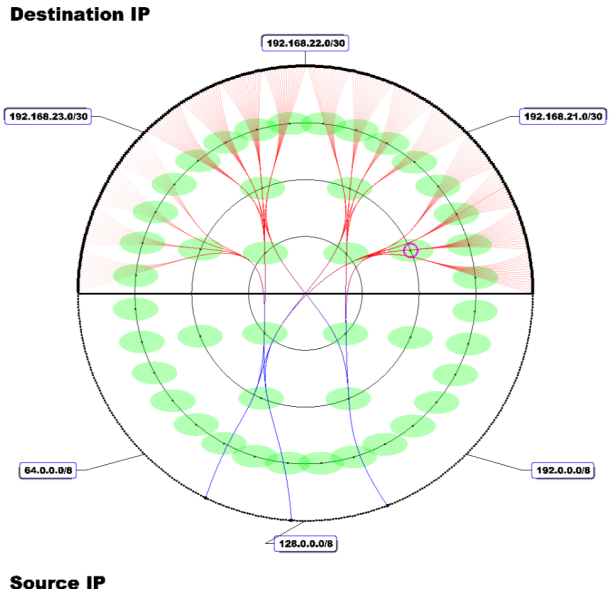

Figure 3. Bundle drill down mechanism.

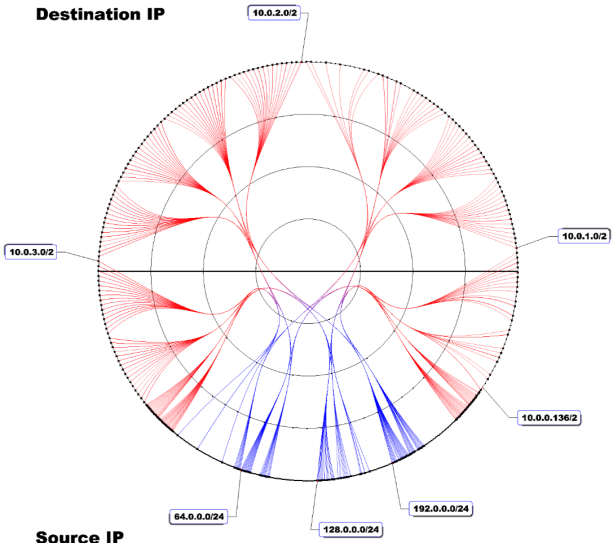

Figure 4. Linear distortion puts focus on key points on the circle.

focuses on the flow of data into and out of a single entity and visualizes flow volumes related to that entity. It utilizes a 3D impulse graph with a time dimension, a port or protocol dimension and a volume dimension as shown in Figure 5. This creates a ”picket fence” effect in which each line of colour shows the Netflow volumes for a specific port (or protocol) on an hourly basis over an extended time period (weeks or months). Such a visualization gives historical context to the port/protocol traffic without having to use animation. Animation relies on the user's short term memory to find intrusive patterns in the data meaning the user could forget or miss content during playback. Animation can also cause change blindness [16].

NetBytes Viewer can either be launched by clicking on a host in the Bundle Diagram or separately by providing information to a configuration dialog. The user has the option to look at volume going to or coming from a host along with selecting source or destination ports. Furthermore, the user can select a date range to look at the data for the host as well as filtering the data based on protocol. To facilitate powerful queries quickly, NetBytes Viewer is built on top of an SQL database which is loaded with port and protocol bags for individual hosts or subnetworks.

The problem with 3D visualizations is that they tend to suffer from two issues. Firstly, is data occlusion. With a 3D environment, data will inevitably be placed behind more data and possibly be out of the user's view. Secondly, putting a 3D space on a 2D surface can create depth perception issues and a loss of head parallax [18] whereby the user is unable to decipher exactly where a data point is located on the graph. To combat these issues, NetBytes Viewer has

a few important interactive features as described below.

First, NetBytes Viewer allows the user to interactively rotate the 3D impulse graph with the mouse. This enables users to look at the data from all angles reducing occlusion. For instance, looking down on the visualization from above can create a compelling image of port traffic patterns.

Next, NetBytes has a selection mode which allows the user highlight a point on the Volume vs. Time axis or the Volume vs. Ports/Protocol axis and view them in a set of 2D graphs on the right hand side panel of the application also shown in Figure 5. In the Figure, NetBytes is highlighting a large block of traffic on port 3306 and this data is shown in the image on the upper right hand corner of the viewer. A similar highlight is available on the Volume vs. Port/Protocol axis for a specific time. The corresponding image is available in the lower right hand corner of the application. A slider is used to move the highlight across the axis and there are two "snap" buttons (up and down) which snap the highlight to the next port (or time period) with data making it easier for the user to get a detailed view of the data he/she wants to analyze.

The 2D images are useful, but they can be too small to do any real detailed analysis. As a result, a swapping feature was added to the application. The swapping feature allows the user to swap one of the 2D views with the main 3D view as shown in Figure 6. All interactive features are still available after a swap has occurred.

Another interesting feature is the ability to select upper and lower boundaries on both the time and ports/protocol axes in order to zoom in on a smaller portion of the graph. As shown in Figure 6, the user can use a pair of sliders (one for the upper boundary and one for the lower boundary) to highlight a section of the graph. This can be done on both 2D graphs simultaneously. Once the boundaries are set, the user can relaunch the visualization in another tab with the new smaller axes. This is handy if there is an area of the graph that the user wants to focus on. Tabs were used so the user can go back the way he/she came and set the boundaries on another portion of the graph. Hierarchical tabbing is again utilized so that multiple hosts can be loaded in the tool at once.

NetBytes is good at giving a historical overview of the traffic into and out of an entity. This makes spotting traffic patterns very easy. For instance, if the user is looking at the graph of an email server, it will have traffic on the main ports that are used for email services. If traffic is seen on strange ports, or traffic patterns change over time, it could be an indication of some intrusive behaviour.

3.3 Existence Graph

We need to write a few paragraphs on the existence graphs....

4 Use Case Scenario: Identifying an Intrusion

In this section we describe a way in which we used FloVis to identify an intrusion of a host on a private network which had been running the SiLK Netflow collection for a little over a year from February 2006 to March 2007. When analyzing a large dataset it can be a daunting task to know where to begin. For this use case we focussed on the scanners in the data. We went through the data with the SiLK tools and classified any external host that had attempted to contact more than 700 internal hosts as a scanner. This is a very coarse form of classification but is suitable for our purposes. From this classification we generated a set of connection bags for these scanners to see who they were attempting to contact within the internal network. One of these visualizations is shown in Figure 1 on page 2. This Diagram shows all the scanning activity for the day June 2nd, 2006. One interesting characteristic of a scanner is very evident in this diagram (hint: look at the transparency in the connections). Scanners send out small amounts of data to each IP address they try to contact. Most of these connections are benign as either the contacted host does not exist or sends a reset. On hosts they compromise, they send a request, get a response, and then send a payload to infect the host. This typically means a significant increase in byte transfers for hosts that get infected. Using transparency to signify percentage of flow or byte transfered therefore helps to highlight scanner connections that may involve the transfer of a payload. Taking this theory to the dataset, we found what we were looking for on June 29th 2006 as shown in Figure 7. On this day there were three scanners, but only one connection (on the right hand side of the diagram) saw a significant amount of data transfer. Let's try to drill down on the bundle by turning on the drill down mode (this is shown in Figure 3 on page 4). The results of the drill down are shown in Figure 8. Adding a label to the darkest connection shows that the host with the possible infection has IP address 192.168.20.163. Loosening the bundle as in Figure 2 on page 4 to see exactly where the connection came from. To verify whether we have an intrusion, let's launch the NetBytes Viewer to take a look at the port activity for host 192.168.20.163. Let's look at 2 months (June and July) worth of data on the host, and filter to look at TCP traffic to the host on the destination ports. Visualizing this results in Figure 5 on page 6. Doing some analysis, we see as in Figure 6 that there is a spike in data on port 3306. Interestingly, As seen in Figure 6, we see that this data spike on the port started on June 29th (the same day of the data transfer from the scanner) ending on July 10th. Next, let's take a look at the data transfer to the host filtered by UDP with the destination ports as seen in Figure 9. Here we see another interesting spike. It turns out that this spike is on port

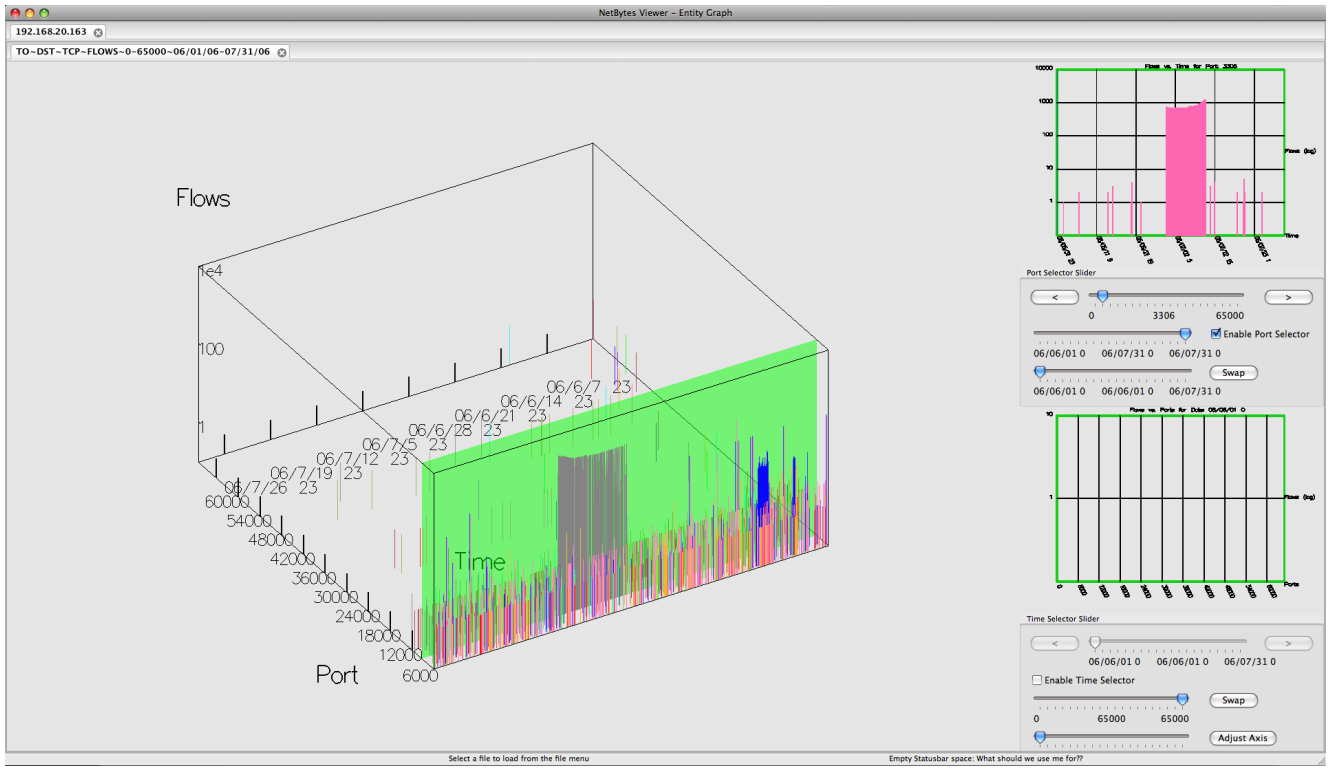

Figure 5. NetBytes Viewer selection mode. A green transparent bar highlights the Volume vs. Time axis for a specific port.

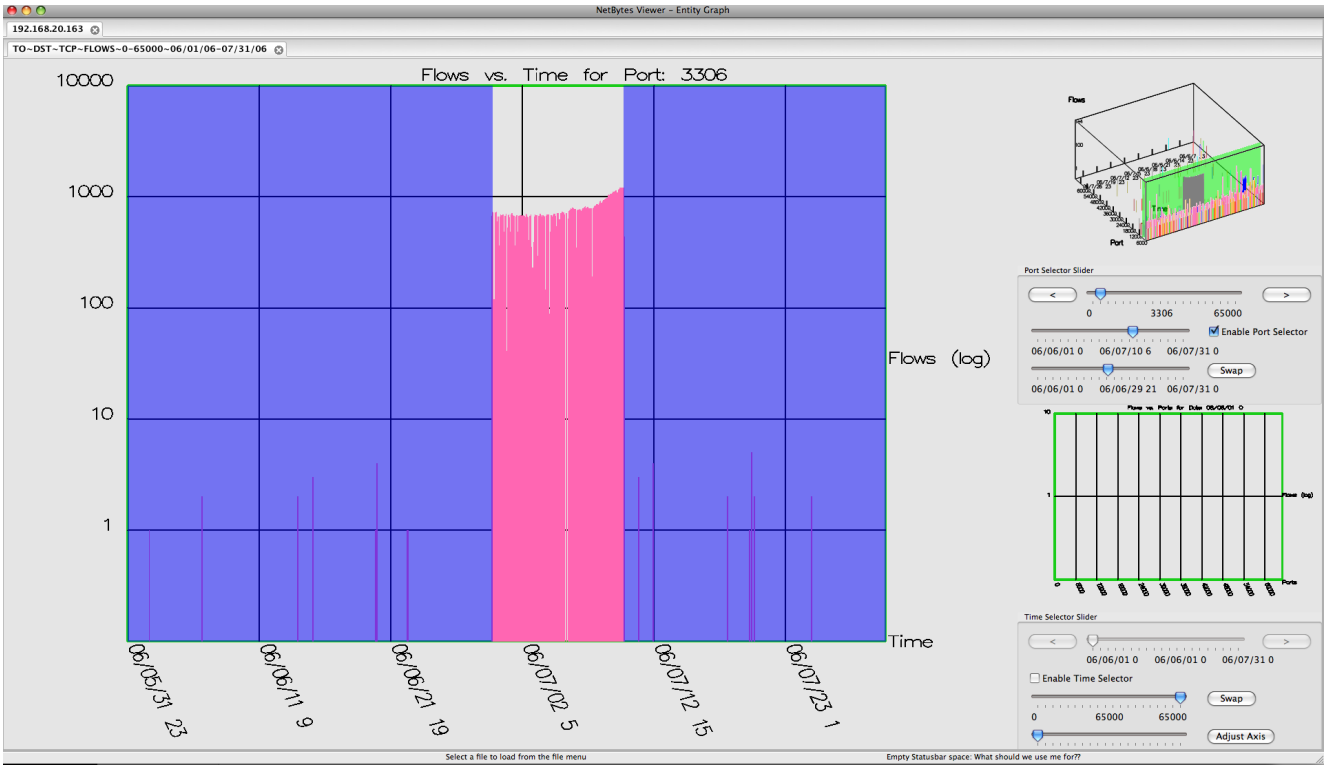

Figure 6. Highlight the upper and lower boundaries of the time axis to zoom in on the data.

137 and occurs between June 29th and July 10th (the same length of time as the TCP spike). Some quick investigation on the Internet on ports 3306 and 137 indicates the likely scenario. Port 3306 is a popular port for communication with a MYSQL database and port 137 is used for NetBios communications on Microsoft hosts. MySQL systems with weak passwords can be susceptible to a mysql botnet attack where the bot tries to logon with root access to the server and drop a payload onto the server. Port 137 may in fact be a cover for transferring some sort of data between the scanner and the localhost. Without the payload packet information it is difficult to say exactly what happened but this machine has exhibited warning bells that a security administrator should look into. Further investigation of the port activity on the host shows a great deal of traffic on ports 80 and 443. This machine was likely some sort of E-commerce website making it highly reasonable that a MySQL server was running on the machine to facilitate transactions for the site.

5 Related Research

There are an abundance of security visualizations available that target all different aspects of network data analysis [3] [6] [12] [1]. The ones discussed below are the most relevant to this project.

Visualizations such as VisFlowConnect [19] and RUMINT[2] use parallel axes to show connection flows. Hosts are represented as points along a parallel axis and each axis represents an internal or external network. Connections between hosts are represented by straight lines and the time dimension is shown using animation. These visualizations do a good job of creating a mental image of traffic flow; however, they are susceptible to data occlusion under high traffic scenarios and do not relate traffic volume information. The FloVis bundle diagram visualization helps to mitigate these issues by bundling connections together and providing node aggregation and drill down capabilities. Furthermore, the application also utilizes transparency to indicate volume which has proven to be powerful under certain circumstances.

Flamingo [15] takes the parallel axes concept into a third dimension. It uses two quadtree squares in a cube formation to represent the individual IP addresses and subnets in a network. Connections are again represented by straight lines from one quadtree to another. This allows the application to visualize many more connections but creates the 3D issues of occlusion and depth perception as described earlier. Flamingo can also visualize traffic volume per IP by using a

[2] http://www.rumint.org

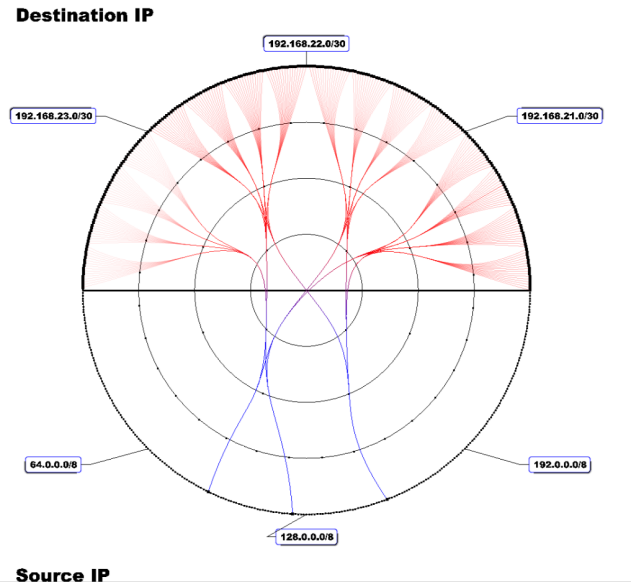

Figure 7. Bundle Diagram of 3 scanners with one intrusion. June 29th, 2006.

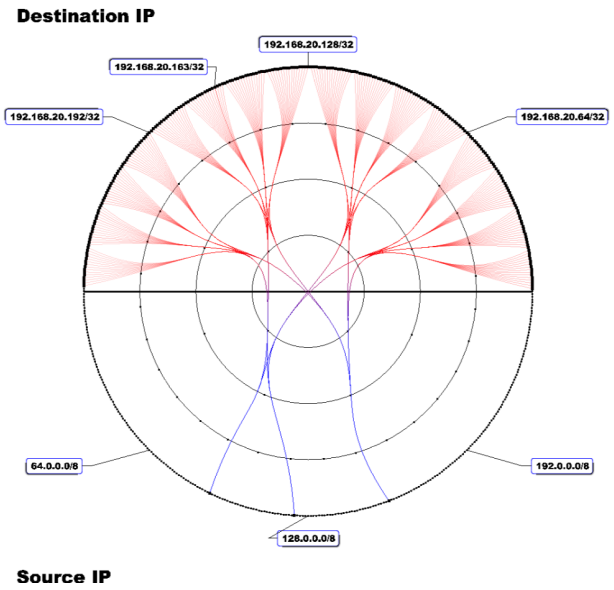

Figure 8. Results of the drill down shows host 192.168.20.163 with possible intrusion.

bar graph within the quadtree structure with bar height represent flow count. Individual port traffics are shown by rendering several quadtree square layouts at different heights along the cube. Volume is again shown using bar height. Flamingo helps to show the need for multiple visualizations in network security analysis, but at times tries to display too much information on a single image.

Another popular technique for showing data flow on a network is to use link-node graphs. In this situation, hosts are represented by nodes on the graph while connections are represented by edges. These types of graphs also suffer some occlusion issues and start to become confusing as more nodes are added. VISUAL [2] is an example of a node/link graph which works well for small networks. It encodes host volume flows and port information inside each node. FlowVis builds on this by providing the NetBytes Viewer to give detailed port information.

In one of the better named visualizations, The Spinning Cube of Potential Doom [10] takes a 3D approach to connection flows. It uses a cube, where the X axis represents local IP addresses, the Y axis global IP addresses and the Z axis is applied to ports. Connections are drawn in space using small colored glyphs. Color is used to distinguish between successful and unsuccessful TCP connections and animation is used to represent time. This visualization shows some interesting patterns that are signatures for intrusions. It is a novel technique but suffers from depth perception issues and does not have many interactive features.

The next two visualizations use a scatterplot technique to visualize network traffic and volume. NVisionIP [9] visualizes an entire class-B IP network on a single screen. The overview screen contains horizontal and vertical axes where all subnets of a network are listed along the top axis while the hosts in each subnet are listed on the vertical axis. Each host is colored based on some characteristic of interest which include traffic volume, number of flows or flows on a particular port. Animation is used to show traffic activity for a given period of time. Users can select a region of the overview screen to launch another window which provides more detailed information about hosts in the selected region. Each host is represented by two bar charts. One chart displays the traffic on a number of well known ports while the other shows traffic on all other ports. Color is assigned to traffic on different ports to make it easier to compare flows of interest. This visualization has some interesting drill down features to get a more detailed view port and traffic volumes but it relies on short term memory to see patterns in the data. The NetBytes Viewer over comes this by showing time along a third dimension.

Portvis [14] is another application that capitalizes on the scatterplot design. It utilizes three different displays to visualize TCP traffic. The first display is essentially a scatterplot with the horizontal axis representing time while the vertical axis represent port aggregations. Color is used to represent port activity levels during a particular time period. A selector is used to select a specific time unit for which the data is rendered in another visualization. This visualization is a 256x256 scatterplot where each point represents one of the 65,000+ ports. The grid can be magnified in certain

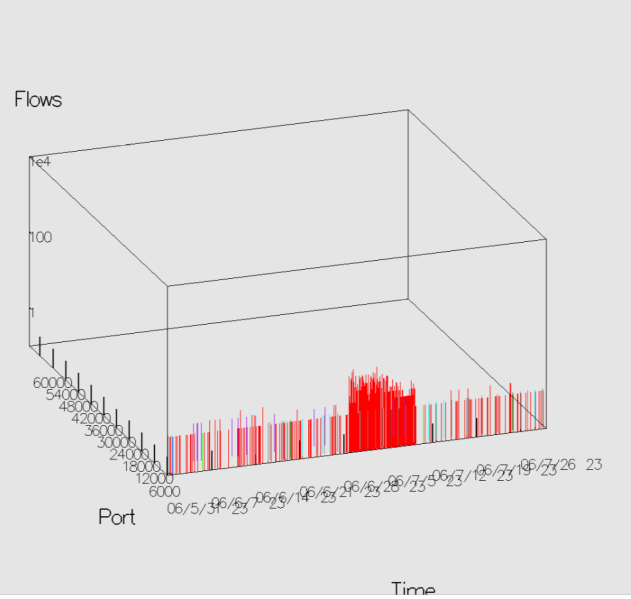

Figure 9. NetBytes Viewer for host 192.168.20.163 filtered by UDP (data to host with destination ports).

areas to get a more detailed look at specific ports. These ports can then be visualized in 3D bar graphs. Portvis is an interesting visualization for showing ports; however, time dependent data is defined very coarsely (2048 port buckets) making finding patterns over time difficult. Furthermore, there might be some information overload under conditions of extreme port traffic when the port grid is filled with many colors. NetBytes Viewer improves on this by showing time as a third dimension. This allows the user to see trends on ports more quickly without the need for port volume aggregation.

Komlodi et al.'s [8] work is quite interesting as it allows the user to tailor a glyph-based visualization to their own needs by mapping data variables to visual glyph attributes (like color, size, position, etc.). Visualizations can be rendered in both 2D and 3D forms. This user centric approach underlines the need for visualization tools to be flexible.

Time-based Network Traffic Visualizer or TNV [5] takes a focus + context [4] approach to visualizing flow data. TNV uses a table, where the columns represent time periods and the rows represent hosts. Time periods with more focus show more detailed information (such as connection information and port activity) than contextual columns. TNV shows that having historical information is essential in detecting abnormal patterns in the data. However, it does have some scaling issues to large networks.

6 Conclusion and Future Work

With the amount of data a security administrator must sift through on a daily basis, visualization is a becoming more of a key cog in analysis for network security. In this paper, we presented a new suite of visualizations called FloVis which is designed to show several views of network data in order to facilitate network data analysis. With FloVis, a user can view host to host or network to network interactions using Bundle Diagrams; entity-based volume information using the NetBytes Viewer; and role-based host information using an Existence Graph. These visualizations complement and interact with one another to create an application suite useful in the hunt against intrusions. FloVis is built on top of the SiLK collection and analysis tools. Such tools provide powerful data processing capabilities and allow FloVis to be a highly flexible platform.

FloVis is just in its initial stages of development. In the future, we plan to continue to evaluate new visualization approaches that will complement the existing visualizations in the current platform and provide more insight to security administrators. In providing new visualizations, we are going to investigate statistical approaches that might help bring important information to the attention of the user. Along with new visualizations, we are looking at adding more features to the existing views as well as further integration so transitions between visualizations are more seamless. We also want to integrate the tools further with SiLK so they can return the underlying SiLK datasets on demand. The visualizations are powerful tools but we need to provide a way for the administrator to get back to the raw data when necessary.

We continuously look for feedback on the tool and a user study is something that we would like to conduct to get more insight on how to improve the application. Along with a user study, we are trying to get access to more datasets. Real datasets are the key to creating better tools. We currently have one real dataset that has provided several intrusions and will continue to look for more in the future.

Acknowledgements

We wish to acknowledge the support of The Department of Homeland Security, CA Labs, and NSERC in this research initiative.

References

[1] K. Abdullah, C. Lee, G. Conti, J. A. Copeland, and J. Stasko. Ids rainstorm: Visualizing ids alarms. In *VIZSEC '05: Proceedings of the IEEE Workshops on Visualization for Computer Security*, page 1, Washington, DC, USA, 2005. IEEE Computer Society.

[2] R. Ball, G. A. Fink, and C. North. Home-centric visualization of network traffic for security administration. In *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 55–64, New York, NY, USA, 2004. ACM.

[3] G. A. Fink, P. Muessig, and C. North. Visual correlation of host processes and network traffic. In *VIZSEC '05: Proceedings of the IEEE Workshops on Visualization for Computer Security*, page 2, Washington, DC, USA, 2005. IEEE Computer Society.

[4] G. W. Furnas. Generalized fisheye views. *SIGCHI Bull.*, 17(4):16–23, 1986.

[5] J. R. Goodall, W. G. Lutters, P. Rheingans, and A. Komlodi. Focusing on context in network traffic analysis. *IEEE Comput. Graph. Appl.*, 26(2):72–80, 2006.

[6] Y. Hideshima and H. Koike. Starmine: a visualization system for cyber attacks. In K. Misue, K. Sugiyama, and J. Tanaka, editors, *APVIS*, volume 60 of *CRPIT*, pages 131–138. Australian Computer Society, 2006.

[7] D. Holten. Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data. *IEEE Transactions on Visualization and Computer Graphics*, 12(5):741–748, 2006.

[8] A. Komlodi, P. Rheingans, U. Ayachit, J. R. Goodall, and A. Joshi. A user-centered look at glyph-based security visualization. In *VIZSEC '05: Proceedings of the IEEE Workshops on Visualization for Computer Security*, page 3, Washington, DC, USA, 2005. IEEE Computer Society.

[9] K. Lakkaraju, W. Yurcik, and A. J. Lee. Nvisionip: netflow visualizations of system state for security situational awareness. In *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 65–72, New York, NY, USA, 2004. ACM.

[10] S. Lau. The spinning cube of potential doom. *Commun. ACM*, 47(6):25–26, 2004.

[11] Y. K. Leung and M. D. Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM Trans. Comput.-Hum. Interact.*, 1(2):126–160, 1994.

[12] F. Mansmann, D. A. Keim, S. C. North, B. Rexroad, and D. Sheleheda. Visual analysis of network traffic for resource planning, interactive monitoring, and interpretation of security threats. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1105–1112, 2007.

[13] J. McHugh, C. Gates, and D. Becknel. Situational awareness and network traffic analysis. In *Proceedings of the Gdansk NATO Workshop on Cyberspace Security and Defence: Research Issues*, volume 196 of *NATO Science Series II. Mathematics, Physics, and Chemistry*, pages 209 – 228, Gdansk, Poland, September 2004. Springer.

[14] J. McPherson, K.-L. Ma, P. Krystosk, T. Bartoletti, and M. Christensen. Portvis: a tool for port-based detection of security events. In *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 73–81, New York, NY, USA, 2004. ACM.

[15] J. Oberheide, M. Goff, and M. Karir. Flamingo: Visualizing internet traffic. pages 150–161, 2006.

[16] R. A. Rensink, J. K. O'Regan, and J. J. Clark. To see or not to see: The need for attention to perceive changes in scenes. *Psychological Science*, 8:368–373, 1997.

[17] T. Taylor, S. Brooks, and J. McHugh. Netbytes viewer: An entity-based netflow visualization utility for identifying intrusive behavior. In *VizSEC 2007: Proceedings of the 2007 workshop on visualization for computer security*, pages 101–114, Berlin, Germany, 2008. Springer-Verlag.

[18] C. Ware. *Information visualization: perception for design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.

[19] X. Yin, W. Yurcik, M. Treaster, Y. Li, and K. Lakkaraju. Visflowconnect: netflow visualizations of link relationships for security situational awareness. In *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 26–34, New York, NY, USA, 2004. ACM.