NetBytes Viewer: An Entity-Based NetFlow Visualization Utility for Identifying Intrusive Behavior

T. Taylor, S. Brooks, and J. McHugh

Abstract NetBytes Host Viewer is an interactive visualization tool designed to show the historical network flow data per port of an individual host machine or subnet on a network over time, using a 3D impulse graph plot. Such visualizations allow network administrators to quickly and effectively diagnose infected or malfunctioning computers by viewing data transmission patterns for each port on the entity. NetBytes has a set of interactive features which help to deal with the problems associated with displaying a 3D graph on a 2D screen. First, NetBytes offers a "selector" mode which allows the user to highlight specific ports (or times) on the graph using a slider and snap buttons. From the selector, the user can launch a set of 2D graphs (Bytes vs. Time and Bytes vs. Ports) to acquire more detailed information about the host with less clutter. Lastly, the user is able to rotate the 3D graph in any direction to mitigate occlusion. The long term objectives of this work include the integration of the NetBytes Viewer with complementary visualizations of the overall network. This application will integrate with a larger network analysis tool and be utilized as a drill-down mechanism.

1 Introduction

Over the years, the Internet has increasingly become host to Trojans, scanners and other viruses which can wreak havoc on private networks. These hostile programs have made it necessary for security administrators to watch over their networks using intrusion detection systems which trace large amounts of packet data. This data must be analyzed for signs of viruses. To help in this analysis, researchers have turned to visualization techniques. Visualizing network data allows network administrators to more quickly discover patterns of hostile activity and diagnose problem

T. Taylor, S. Brooks, and J. McHugh
Dalhousie University, Halifax NS, e-mail: teryl@cs.dal.ca, sbrooks@cs.dal.ca, mchugh@cs.dal.ca

computers. However, current techniques focus on visualizing network traffic as a whole and do not focus on visualizing historical data of individual hosts.

The NetBytes Viewer is an interactive visualization tool designed to show the activity of a network entity over time using a 3D impulse plot. The entity could be a single host (e.g. an email server) and the visualization shows the per port outbound NetFlow volumes for a certain time interval (e.g. hourly) over a specified time period (e.g. a week). The entity could alternatively be a subnet, showing per host behavior or a larger network aggregation showing per subnet behavior. Such a visualization allows network administrators to quickly and effectively diagnose infected or malfunctioning computers by viewing data transmission patterns for each port on the host. For example, an email server would typically have a significant traffic load on port 25 (STMP) and port 110 (POP). A visualization using NetBytes would show this as "picket fences" corresponding to the ports with the height of each picket (impulse) representing an hourly volume. NetBytes allows the user to monitor a range of ports for significant network traffic. Traffic on ports that should be unused may be a sign of a worm, virus, or other malcode. Unusually high volumes on supported services may also indicate a compromise. Using a historical database, NetBytes can help pinpoint the time at which a host was compromised. Tracing back through the original NetFlow data for this time allows the user to determine the source of the compromise. Working forward from that time can identify secondary infections.

This paper will discuss the design and implementation of the NetBytes Viewer and discuss how it can be an effective tool for analyzing network traffic. Furthermore, the document will outline some of the other network visualization tools currently under research as well as discussing the future plans for NetBytes.

2 Related Work

There exist several network traffic visualization tools that enable network administrators to monitor traffic flows on their internal networks. VISUAL (Ball et al., 2004) is a 2D visualization application which shows the traffic for a small to medium-sized network. All internal hosts are represented as cells in a square grid in the center of the screen. External hosts are represented as squares placed outside the internal grid. Square-size denotes the amount of IP activity of the host and lines in the square are used to denote port traffic. Line connections between internal and external hosts denote traffic flow while grid color denotes communication between computers internally. VISUAL allows interactive filtering of specific computers to reduce screen clutter and utilizes animation to show changes in network flow data over time. Detailed information of individual hosts is available in "text" form and includes host IP, IP addresses of all computers that are communicating, ports used and percentage of overall traffic. However, VISUAL is limited in its ability to show individual port traffic for a specific host over time.

VisFlowConnect (Yin et al., 2004) utilizes a set of parallel axes view on a 2D screen. Each point on an axis represents an IP address (of a machine or domain) and connections between points on parallel axes represent network connections and flow of data. Users can drill down in an overall global network view to see a subset of IP addresses in a specific domain or a view of the internal network. VisFlowConnect depicts time by using animation to show connections appearing and disappearing between computers. Furthermore, it has the ability to filter by port, protocol, transfer rate, and packet size. Lastly, VisFlowConnect does show individual host statistics in a text view. Stats include: total number of bytes into and out of the machine in the current time window, as well as, bytes transferred per machine.

NVisionIP (Lakkaraju et al., 2004) presents a visual representation of an entire class-B IP network on a single screen. The overview screen has horizontal and vertical axes in which 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. Once again animation is used to show traffic collected over a given time period. 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 with 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. NVisionIP also has a machine view which is launched by clicking on a single host. This launches a window which visualizes byte and flow count for all protocols and ports using 2D bar graphs. Although the drill down mechanism allows users to view data on individual hosts, it does not visualize patterns in the data over time.

The "Spinning Cube of Potential Doom" (Lau, 2004) is another network visualization tool which was built on top of the Bro intrusion detection system. It displays a 3D cube which a user can interactively spin. "Each axis represents a different component of a TCP connection: X is the local IP address space; Z is the global IP addresses space; and Y is the port numbers used in connections to locate services and coordinate communication" (Lau, 2004). TCP connections are displayed as individual dots with color used to distinguish a successful from an unsuccessful connection. Time is again displayed through the use of animation. This application has no drill down mechanisms for showing detailed data for individual hosts.

None of the above applications give detailed information on individual hosts. NetBytes complements these applications by showing detailed time and port information for individual host all on a single screen. It would therefore be an excellent addition to any of these applications.

Portall (Fink et al., 2005) is a network visualization tool which displays TCP connections between hosts and is able to correlate those connections with the processes that generate them. The main application window shows two parallel axes. One axis represents client hosts and processes of the machines that are monitored on the network while the other axis represents server hosts and processes. Client-server connections are displayed using straight lines from one axis to another while transmission volumes are visualized by tool tip popups. Time is represented using

animation with processes and connections appearing and disappearing. Portall is the first visualization tool to try to correlate traffic flow to the processes sending and receiving the traffic. It works well for very small networks but suffers from clutter and occlusion issues as the number of hosts and processes grow on the main display. Having said this, Portall and NetBytes could be good complements to one another as NetBytes can show traffic volumes and history for a host on a single screen while Portall could show from which processes that data is coming from.

Portvis (McPherson et al., 2004) is another visualization tool which incorporates three different displays to view TCP port traffic. The first visualization corresponds to a 2D grid where rows along the vertical axis represent units of time while each column along the horizontal axis represents a range of 2,048 ports. Color in each column is used to represent the level of activity on the ports during a particular time unit. A selector is used to select the unit of time to be displayed on the main visualization. The main visualization contains a 256×256 grid where each point represents one of the 65,536 ports. The location of the port on the grid is determined by breaking the port number into a two-byte (x,y) location. The grid can be magnified in specific areas to provide more detailed information about specific ports. Lastly, the tool contains a port display which displays information on five attributes of a selected port. The 3D display is a set of 5 bar graphs where time is represented on the horizontal axis and traffic volume on the vertical. This visualization tool shows many different types of data at the same time using several different 2D graphs. Portvis takes an interesting approach to visualizing port traffic; however, time-dependent patterns are defined very coarsely (2,048 port buckets) which could make detecting temporal patterns difficult. Furthermore, there may be issues of information overload especially 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 data trends on ports more quickly with no need for port volume aggregation.

IDS Rainstorm (Abdullah et al., 2005) was designed to visualize IDS alerts for the Stealthwatch anomaly-based IDS system. The main visualization divides the display into a set of columns that represent a contiguous set of IP addresses. The horizontal portion of each column represents a 24 h period of alerts. Colored dots in a row represent the total alarms for the IP addresses at a particular instance in time. The user can use a mouse to highlight an IP address range. Upon clicking on a selection, a secondary window is launched with an enlarged view. The selected IP addresses appear on a vertical axis on the right side of the window. Time appears on a horizontal axis and the alarms appear as large colored glyphs. This works shows how a drilldown mechanism can be used to view specific data in more detail. NetBytes Viewer utilizes drilldown to launch 2D views of multiset data to gain more insight into network traffic patterns.

The work of (Komlodi et al., 2004) is compelling as it places the user at the center of the design process itself, and offers a great deal of flexibility over the visualization. The user is able to visualize IDS alerts in both 2D and 3D windows and set the mappings between variables and visual attributes of glyphs such as position, size,

opacity and color. In addition, they derive a set of new guidelines for the design of information visualization tools for intrusion detection.

In "Situational Awareness and Network Traffic Analysis" (McHugh et al., 2004), the authors look at various techniques for analyzing and visualizing network traffic. Although the authors did not develop any interactive applications as part of the study, they do show a set of graphs which help to visualize network traffic at aggregate and individual host levels. One such graph is a 3D chart of an individual host which displays bytes transferred from a host per port per hour over a specified range of time. It allows the user to easily see patterns in port traffic which allow for quick diagnosis of anomalous behavior of the machine. NetBytes extends this 3D graph visualization by making it interactive so that patterns can be analyzed much more in-depth.

3 Technical Approach

The NetBytes Viewer was designed to visualize processed NetFlow data. Application volume data is obtained from multisets (bags) produced by the SiLK Analysis Tools from NetFlow data and can represent flow, packet, or byte counts.

NetFlow is an open protocol developed by Cisco for collecting IP traffic information. It "is an abstraction that provides a level of detail that is less than that from packet headers, but greater than session summaries. Flow records capture source and destination addresses, protocols, ports (for TCP and UDP), traffic volumes (packets and bytes), and start and end times" (McHugh et al., 2004). Such records can be generated using a NetFlow enabled Cisco router on a border network then collected and analyzed using a set of data analysis tools called SiLK. SiLK has the ability to perform many operations on the data including bagging (counting) flows, bytes or packets for specific unique key identifiers such as source ports, destination ports, source IP addresses, or destination IP addresses. This bagging technique was used to manipulate raw NetFlow records which were subsequently placed in a relational database and selected for visualization using relational queries allowing for multiscalar visualization over a wide range of entities.

The NetFlow data used in this project was collected from a private office network. The data shown in this paper is from a network email server with significant IP traffic on ports 25 (SMTP), 110 (POP), and 443 (https used for email web access).

3.1 NetBytes Viewer User Interface

NetBytes offers a comprehensive set of different views to make it possible for a network administrator to thoroughly analyze the traffic for an individual host (or aggregate subnet) on a network. It does this by showing the traffic activity per port on an hourly basis over a specified time period. This allows the administrator to

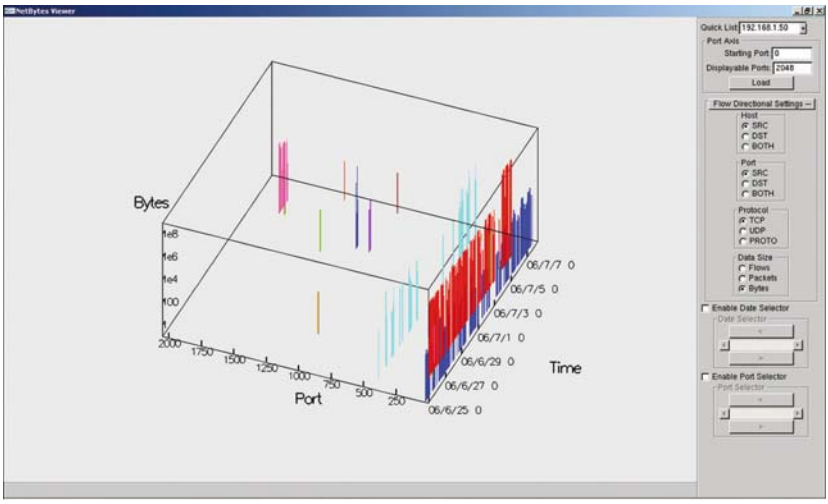

Fig. 1 NetBytes host overview representation of a 3D impulse graph (bytes vs. ports vs. time)

see patterns of anomalous traffic activity (e.g. traffic spikes or scanner activity) on suspicious ports. Furthermore, NetBytes enables the admin to pinpoint the time at which the anomaly occurred so that he/she may determine how the host became infected.

The initial overview screen (shown in Fig. 1) displays a global view of a single host's traffic structured as an orthogonal 3D impulse graph. Along the *Z* axis of the display is the port list for the machine (ranging from 0 to 2,048 in Fig. 1) while time (at an hourly granularity) is display on the *X* axis. The third dimension represents the magnitude of traffic (in flows, packets, or bytes) seen by the host (or subnet) in an hour.

Time was displayed as a third dimension to avoid the need for animation. Animation requires users to use short-term memory to remember trends and patterns in the data which could be missed or forgotten during playback. Furthermore, animation can cause issues of change blindness for the individual (Rensink et al., 1997). By contrast, with time on the third dimension, all the information is displayed on the screen, allowing the user to view and retrieve important information quickly. Color is used to specify all the data over time for a specific port. This helps to differentiate between the ports in the host viewer.

On the right-hand side of the main viewing window there is a sub window which contains a set of controls allowing the user to interact with the graph for a more detailed analysis of the data. A list box of IP addresses is used to select the host or subnet entity to be displayed in the main graphic window. NetBytes Viewer also gives the user control over setting which port data is displayed via a set of text boxes. Radio button groups allow for the data to be filtered based on data direction, protocol and volume measure (i.e. bytes, packets or flows). Sliders are also provided to select data points as will be described further in the next section.

3.2 User Interaction

Rendering a 3D graph on a 2D surface can cause several difficulties including occlusion, loss of depth perception and loss of head parallax (Ware, 2004). To deal with these issues, the overview screen is equipped with a number of interactive features. First, to help eliminate occlusion and perception issues, the 3D graph can be rotated around its center axis in all directions by holding down the left mouse button and dragging the view in any direction. This allows the user to view the data from all angles which facilitates the recognition of patterns and can also be used to disambiguate the 2D projection of the 3D graph. Furthermore, the user can drag the graph into two separate 2D orthogonal views (Volumes vs. Ports and Volumes vs. Time) as shown in Fig. 2. The "Volumes vs. Ports" view is useful to see the maximum network traffic per port while the "Volumes vs. Time" graph is useful for investigating peak traffic times.

Another issue with the 3D graph is that it can be difficult to perceive depth especially when there are several ports on the screen with associated data points. To help

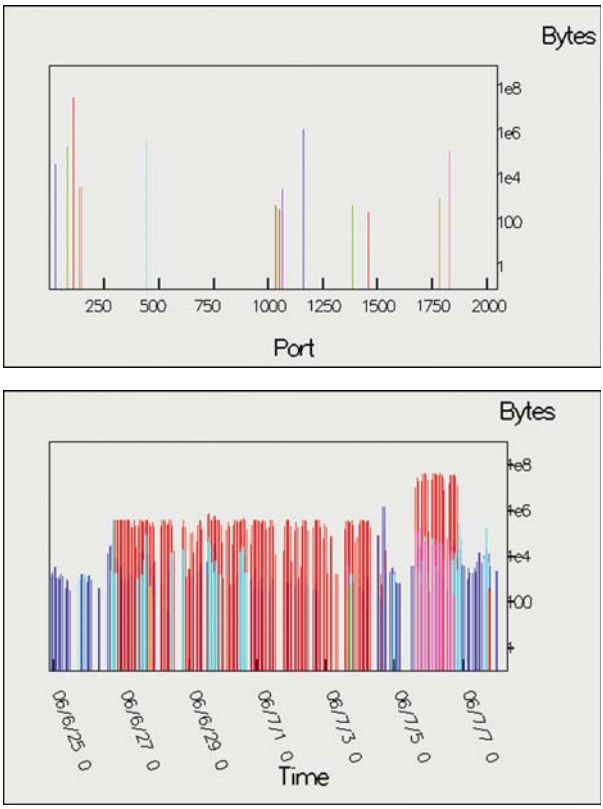

Fig. 2 Rotating the 3D graph into two separate 2D orthogonal views "Bytes vs. Ports" (*top*) and "Bytes vs. Time" (*bottom*)

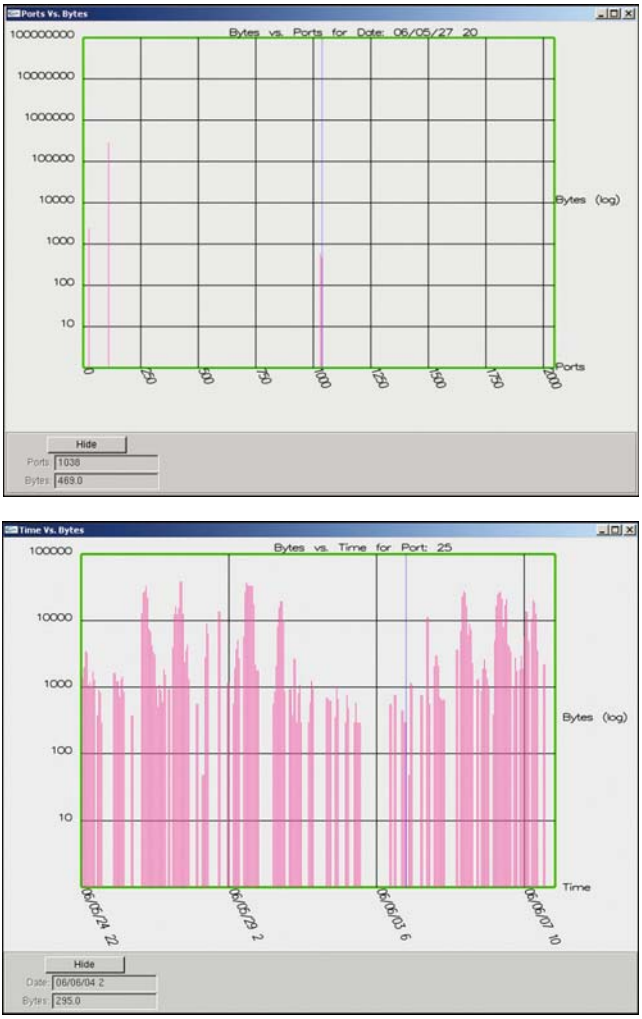

Fig. 3 "Bytes vs. Time" and "Bytes vs. Ports" graphs launched when application is in selection mode

the user deal with these issues, the NetBytes Viewer allows the user to look at data in more detail by launching a set of auxiliary 2D graphs (Volumes vs. Ports) and (Volumes vs. Time) as shown in Fig. 3. These 2D graphs are restricted to either a single time or a single port. This further reduces occlusion and perspective issues and increases the user's ability to interact with the data. The 2D Volume vs. Ports graph, as shown in Fig. 3 (*top*), can be launched by clicking on the **Enable Date Selector** selection box, while the Volume vs. Time graph (Fig. 3, *bottom*) can be launched by clicking on the **Enable Port Selector** selection box. Clicking on these boxes will not only launch the graphs in a separate window, but also enable the corresponding slider in the main 3D view. Using the slider, the user can select a point

on the time axis (if the date selector is enabled) or the user can select a point on the port axis (if the port selector is enabled).

When the **Port Selector** slider is enabled, a green semi-transparent highlight band is displayed along the port axis and a slider is enabled at the bottom of the screen as shown in Fig. 4 (*top*). When dragging the slider from side to side, the highlight band moves back and forth along the port axis and the corresponding 2D graph is updated in real-time with the time and volume data for the newly selected port. Clicking on the arrow buttons on either side of the slider causes the highlight band to jump to the next port which has data values. A similar highlight band (in blue) is displayable on the time axis when the **Date Selector** slider is enabled as shown in Fig. 4 (*bottom*).

It is worth noting that alternate approaches were considered for implementing the selection feature. Initially we allowed the user to click on a port number on the

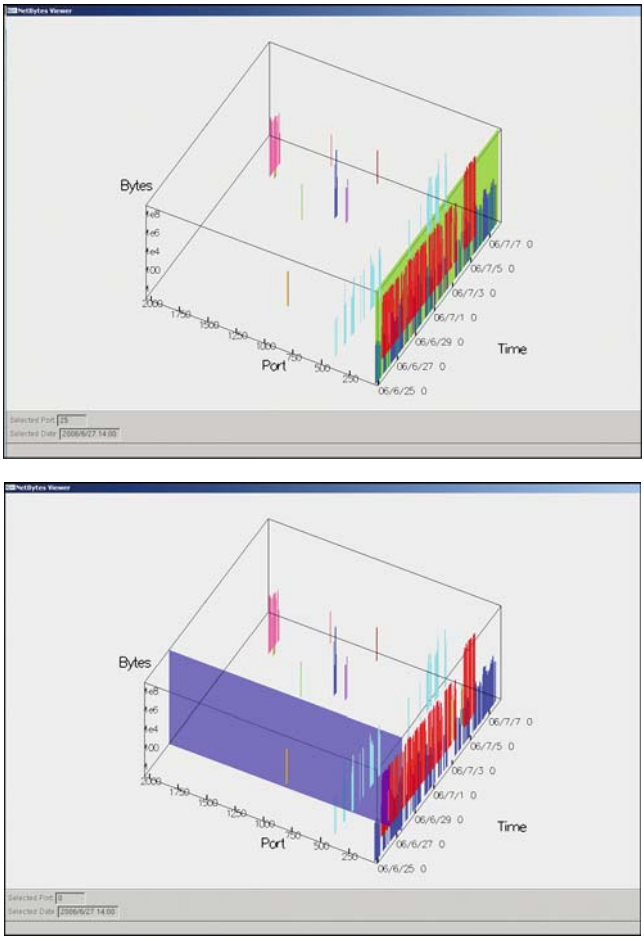

Fig. 4 Overview screen in port selector mode (*top*) and date selector mode (*bottom*)

port axis to highlight the data; however, with a large port range (around 2000+) this became impractical as it was difficult to accurately click on a port. A slider with snap buttons was more effective as the user is able to drag the slider to the vicinity of the intended port and then use the snap buttons to accurately select one.

The user can interact with the 2D graph using his or her mouse to highlight data points. Corresponding data values are displayed in the bottom left hand corner of the window, as can be seen in Fig. 3. These values are updated in real-time. Axes with a large number of data points can make it difficult to precisely highlight these values using the mouse; therefore, the **UP** and **DOWN** arrows were enabled on the window. Moving the cursor to a region on the graph and pressing the **UP** arrow selects the closest data point to the right-hand side of the cursor. Furthermore, pressing the **DOWN** arrow selects the closest data point to the left side of the cursor.

3.3 Implementation Details

NetBytes was written in C++ using the OpenGL graphics library (Woo et al., 1999). OpenGL was chosen for its power and flexibility in developing 3D graphics. OpenGL is a high performance graphics package written in C++ which is important when dealing with large amounts of data. Furthermore, it is platform independent which makes it useful for both Windows and UNIX users. Our aim is to release all tools as open source software on a variety of platforms.

NetFlows are gathered from a network border using the SiLK collection utilities and then processed with the SiLK bag tools. Afterwards, data is loaded into an SQL database by a small automated process. NetBytes uses this database when rendering its graphs. Using an intermediate database allows the application to capitalize on the more powerful SQL query language. It also ensures that the viewer can be used with other data analysis tools other than SiLK.

3.4 Case Studies

The goal of NetBytes Viewer is to enable administrators to quickly analyze the flow of traffic into and out of a host or subnet to determine if any malicious activity is occurring. A good example of where NetBytes Viewer proved useful was at a small networking research laboratory. The SiLK tools were installed to collect Netflow traffic at a border between the internal and external networks. As part of the research facility, the administrators house computers for external companies. Data was collected between February and March 2006 yielding some interesting results.

One host in particular produced fascinating visualizations indicating signs of systems being hijacked. Using NetBytes Viewer, evidence that the host was compromised is seen immediately as shown in Fig. 5 (*top*). As can be seen, there is

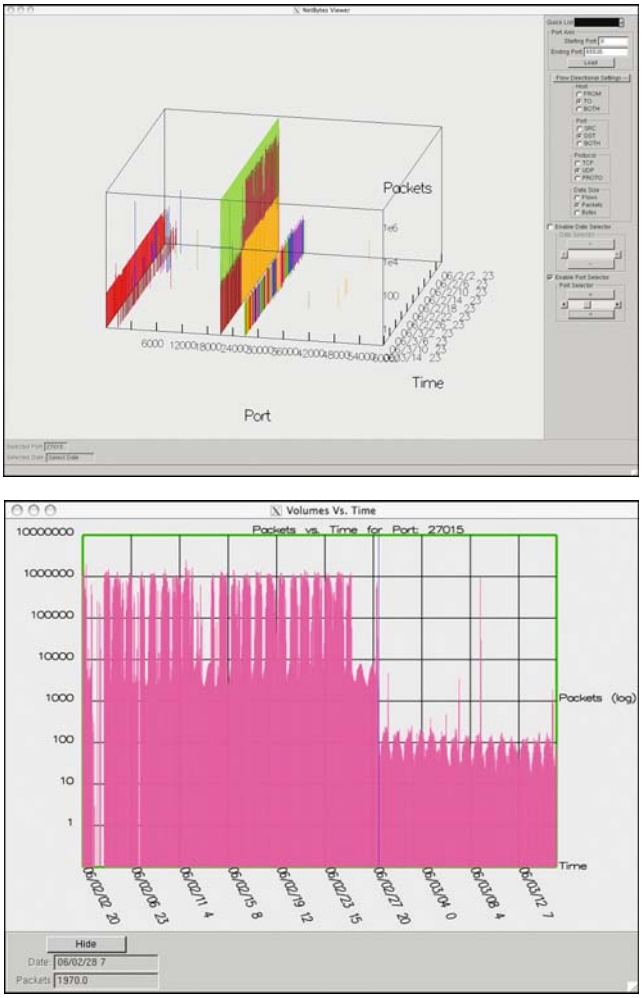

Fig. 5 Host compromised by game server (*top*) and port 27,015 (*bottom*)

a significant amount of incoming UDP traffic on ports 26,900 (yellow band) and 27,015 (brown band). These ports are typically utilized as game server ports for games like Half Life. The machine had been compromised and turned into a game server. Using the port selector feature, the port traffic was investigated in further detail in the 2D Packets vs. Time graph also shown in Fig. 5 (*bottom*). Many worms and viruses utilize high numbered ports to transfer data. NetBytes Viewer makes it easy to see the traffic on these higher ports to identify intrusion. Also of interest is the consistent traffic volumes on port 123 (red band in Fig. 5) which suggests this system is likely an NTP server for the network. Changing the filter parameters on the right control window allows the user to investigate the outgoing UDP traffic as

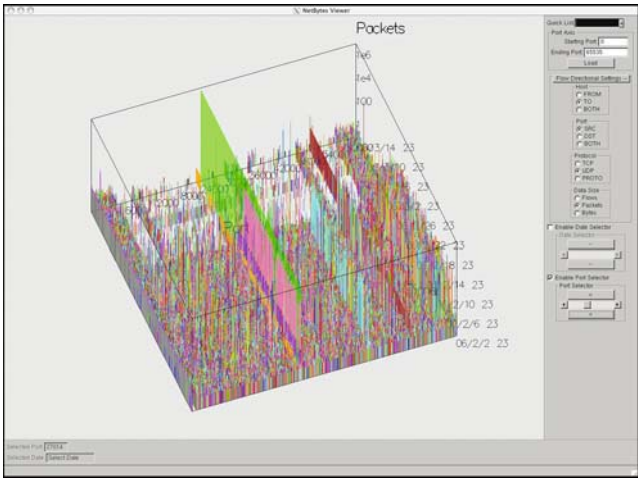

Fig. 6 Outgoing traffic for compromised host

shown in Fig. 6. Since the figure shows outgoing traffic, there is plenty of ephemeral port activity including two very prevalent bands of consistent data flow. One data flow is on port 27,014 (pink band that is highlighted by the port selector) and is likely the outgoing traffic for the game server running on the machine. The brown band maps to port 54,020 which seems to be typically used for client applications like FTP, messaging etc. on Linux servers.

Since NetBytes Viewer shows temporal data as well as traffic volume data on the same graph, administrators are able to see large data spikes in the context of normal traffic volumes. Excessive traffic volumes could be an indication of a denial of service attack or large file upload or download. Another SiLK data set from a private company showed a large spike in TCP traffic one evening during a two week collection period. This spike was clearly visible when using NetBytes Viewer using a volume-protocol-time graph. With a small amount of further investigation it was found that such a spike was the result of a planned back up of one of the company's data servers.

NetBytes Viewer is also good for looking at cyclic patterns in a network's traffic. For instance, in a company where workers would typically work from 9 to 5 Monday to Friday, one would typically expect the HTTP traffic to be relatively high through the week and almost non-existent on evenings and weekends. With NetBytes Viewer, network administrators can see these patterns by producing visualizations of entire subnets and networks and drilling down to see specific port traffic. If patterns do not exist, administrators can use the application on smaller subnets and hosts to track down the offending hosts. Ultimately, NetBytes allows the user to monitor how much data traffic is on his/her network over time and using this information can track down malicious behavior.

4 Future Work

The current NetBytes' feature set provides a good basis for an interactive host visualization; however, there are many more features that we intend to implement. A simple addition would be a mechanism for selecting a range of dates so that the user has more control over what is displayed in the 3D view. One way to do this might be to provide the entire date range on the selection slider, and then allow the user to highlight a range which would dynamically alter the corresponding axes on the graph. Another option is to provide a set of date pickers along the right-hand side of the main window, where the user could enter in the range values.

Another useful visualization technique that we intend to incorporate is non-linear distortion (Leung and Apperley, 1994). NetBytes is able to display all 65,000 ports as some viruses and Trojans are active on higher ports. But placing such a large port range on the screen at once causes the graph to become very large which can make it much more difficult to analyze. Since data tends to be sparse in higher port range, this range (2,048–65,000) could be distorted (compressed) to show all the data but in far less space. One way to do this might be to group ports into 256 partitions and collapse all the traffic for a partition into 1 aggregate impulse.

The application also requires a mechanism for displaying all the remote hosts that interact with the current host as well as the amount of data transferred per hour between the hosts. A possible approach might be to create another 3D graph which depicts "volumes vs. destination hosts vs. time". Another option might be to launch the 2D graph of "volumes vs. ports" and then list the remote hosts on a per port basis off the 2D graph.

Other features that will be explored include:

- The ability to filter based on protocol (UDP, TCP, ICMP, etc.), magnitude of bytes transferred per hour, and individual remote host IP addresses. The latter filter would be excellent to view what data is being transferred per port between two machines on a network.
- The 3D graph could be made bidirectional which enables the user to view network data on both the source and destination ports of a particular host.
- Standardize the scaling values across 2D graphs so that they can be more easily compared.
- Ability to generate PDF files or jpegs of the visualizations.
- Ability to snap the 3D graph to a 2D orthogonal view.
- Use flooding techniques to display accurate Byte values on the 3D graph.
- Integrate NetBytes more closely with the SiLK NetFlow analysis tools.

The NetBytes viewer will also become a component of a larger network level visualization tool which shows connections between entities as well as entity behavior. This tool will support drill down to follow links between entity levels and to identify the sources or destinations of the volumes contributing to a single impulse. A user study will also be conducted to analyze the impact of the visualization tool on a network administrator's ability to do his/her job. Such a study is invaluable in gaining feedback on the tool's usefulness.

5 Conclusions

In conclusion, NetBytes provides a useful technique for visualizing network traffic flows to an individual host using a 3D Volume vs. Ports vs. Time graph. Furthermore, it provides a set of interactive features enabling network administrators to discover anomalous traffic patterns that could indicate signs of a virus or Trojan. As a result, it offers an effective drill down mechanism for a larger network analysis visualization tool. Many features will be added to NetBytes to improve its ability to effectively help in the fight against network intrusion. This work can therefore be viewed as the initial stage of a comprehensive network visualization project.

Acknowledgements This work was supported by NSERC and the Canadian Foundation for Innovation.

References

Abdullah K, Lee C, Conti G, Copeland J A, Stasko J (2005) IDS RainStorm: visualizing IDS alerts. Proceedings of the IEEE Workshop on Visualization for Computer Security (VizSEC), 1–10

Ball R, Fink G A, North C (2004) Home-centric visualization of network traffic for security administration. Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security, 55–64

Fink G A, Muessig P, North C (2005) Visual correlation of host processes and network traffic. Proceedings of the IEEE Workshop on Visualization for Computer Security (VizSEC), 11–19

Komlodi A, Rheingans P, Ayachit U, Goodall J R, and Joshi A (2004) A user-centered look at glyph-based security visualization. IEEE Workshops on Visualization for Computer Security, 21–28

Lakkaraju K, Yurcik W, Bearavolu R, Lee A J (2004) NVisionIP: netflow visualizations of system state for security situational awareness. Proceedings of the ACM Workshop on Visualization and Data Mining for Computer Security, 65–72

Lau, S (2004) The spinning cube of potential doom. Communications of the ACM, 47(6):25–36

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

McHugh J, Gates C, Becknel D (2004) Situational awareness and network traffic analysis. Cyberspace Security and Defense: Research Issues, 209–228

McPherson J, Ma K, Krystosk P, Bartoletti T, Christensen M (2004) PortVis: a tool for port-based detection of security events. Proceedings of the ACM Workshop on Visualization and Data Mining for Computer Security, 73–81

Rensink R A, O'Regan J K, Clark J J (1997) To see or not to see: the need for attention to perceive changes in scenes. Psychological Science 8(5): 368–373

Ware C (2004) Information Visualization: Perception for Design. Morgan Kaufman, Los Altos, CA

Woo M, Neider J, Davis T, Shreiner D (1999) OpenGL Programming Guide: The Official Guide to Learning OpenGL. OpenGL Architecture Review Board, Addison-Wesley Professional

Yin X, Yurcik W, Li Y, Lakkaraju K, Abad C (2004) VisflowConnect: netflow visualizations of link relationships for security situational awareness. Proceedings of the ACM Workshop on Visualization and Data Mining for Computer Security, 26–34