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
glanfield@cs.dal.ca

Carrie Gates
CA Labs
New York, NY, USA
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 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 challenge to the security analyst due to its volume. We report preliminary results on the development of a suite of visualization tools that are intended to complement 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[1] *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

The operation of even a modest network generates an immense amount of data. A security analyst must examine vast amounts of network data, host and IDS logs and information from other sources in order to understand network activity. Buried in this mass of data are the signs of intrusive behavior. The tasks associated with network analysis are time consuming and cumbersome. It is difficult to find patterns in the data and trends over time.

Recent research (see section 5 *infra*) has focused on using visualization techniques to help analysts gain a mental image of network behavior. Visualization is powerful because it allows us to see a significant amount of data at once and utilize our cognitive and preattentive processing abilities to find patterns much more quickly than sifting through packets or flow records. The real challenge for visualization is creating images that are more than just pretty pictures. To be useful, visualizations must provide insight into the underlying activity. It is possible to show too much information in a picture, thus confusing rather than helping the user, or too little, depriving the user of key facts.

Some years ago, we developed a number of static visualization approaches that provided insight into both network wide [6] and host [17, 18] behaviors. The work reported here extends these preliminary ideas to an interactive tool that is integrated with the SiLK NetFlow tools to allow the analyst a rich and flexible visualization capability that complements the textual features of the usual SiLK analysis. Our tool, FloVis, is a suite of interactive visualizations designed to show various aspects of network data flow. Each tool in the suite complements the others, visualizing different aspects of host and network behavior. We believe that the ability to obtain different views of the same data will reveal new patterns and provide the analyst with better insights.

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 approach the

[1] The tools are designed to take advantage of color for encoding information. The monochrome figures used in the printed version have been adjusted in an attempt to convey as much of the coded information as possible, but are far from ideal.

visual representation of flow data. This is followed by the introduction of our three key visual paradigms in Section 3. The representations are 1) time–series behavior for a group of hosts as color coded blocks representing categorical aspects of hourly behavior; 2) bundles of flows from related sources to related destinations; and 3) detailed, pseudo three dimensional displays of per port volume activity over time. Using data available from a local network, Section 4 describes several analysis scenarios in which the ability to visualize connection behavior leads to the discovery of compromised hosts. Section 5 gives an overview of related work in visual analysis for security. The paper ends with our plans for future work in Section 6 and some concluding comments in Section 7.

2. Approach

FloVis works with the SiLK Toolkit[2] for network data analysis. As collected by SiLK, NetFlow data is best suited for forensic analysis and retrospective studies since the data can appear in the repository several hours after the start of a flow. As a result, we assume that the analyst selects data to be examined visually based on the results of other analyses, IDS reports, and the like and is using FloVis primarily for the purpose understanding network behaviors already deemed to be of interest. The primary users of the SiLK suite are US CERT and DoD analysts who monitor the borders of very large networks with several million internal hosts. At this scale, we feel that visual "fishing" or unguided exploration is probably not approach to starting an analysis. Our goal is to provide the analysts with ways to view network behaviors that would be difficult to understand using report generating tools, or conventional IDSs. FloVis, like SiLK, is not intended as an IDS *per se*, but rather as a framework for helping the analyst understand behaviors that may be malicious or benign.

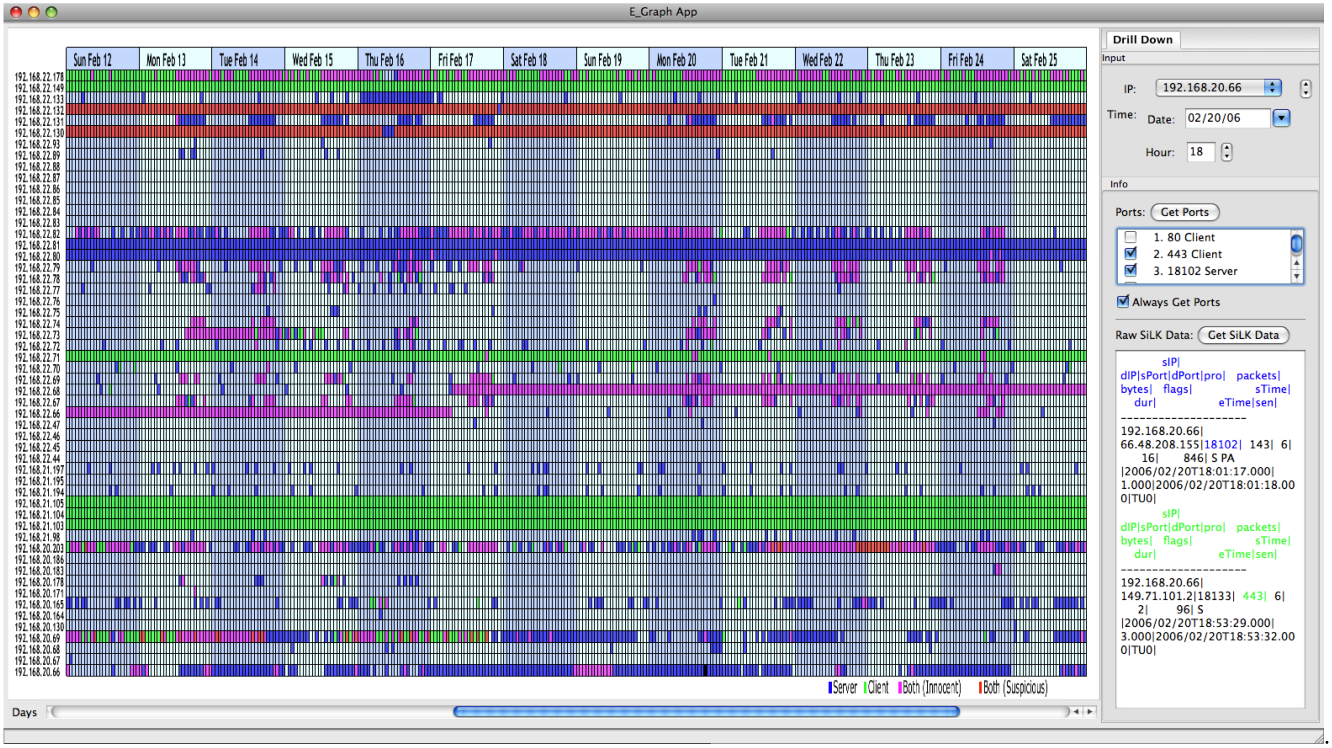
The SiLK tools are used to filter the raw flow data and to aggregate the selected NetFlow data into counted sets or bags. SiLK bags [16] contain counts of flows, packets, or bytes and can be indexed by IP addresses, ports, protocols and other scalar fields contained in flow data. They can also be indexed by composite values composed of up to 32 bits taken from any two scalar fields of the flow records. These can be used to indicate connections from 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 a database (MySQL at present) 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 from other sources to be treated in similar ways.

FloVis is a client based application designed to run on a workstation. It is built in OpenGL and utilizes a platform independent windowing system which allows it to be built on most operating systems although the initial implementation runs on OS X. FloVis currently contains three different types of visualizations: Flow Bundle Diagrams, the NetBytes Entity Viewer and an Activity Plot. 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 different views: an interesting region of one visualization can 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. In the future, the scripts will be parameterized and can be saved in a library for subsequent execution or modification as circumstances dictate.

3. Sample Visualizations

At the present time, FloVis incorporates three primary visualization modes. The first of these, an activity plot, captures some aspect of the temporal behavior of an arbitrary group of individual hosts as a time series of color coded blocks, one series per host. The second, the bundle diagram, captures interactions between hosts or subnets, allowing the analyst to see related flows grouped into bundles. The third, the NetBytes viewer, allows detailed inspection of the behavior of an individual host over time, making it easy to detect and understand changes in behavior that manifest as unusual port usage or traffic volume.

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



IP: 192.168.20.66

Time: Date: 02/20/06

Hour: 18

Ports:

- □ 1. 80 Client
- ☑ 2. 443 Client
- ☑ 3. 18102 Server

☑ Always Get Ports

Raw SiLK Data:

```
           sIP|
dIP|sPort|dPort|pro|   packets|
bytes|   flags|              sTime|
   dur|              eTime|sen|
---------------------
192.168.20.66|
66.48.208.155|18102|  143|  6|
     16|       846| S PA
|2006/02/20T18:01:17.000|
1.000|2006/02/20T18:01:18.00
0|TU0|
           sIP|
dIP|sPort|dPort|pro|   packets|
bytes|   flags|              sTime|
   dur|              eTime|sen|
---------------------
192.168.20.66|
149.71.101.2|18133|  443|  6|
    2|       96| S
|2006/02/20T18:53:29.000|
3.000|2006/02/20T18:53:32.00
0|TU0|
```

Figure 1. Activity Plot Application

3.1. Activity Plots

The Activity Plot is an interactive visualization tool for displaying host activity as a function of time using a small selection of colors to capture a limited number of activity categories. The activity plot is based in part on the existence plot introduced by Phil Groce and Jeff Janies at FloCon 2008 [8, slides 9 and 10]. An existence plot is a static rendering of the network traffic in which a dash is used to represent the activity of an entity per unit of time. In the activity plot, the activities of individual hosts are plotted against time in a simple two dimensional grid. For the time periods in which activity occurs, for a given host, the corresponding square is filled with a color that indicates the nature of the activity. The absence of activity is represented by an empty square. The nature of the represented activity is arbitrary. It can be host relative activity allowing the temporal variability of each host to be placed in a historical perspective for that host. It can be time relative activity in which the activity of each host for a given interval is compared to the activities of its peers. As in the present case, the colors can encode more complex behaviors, *e.g.*, whether a host exhibits client or server behavior or both. Figure 1 shows the activity plot for the active hosts in small network.

The current application displays fourteen days of TCP host activity in the main window. This allows the user to view an entire weeks worth of data with a few days from the previous and subsequent weeks for visible context. Each day is broken into 24 data blocks, with each block representing one hour of traffic per host. Colored background highlighting in used to visually group the hours in a day making it easier to distinguish between hours in different days. A scroll bar under the data view gives the user the ability to scroll through the data by days. The data view is interactive and gives the user the ability to select an individual data block on the display. Selecting a data point will cause the block in the data display to turn black and the control panel, on the right, to identify the current selection.

Color is used to depict the host's role based on the server-client paradigm. Blue indicates that the host acted as a server, *i.e.*, it responded to service requests but did not make them during the time period. Green means the host was a client, making service requests, but not responding to requests for service. Magenta indicates a host that acted as both a server and a client but that the service port set was disjoint from the client port set. Red indicates the host acted as both a server and a client on the same port(s). Hosts using the same port in both client and server roles may warrant investigation by the analyst if this behavior is uncommon. In addition, hosts such as workstations offering unexpected services, are suspicious. A number of services are bidirectional, SMTP for example, while others, such as DNS, often use the same port for both source and destination. We will take this into account as the tool evolves.

To the right of the main window is a control panel. At the top of the panel there is an input area for selecting a data point in the currently loaded data set by entering the IP address, date, and time of interest. The selected data point is displayed as black if it is visible in the main window. Data points that are not currently visible but are in the data set may also be selected. Visible points can also be selected using the mouse.

Under the input area on the control panel is an area for displaying information about the selected data point. For data points with activity, a list of the ports contributing to the activity can be generated as a check box list in the window labeled "Ports". By default, the port list is automatically generated for the selected data point. The user may disable this feature by unchecking the "Always Get Ports" check box and manually retrieving port data of interest using the "Get Ports" button. The role each port takes on the host is displayed. Individual or multiple ports can be selected from the port list for further drill down to the SiLK data.

Further drill down into the underlying SiLK data that generated the data point is available in the area below the port list. SiLK tools are run in a separate thread using the time of the selected data point and the selected ports as filter arguments. The data retrieved from the SiLK data repository is returned in the window provided. If a server port(s) is (are) chosen only server SiLK data is returned. Similarly, if a client port(s) is(are) chosen only client SiLK data is returned. If a port is selected which acts as both a server and a client then both server and client SiLK data is returned. If no ports are selected then the application tries to retrieve the SiLK data for both server and client data. If there is no data for one of these classes then an empty list is returned. Server data has a blue header and server ports and is always returned first. Client data is returned next with a green header and client ports. Color coding the ports in the output gives the user a visual clue that is consistent with the main display.

The drill down window gives the investigator details of host and port behavior at a particular point in time. In Figure 1, the currently selected data point is at the bottom of the main data display for IP address 192.168.20.66 at the 18:00 hour on 02/20/06. Two ports on the port list have been checked and the SiLK data has been retrieved for these ports.

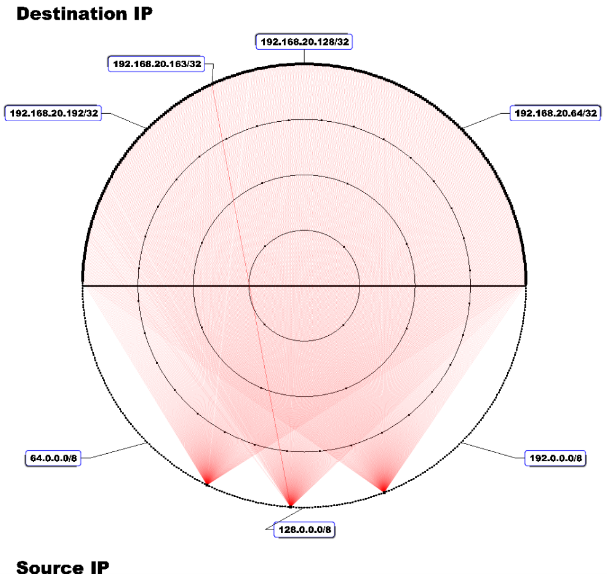
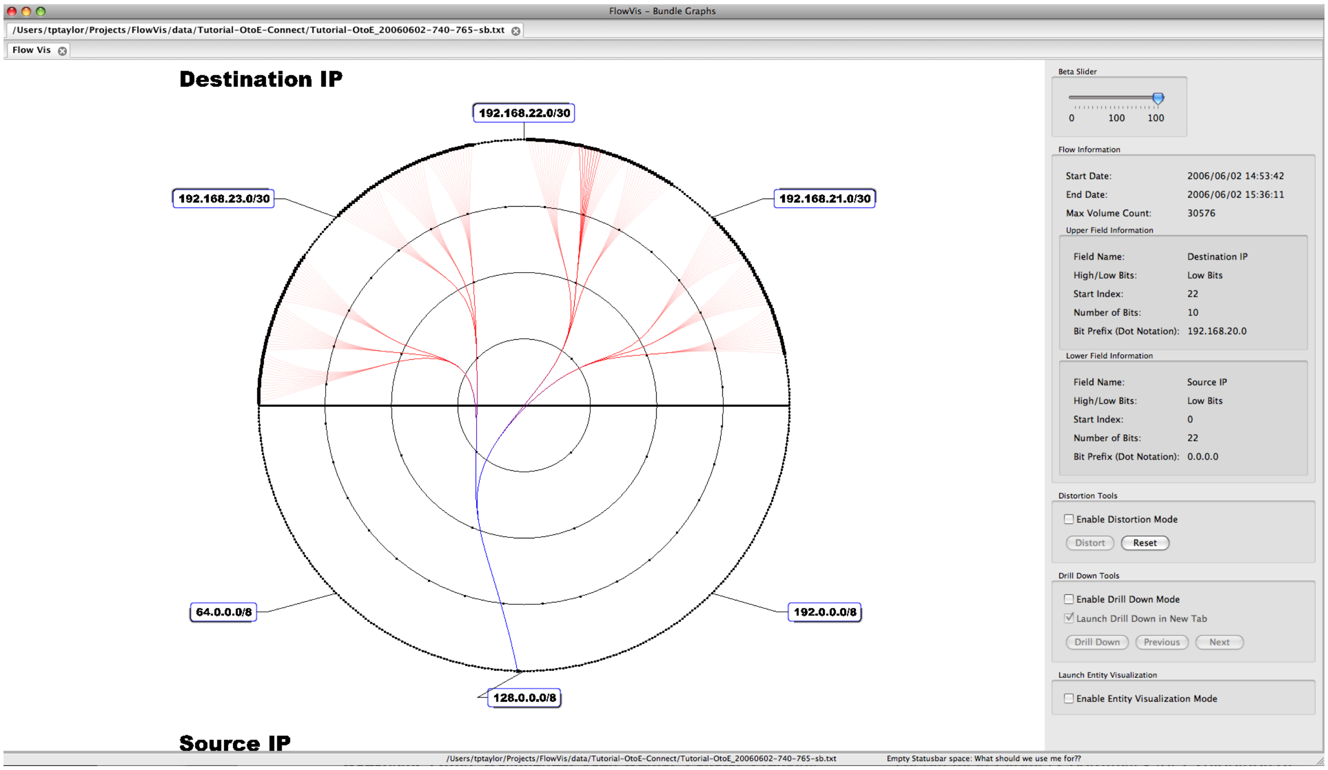


Figure 3. Loosening the bundle for a more detailed view.

3.2. Bundle Diagrams

The Bundle Diagram seen in Figure 2 (on the next page) displays connections between entities (*i.e.*, hosts or subnetworks) on a network. The disadvantage of many applications that visualize network connections is that they exhibit significant visual occlusion when showing large datasets. The Bundle Diagram [10] is based on a visualization designed to represent calling relationships among subroutines in complex software systems. It attempts to mitigate visual occlusion by bundling connections together and utilizing node aggregation. Figure 2 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. Color 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 that is traversing it (for a specific time period).

By using a slider control the user can choose to loosen the bundles, as shown in Figure 3 (above). Bundle loosening straightens the bundles, essentially separating them so the user can get a better look at individual connections. Note that this potentially increases occlusion, as described above.



Beta Slider

0 100 100

Flow Information

Start Date: 2006/06/02 14:53:42
End Date: 2006/06/02 15:36:11
Max Volume Count: 30576

Upper Field Information

Field Name: Destination IP
High/Low Bits: Low Bits
Start Index: 22
Number of Bits: 10
Bit Prefix (Dot Notation): 192.168.20.0

Lower Field Information

Field Name: Source IP
High/Low Bits: Low Bits
Start Index: 0
Number of Bits: 22
Bit Prefix (Dot Notation): 0.0.0.0

Distortion Tools

☐ Enable Distortion Mode

Distort Reset

Drill Down Tools

☐ Enable Drill Down Mode
☑ Launch Drill Down in New Tab

Drill Down Previous Next

Launch Entity Visualization

☐ Enable Entity Visualization Mode

/Users/tptaylor/Projects/FlowVis/data/Tutorial-OtoE-Connect/Tutorial-OtoE_20060602-740-765-sb.txt Empty Statusbar space: What should we use me for??

Figure 2. FloVis Bundle Diagram which shows data flows between IP entities.

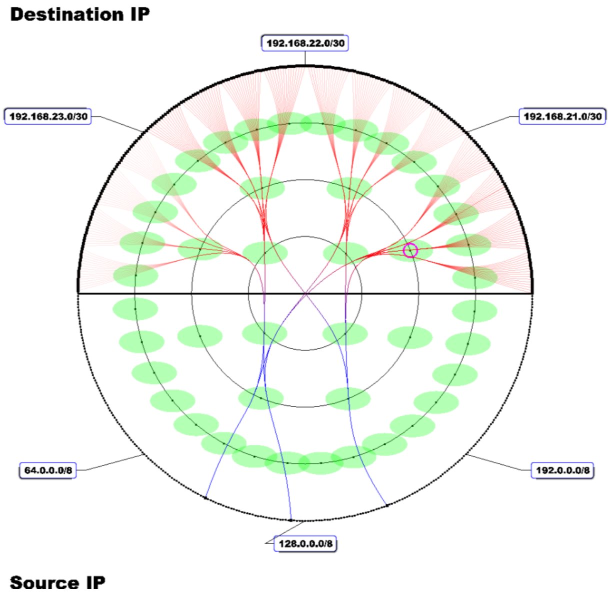


Figure 4. Bundle drill down mechanism.

The inner rings of the bundle diagram create a radial tree layout for the underlying data source, which facilitates the ability to drill down within the data. The bundle diagram supports a drill down mode as shown in Figure 4. Drill down activates a set of transparent green ovals on top 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 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 the Use Case in section 4.3 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 connection files while keeping the drilled down visualizations organized. Tabbing also allows two diagrams to be compared on the same screen for further analysis.

Another interactive feature of the Bundle Diagram is the ability to linearly distort points on the circle as shown in Figure 5. Distortion [14] is a well known presentation tech-

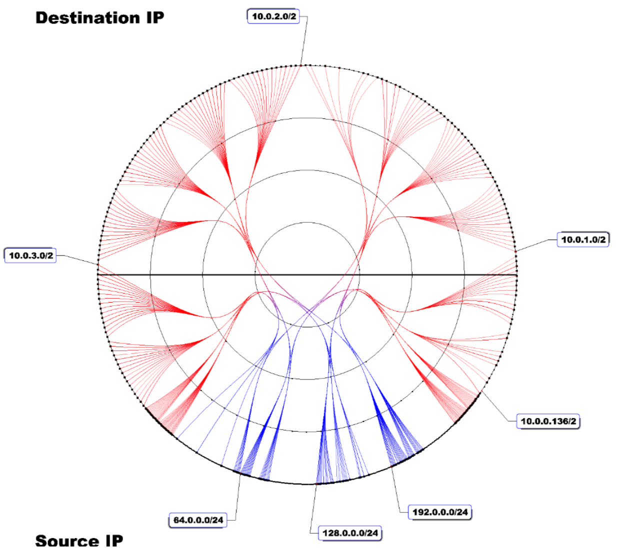


Figure 5. Linear distortion puts focus on key points on the circle.

nique that allows users to focus on key points on a visualization that might show interesting behavior while pushing aside points that are not that interesting.

Labeling is an important aspect of a useful visualization. Too many labels create occlusion issues, while too few might not provide sufficient information to the user. Our approach is to label a set of strategic points around the circle, as IP addresses are organized in order and so the user can infer the pattern. 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 numerous labels in a particular region, the visualization will push the other labels aside to create free space for the new label.

The bundle diagram visualization takes a daily or hourly connection bag[3] as its input. Connection bags contain information about the host to host connections occurring during a particular day or hour and can contain byte, flow or packet volume information for each connection.

The bundle diagram demonstrates which entities on the network are communicating, thus providing a representation of data flow. This allows us to see communication behaviors such as scans (easily detectable by other means) or interactions between internal hosts and scanners (usually undesirable). Using a time series of flow bundles, we can see changes in connection patterns or the appearance of new hosts that may warrant further investigation.

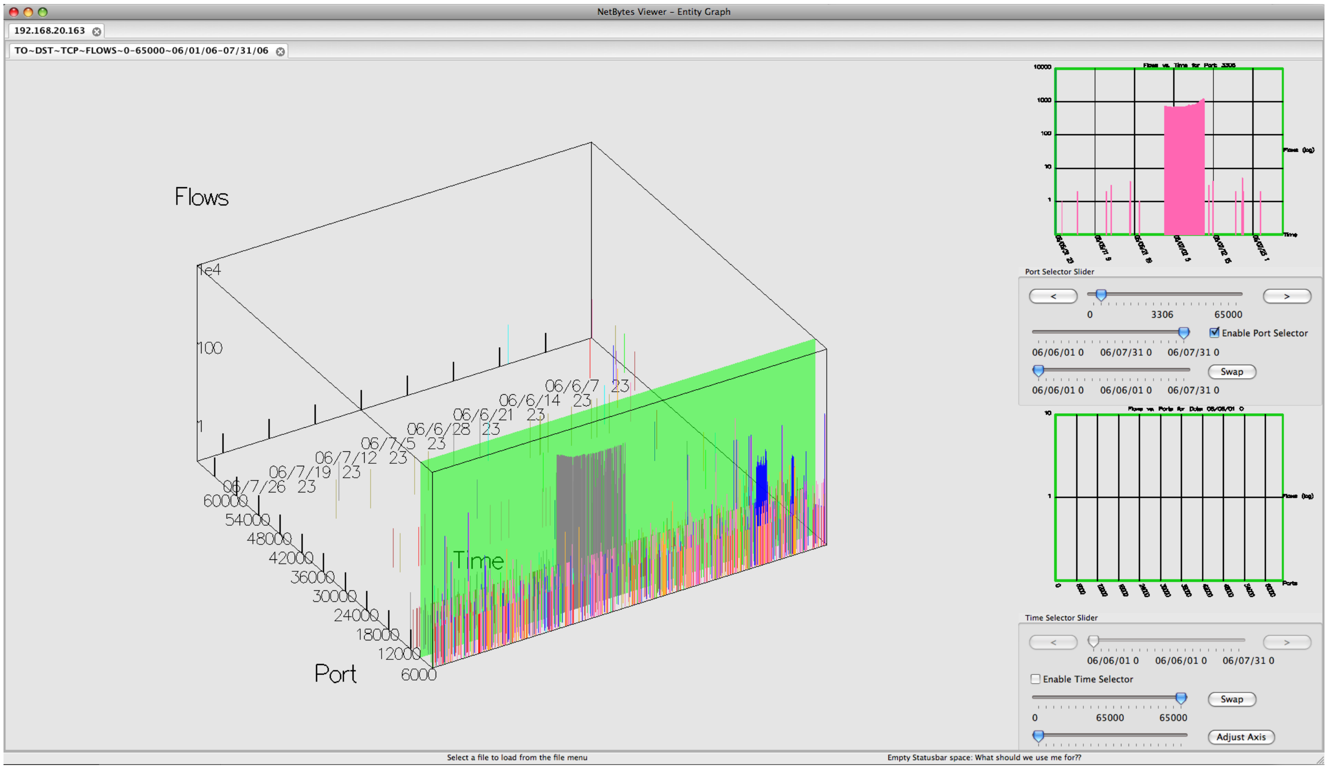
3.3. NetBytes Viewer

The NetBytes Viewer [23] complements the Bundle Diagram by focusing on the flow of data into and out of a single entity (rather than focusing on the interactions between entities), visualizing flow volumes related to that entity over time. It utilizes a 3D impulse graph with time, port (or protocol), and volume dimensions, as shown in Figure 6. This creates a "picket fence" effect in which each line of color shows the NetFlow volumes for a specific port (or protocol) on an hourly basis over an extended time period (weeks or months). This provides historical perspective for port (protocol) traffic without having to use animation. Animation relies on the user's short term memory to find temporal patterns in data and the user may forget or miss content during playback. Animation can also cause *change blindness* [22], suppressing the user's ability to see small changes.

The NetBytes Viewer can be launched either by clicking on an IP address in the Bundle Diagram or by providing host information to a configuration dialog. The user has the choice of displaying traffic volume going to or coming from a host, can choose source or destination ports, can specify a date range, and can select TCP or UDP data. The NetBytes Viewer is built on top of a MySQL database that is pre–loaded with port and protocol bags for individual hosts or subnets. This pre–computation would be done routinely for hosts on a "watch list" in a production system.

Three dimensional visualizations tend to suffer from two problems. The first is data occlusion, where some data is placed behind other data and is thus out of the user's view. The second is the loss of head parallax and depth perception [24] that accompanies 2D representations of 3D information. These leave the user unable to determine the exact location of a data point on the graph. The NetBytes Viewer addresses these problems through interactivity, allowing the user to rotate the 3D impulse graph using the mouse. This enables users to look at the data from all angles reducing occlusion. Looking down on the visualization from above can create a compelling image of port traffic patterns. NetBytes has a selection mode which allows the user to highlight a point on the volume versus time axis, or the volume versus ports (protocol) axis, and view the selected points as a pair of 2D graphs on the right hand side panel of the application as seen in Figure 6. In this figure, NetBytes is highlighting a large block of traffic on port 3306. This data is shown in the image on the upper right hand corner of the viewer. A similar highlight is available on the volume versus 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) that snap the highlight to the next port (or time period) with data, making it easier for the user to locate data precisely.

[3] We have extended the SiLk tools to allow bags indexed by the concatenation of subfields of any pair of normal indexing fields so long as the total length of the subfields is 32 bits or less. This allows us to display, for example, connections from all /24 prefixes to the individual hosts of a single /24.



Select a file to load from the file menu

Empty Statusbar space: What should we use me for??

Figure 6. NetBytes Viewer selection mode. The transparent rectangle highlights the volume versus time axis for a specific port.

The 2D images are useful, but they can be too small for detailed analysis. A swapping feature allows the user to swap one of the 2D views with the main 3D view as shown in Figure 7. All of the interactive features remain available.

NetBytes also has 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. Figure 7 shows sliders (one for the upper boundary and one for the lower boundary) being used 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. Tabs were used so the user can go back through a hierarchy of zoomed images and set the boundaries on another portion of the graph. Hierarchical tabbing is again utilized so that data for multiple hosts can be loaded in the tool at one time.

NetBytes provides a historical overview of the traffic into and out of an entity, enabling a user to recognize traffic patterns. For instance, if the user is looking at the graph of an email server, it will show traffic on the main ports that are used for email services, as well as on DNS ports for address lookup. Traffic seen on atypical ports, or traffic patterns that change over time, could be an indication of compromise or malicious behavior.

4. Use Cases

We illustrate the use of the FloVis system with three use cases, two involving the activity plot viewer and a more complex case that uses both the bundle diagrams and the NetBytes viewer. We have been collecting SiLK NetFlow on the network of a local enterprise since February 2006 and have a fairly complete data set up to March 2007[4], which we use for these examples. The first two cases examine the behaviors of individual hosts in the context of their historical behavior while the last one traces an intrusion resulting from interaction with a scanner.

4.1. IP Swapping?

In Figure 8, hosts 192.168.22.66 and 192.168.22.68 can be seen to have significant changes in their time

[4] Addresses have been anonomyzed to protect the identity of both the network and the external hosts. The monitored network has been translated into private space and routable outside addresses have had their /16 prefixes permuted. This preserves scans. In mid March of 2007, the organization adopted NAT for most of their hosts. This, combined with frequent collection equipment failures, limits the utility of data collected after that time.

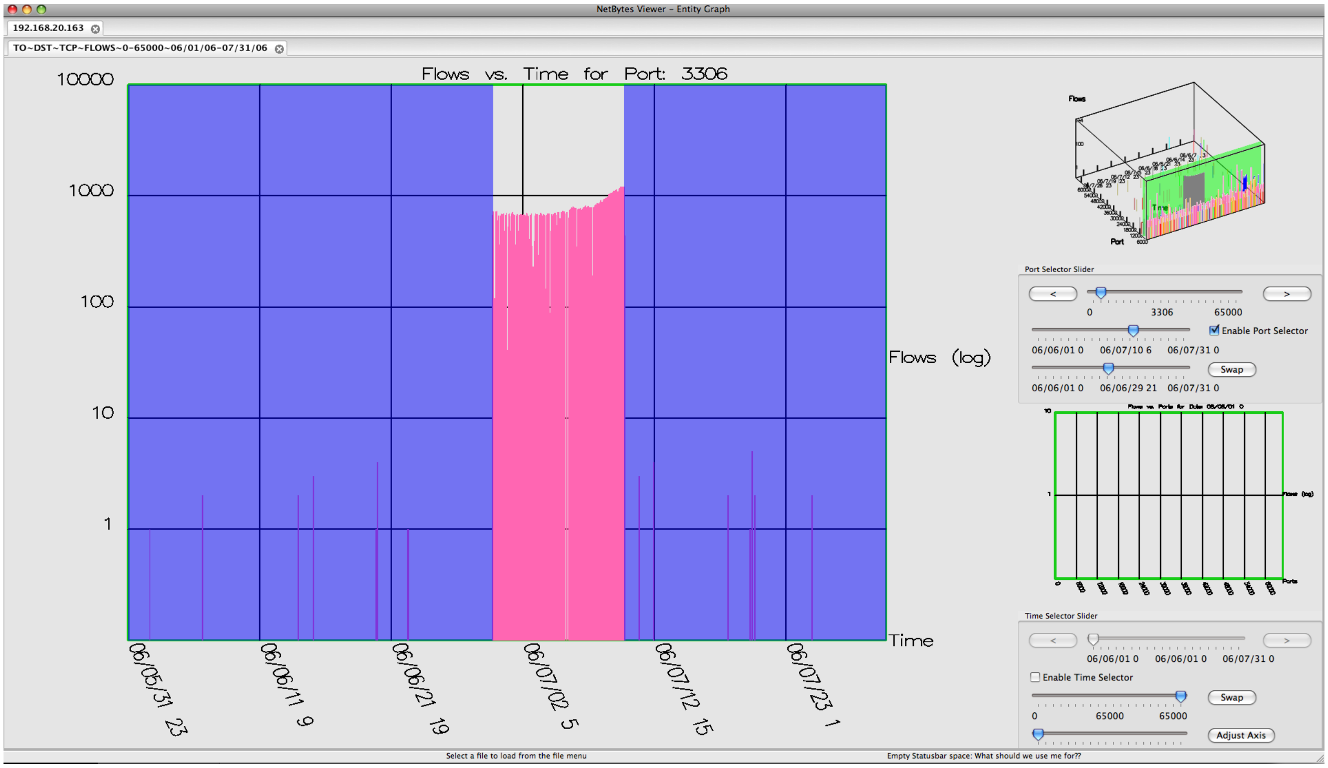


Figure 7. Highlight the upper and lower boundaries of the time axis to zoom in on the data.

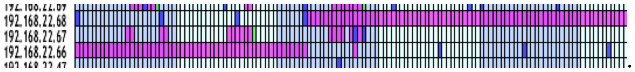


Figure 8. DHCP IP exchange.

based activity patterns with respect to their prior activity. 192.168.22.66 has nearly continuous server/client activity that stops abruptly. 192.168.22.68's activity pattern is the opposite, going from intermittent server activity to continuous server/client activity. According to the activity display, the two hosts appear to have "swapped" roles. Further investigation into the traffic generated by these hosts supports the conjecture. The network is a DHCP environment where the hosts usually maintain a constant IP address, but are not always guaranteed the same address when leases expire. Since the services are reached via DNS lookups, the swap causes, at worst, transient outages in access.

The paired behavior is quite striking and shows the effect of address reallocation. In an environment, such as a wireless network of convenience, a much higher degree of IP reuse would be expected and capture of platform specific attributes might be useful for creating IP address independent platform histories. We have used MAC addresses for this purpose with one dataset.

4.2. A Malicious Outbreak

On 28 February 2006 at seventeen hundred hours, the host at 192.168.22.82 started using the same ports as client and server and its square in the graph turned red. Up until this point in time, the activity had been server (blue), both server and client on different ports (magenta), and client (green) at various times. Further examination revealed that the host had activity on over nine thousand ports during the hour. A quick glance through the port list showed that the majority of the activity appeared to be client traffic (SYN packets) directed at another internal host and that the ports that were contacted were consecutively numbered. Further investigation was done with the SiLK tools. The host contacted fourteen external and one internal hosts as a client. The external hosts were contacted on port 80 and the flows are typical web connections. The connections to the internal host involved some 9000 ports. This appears to be scanning activity, but with some strange aspects. It begins with a pair of fast TCP SYN scans that concentrate on ports in the range of 1 to several thousand, *omitting the most common low numbered service ports*. Each of these scans uses a single reply port (different for each scan). This is followed by a much slower SYN scan targeting ports between 1 and 65535 in no particular order with a normal progression of reply ports, starting with 1025, going up to 5000 and repeating.

Matching this traffic shows that the scanned host responds with a ACK/RST pair on most of the ports[5]. The scanning activity continued for several days until the machine was removed from the network and its software reinstalled. It is unclear if the ports being used as both servers and clients are related to this scanning activity but their appearance as the scanning activity started is highly suspicious. The target of the scans is a machine known to have been compromised prior to the start of our collection.

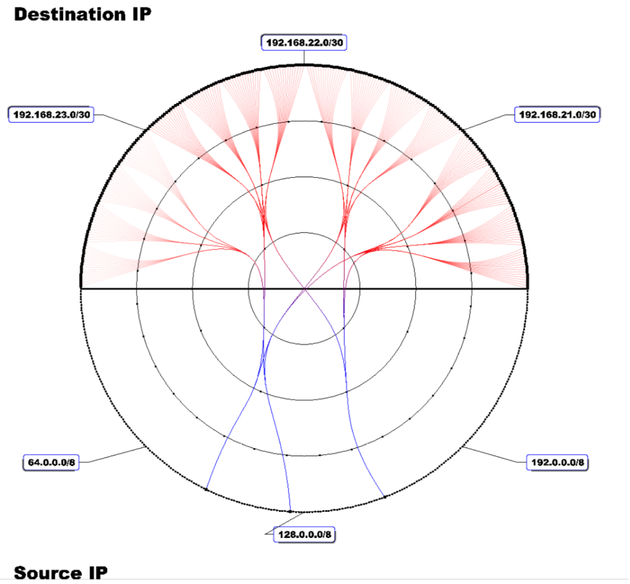


Figure 9. Bundle Diagram of 3 scanners with one intrusion. June 29th, 2006.

4.3. Analyzing an Intrusion

In this section we describe the use of FloVis to analyze an intrusion. For this use case we focused on the scanners. We note that identifying high volume scanners is trivial and essentially uninteresting, but once identified, their interactions with the monitored network can prove instructive. Using the SiLK tools, enhanced by a locally developed Bloom filter plugin, we counted the number of unique internal destinations targeted by each external host. Since at most about 100 of the 1016 possibly occupied addresses[6] in the network are monitored, we are safe in assuming that anyone who targets more than a few hundred addresses is scanning. The number of contacted addresses exhibits well defined peaks at 254, 508, 762, and 1016 hosts [6, Figure 11] and we arbitrarily selected scanners targeting between 740 and 765 internal addresses during June and July of 2008 for the purpose of this analysis. The set of 14 scanning sources thus identified came from 14 distinct /16s. Looking at traffic from the monitored network to the scanners, we found that significant interactions occur. We generated a set of daily 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 2 on page 5. This Diagram shows all the scanning activity for June 2nd, 2006. The scan is characterized by the large degree of fanout from the single scanning source. Most of the connections are faint, indicating a single connection attempt, however one group, just to the right of the 12 o'clock position is more opaque, indicating multiple connections or connection attempts. A high level of interaction may be a visual signature of a successful intrusion or it may simply indicate repeated connection attempts. Future versions of the analysis that drives the visualization will annotate the data to clearly identify suspicious interactions, but for now, we examine them individually.

The data for June 29th, 2006, shows a suspicious interaction, visualized in Figure 9. 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. Using the drill down mode described previously on page 5 and shown in Figure 5 we discover the details that are seen in Figure 10. 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 3 on page 4 we see exactly where the connection came from, in this case 123.99.63.81. We used the NetBytes Viewer to take a look at the port activity for host 192.168.20.163. Selecting TCP traffic to destination ports on this host for June and July 2006, we get the display shown in Figure 6 on page 7. We see in Figure 7 that there is a spike in data on port 3306 and that this activity started on June 29th (the same day of the data transfer from the scanner) and ended on July 10th. Looking at UDP data transfer to the host destination ports as seen in Figure 11, we see another interesting spike. This spike is on port 137 and occurs between June 29th and July 10th (the same duration as the TCP spike). Port 3306 is used for communication with MySQL databases and port 137 is used for NetBios communications on Microsoft hosts. Off site port 137 UDP transactions are suspicious in their own right. MySQL systems with weak passwords can be susceptible to a MySQL botnet attack where the bot tries to login to obtain root access to the server. If it succeeds, it installs malicious code on the server. There is also some indication that port 3306 TCP is used by the

[5] The identification of the host as showing same port client/server activity may result from the overlap in scanned and reply ports and our inability to definitively determine the origin of a session in this data set due to having time resolution of one second and our inability to separate the TCP flags for the first packet of a flow. We hope to deploy a collector in the near future that will resolve these problems.

[6] .0 and .255 in each of the four /24s that make up the /22 are not forwarded by the router and do not appear in the data.

eMule or eDonkey peer to peer applications. Port 137 may be a cover for transferring some sort of data between the scanner and the local host. Without the payload packet information it is difficult to say exactly what happened, but this machine exhibited behavior that should be investigated. Further investigation of activity on the host shows significant traffic on ports 80 and 443. This is consistent with a web server supported by a MySQL server to facilitate transactions for the site.

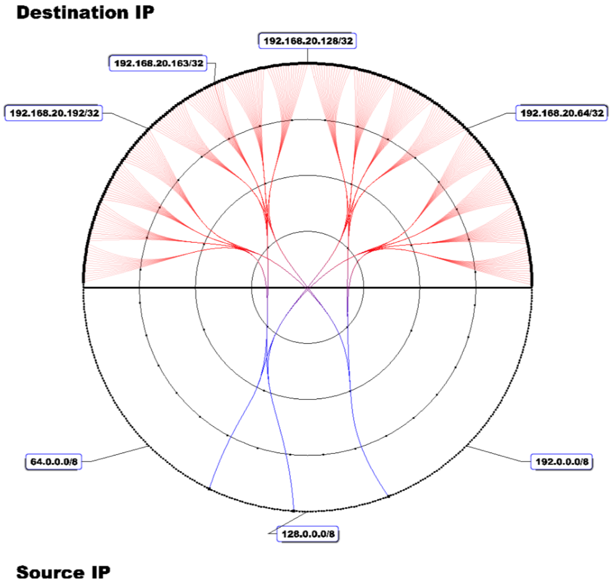


Figure 10. Results of the drill down shows host 192.168.20.163 with possible intrusion.

5. Related Research

There are numerous security visualizations that address various aspects of network data analysis [3, 9, 15, 1]. The ones discussed below are the most relevant to this project. Space limitations preclude including illustrations.

VisFlowConnect [25] and RUMINT[7] 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.

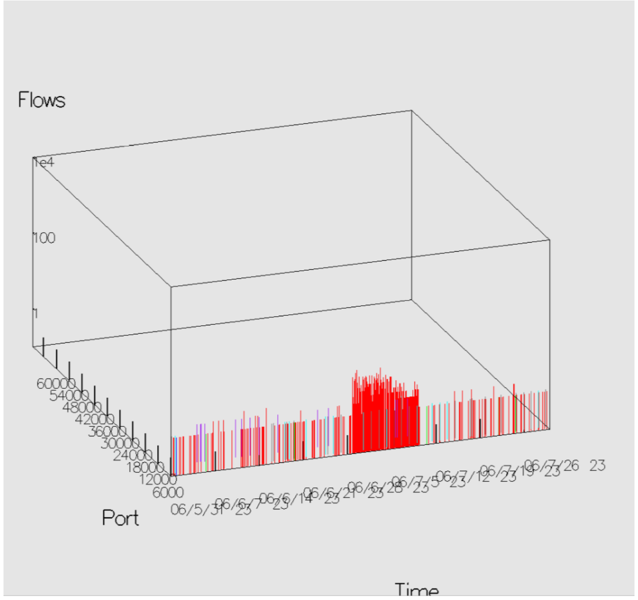


Figure 11. NetBytes Viewer for host 192.168.20.163 filtered by UDP (data to host with destination ports).

Flamingo [20] takes the parallel axes concept into a third dimension. It uses two quad-tree 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 quad-tree to another. This allows the application to visualize many more connections but creates the 3D issues of occlusion and depth perception described earlier. Flamingo can also visualize traffic volume per IP by using a bar graph within the quad-tree structure where bar height represent flow count. Individual port traffic is shown by rendering several quad-tree square layouts at different heights along the cube. Volume is again shown using bar height. Flamingo illustrates the need for multiple visualizations in network security analysis, but at times tries to display too much information on a single image.

Link-node graphs are another technique for showing data flow in a network. Hosts are represented by nodes on the graph while connections are represented by edges. These graphs also suffer from occlusion and become confusing as more nodes are added. VISUAL [2] is an example of a node/link graph that works well for small networks. It encodes host volume flows and port information inside each node. FloVis builds on this by providing the NetBytes Viewer to give detailed port information.

One of the better named visualizations, “The Spinning Cube of Potential Doom” [13] 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

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

Z axis 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, notably scans. It is a novel technique but suffers from depth perception issues and has minimal interactivity.

Scatterplot techniques have been used to visualize network traffic and volume. NVisionIP [12] visualizes an entire /16 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: traffic volume, number of flows or flows on a particular port. Animation is used to show traffic activity over 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 a user's short term memory find patterns in the data. The NetBytes Viewer overcomes this by showing time along a third dimension.

Portvis [19] is another application based 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 represents 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 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, allowing the user to see trends on ports more quickly without the need for port volume aggregation.

Komlodi *et al.* [11] describe an approach where a user can tailor a glyph-based visualization to their own needs by mapping data variables to visual glyph attributes (such as color, size, position). 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 [7] takes a focus + context [5] 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.

Recent work by Phan *et al.* [21] (Isis) is close, philosophically, to the FloVis approach in that it supports similar concepts for pivoting and drill down. We allow pivoting on IP addresses in both the activity viewer and in the bundle viewer with the possibility of a detailed examination of individual host behavior using the NetBytes viewer. As future work, we intend to provide drill down to the sources or destinations of a given impulse on the plot along with pivoting on any of these. Like Isis, we make use of a relational database, however the scale of the SiLK repository with which we work, precludes direct storage of flow data in the database.

Recently, Fischer *et al.* [4] describe a system using hierarchal edge bundles for visualizing network traffic. The system integrates IDS alerts (that FloVis does not currently assume are available) and seems directed towards visualizing previously known attack forms rather than for exploratory analysis. The home centric nature of the visualizations does not appear to scale well to systems involving many monitored subnets with thousands of hosts.

6. Future Work

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

The activity plot is a simple compact way of representing a global view of the network activity over a period of time. It lends itself to time series data that can be color coded into a small number of groups. The current implementation captures host activity as a result of TCP port activity. Another useful activity encoding would be the use of colors to represent quartile or quintile volume data and possibly other behaviors such as insignificant (RST replies) and significant (data exchange) interactions with scanners. We will look at developing a uniform mechanism for populating the database with the necessary time series information for a variety of displays.

Planned enhancements to the bundle diagrams include better encoding of volume information. We also plan to allow representation of bidirectional flows on a single bundle diagram. We are extending the SiLK bag structure to allow indexing by full connections rather than the partial address information now used. This is part of an effort to extend the set and bag paradigms to IPv6 addresses.

In the near future, the NetBytes viewer will also support bidirectional flows so that traffic *to* a given port will be represented by an upward impulse, and that *from* a given port by a downward impulse. This extension will improve our ability to understand host behaviors. In addition, the ability to generate SiLK scripts as part of the drill down will be unified and expanded and scripts deployed for updating the visualization database automatically as data is added to the archive.

We also intend to demonstrate the tool to the analyst community in order to obtain relevant feedback from our target users. We hope to expand this into a formal user study; however, to do so would require a significant number of professional security analysts who are willing to participate. Using other communities, such as undergraduate computer science students, is not likely to provide meaningful results. We note that performing a comparison with other visualization tools as part of such a user study would be ideal; however, it is outside the scope of the currently funded project.

Finally, we are seeking access to additional data sets for testing the visualization tools. Different networks might represent very different environments, and it is important to ensure that either the tools work across these different environments, or to identify those environments where the visualizations might not provide the best results. Given some of the legal and privacy issues involved in sharing data, obtaining such data set represents an ongoing challenge.

7. Conclusion

With the amount of data a security administrator must sift through on a daily basis, visualization is an essential analysis technique for network security. In this paper, we have presented a new suite of visualizations called FloVis, which is designed to show several views of network data supporting security 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 Activity Plot. These visualizations complement and interact with one another to create an application suite useful for detecting intrusions. In this paper we described each of our three visualization paradigms and provided three case studies that demonstrate the utility of these visualization approaches.

Acknowledgments and Disclaimer

This material is based upon work supported by the Department of Homeland Security under Contract No. N66001-08-C-2032. We also wish to acknowledge the support of Ron McLeod of TARA, CA Labs, and NSERC in this research initiative. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the Department of Homeland Security.

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*, Washington, DC, USA, 2005.

[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, Washington DC, USA, 2004.

[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.

[4] F. Fischer, F. Mansmann, D. A. Keim, S. Pietzko, and M. Waldvogel. Large-scale network monitoring for visual analysis of attacks. In J. R. Goodall, G. J. Conti, and K.-L. Ma, editors, *VizSEC*, volume 5210 of *Lecture Notes in Computer Science*, pages 111–118. Springer, 2008.

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

[6] C. Gates and J. McHugh. The contact surface: A technique for exploring internet scale emergent behaviors. In D. Zamboni, editor, *Proceedings of the Fifth Conference on the Detection of Intrusions and Malware & Vulnerability Assessment*, volume 5137 of *Lecture Notes in Computer Science*, pages 228–246, Paris, France, July 2008. Springer.

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

[8] P. Groce and J. Janies. Visualizations of flow and analytical results. Powerpoint presentation at FloCon, Jan 2008. A pdf of the slides is available from the FloCon 2008 web site, `http://www.cert.org/flocon/2008/proceedings.html`, as of 21 October 2008.

[9] 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.

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

[11] 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*, Washington, DC, USA, 2005.

[12] 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, Washington DC, USA, 2004.

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

[14] Y. K. Leung and M. D. Apperley. A review and taxonomy of distortion-oriented presentation techniques. *ACM Transactions on Computer-Human Interactions*, 1(2):126–160, 1994.

[15] 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.

[16] J. McHugh. Sets, bags, and rock and roll: Analyzing large data sets of network data. In P. Samarati, P. Y. A. Ryan, D. Gollmann, and R. Molva, editors, *Proceedings of the 9th European Symposium on Research in Computer Security*, volume 3193 of *Lecture Notes in Computer Science*, pages 407–422, Sophia Antipolis, France, September 2004. Springer.

[17] J. McHugh and C. Gates. Locality: A new paradigm for thinking about normal behavior and outsider threat. In *NSPW '03: Proceedings of the 2003 Workshop on New Security Paradigms*, pages 3–10, Ascona, Switzerland, August 2003.

[18] 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.

[19] 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, Washington DC, USA, 2004.

[20] J. Oberheide, M. Goff, and M. Karir. Flamingo: Visualizing internet traffic. In *Proceedings of the 10th IEEE/IFIP Network Operations and Management Symposium*, pages 150–161, 2006.

[21] D. Phan, J. Gerth, M. Lee, A. Paepcke, and T. Winograd. Visual analysis of network flow data with timelines and event plots. In J. R. Goodall, G. Conti, and K. L. Ma, editors, *VizSEC 2007: Proceedings of the Workshop on Visualization for Computer Security*, pages 85–99. Springer, 2008. 10.1007/978-3-540-78243-8_6.

[22] 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.

[23] 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, Sacramento, USA, 2008.

[24] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2000.

[25] 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, Washington DC, USA, 2004.