The Design of VisFlowConnect-IP: a Link Analysis System for IP Security Situational Awareness *

Xiaoxin Yin William Yurcik Adam Slagell
National Center for Supercomputing Applications (NCSA)
University of Illinois at Urbana-Champaign
{xiaoxin,byurcik,slagell}@ncsa.uiuc.edu

Abstract

Visualization of IP-based traffic dynamics on networks is a challenging task due to large data volume and the complex, temporal relationships between hosts. We present the architecture of VisFlowConnect-IP, a powerful new tool to visualize IP network traffic flow dynamics for security situational awareness. VisFlowConnect-IP allows an operator to visually assess the connectivity of large and complex networks on a single screen. It provides an overall view of the entire network and filter/drill-down features that allow operators to request more detailed information. Preliminary reports from several organizations using this tool report increased responsiveness to security events as well as new insights into understanding the security dynamics of their networks.

In this paper we focus specifically on the design decisions made during the VisFlowConnect development process so that others may learn from our experience. The current VisFlowConnect architecture—the result of these design decisions—is extensible to processing other high-volume multi-dimensional data streams where link connectivity/activity is a focus of study. We report experimental results quantifying the scalability of the underlying algorithms for representing link analysis given continuous high-volume traffic flows as input.

Keywords*: NetFlow, security situational awareness, link analysis, security visualization*

1. Introduction

Without a doubt, there has been tremendous growth of network applications and services since the mid-1990s. Repercussions of this growth are multifaceted. First, there are new and extremely complex interactions among these many applications and services that are not yet fully understood. Secondly, society has come to rely on these services for everyday tasks (e.g. financial transactions, monitoring the news, communicating with colleagues, etc.). These complex interactions and the newly understood reliance on the services provided have brought network security into the lime light.

Computer and network security measures fall into three categories: prevention, detection, and reaction. Preventative measures often come in the form of policies (e.g. disable any unnecessary services) and user education on new protocols and best practices. The prototypical example of a detection mechanism is the Intrusion Detection System (IDS) [2, 10, 12, 17, 26]. While IDSs are no longer believed to be a panacea, they remain a vital part of the larger set of security mechanisms at a large organization. Reactive activities include, investigating breaches, closing off entry points discovered (this could be through patching, updating firewall rules or closing off services), and pursuing attackers through legal channels. More recently, IDSs are being replaced by Intrusion Prevention Systems (IPS). However, IPSs are not really a preventative measure. Instead, they are a detection and reaction mechanism in one. An IPS will first detect an attack, and then it will act by either dynamically changing a firewall rule, dropping a connection, isolating an attacker, alerting a security administrator or performing some other predefined reaction. Consequently, detection is still a vital part of any security system and has not been replaced by solely preventative measures.

Unfortunately, no detection system is completely effective. IDS/IPS technologies are still plagued by false positives and false negatives. Humans are still in the loop and must sort through alerts manually. While there is research on creating completely autonomic security systems that require little or no human intervention, we are nowhere close to realizing that goal. Since humans must be used in security, it is wise to focus the work of humans on activities at

* This research was supported in part by a grant from the Office of Naval Research (ONR) under the auspices of the National Center for Advanced Secure Systems Research (NCASSR) <http://www.ncassr.org>

which they excel more than computers. One thing we know from the cognitive sciences is that visualization is an important component in human learning. The human brain excels at visual pattern recognition. In fact, it has been shown in studies that visualization enhances comprehension for network situational awareness [15]. Consequently, the use of visualization can reduce reaction time to security incidents.

It is challenging to visualize all the information relevant to network and security administrators. At the NCSA we generate Gigabytes of log data each day. Processing such vast amounts of data on machines with limited storage and computational resources is a daunting task. We must be able to provide near real-time analysis, yet at the the same time be able to process information over the span of months for forensic investigations. Often we find these two goals to be contradictory. To deal with the storage and processing limitations, special data structures must often be created. Analysis is also complicated since not only must we show what is happening on each host, but we must visualize the relationships and interactions between all the hosts.

This paper describes the design and implementation of *VisFlowConnect-IP*, a tool for visualizing IP network traffic flows with a focus on the real-time connectivity between different IP hosts. VisFlowConnect-IP has the following distinct features that make it appropriate for providing situational awareness on IP-based networks.

- VisFlowConnect-IP allows users to visualize network traffic both between an internal network and the Internet as well traffic strictly within an internal network. This visualization is focused on recognizing connections and flows between hosts.
- VisFlowConnect-IP can process large network logs and high-speed stream data while consuming a constant amount of system memory. This enables it to run continuously for long periods of time.
- VisFlowConnect-IP can process network log records in real-time or show dynamic changes over time with animation.
- VisFlowConnect-IP provides an overview of network traffic flows, but it can also present details of specific flows upon demand.
- VisFlowConnect-IP has several filtering capabilities (e.g. filtering on protocols, ports, or flow sizes). This allows the user to focus on traffic flows of interest that otherwise might be obscured by noise.

VisFlowConnect-IP is based upon the general network visualization tool we developed called VisFlowConnect [28, 29]. While VisFlowConnect is focused on monitoring and analyzing IP networks (its design decisions optimized for this functionality), the VisFlowConnect framework can be easily adapted for more general purposes such as monitoring storage systems, clusters and interprocess communications.

The remainder of this paper is organized as follows. Section 2 summarizes related work. We present the VisFlowConnect-IP system architecture in Section 3. Section 4 describes its visualization capabilities and user interface. Section 5 presents experimental results highlighting scalability and visualization for security. We end with a summary and conclusions in Section 7.

2. Related Work

The vast amount of pertinent data available to a network administrator cannot be over-viewed effectively on a single screen of text. Because of the limited density of information relayed through text, visualization techniques to present computer network data sets to humans have been a growing area of research. It is well-known that seeing enables humans to glean knowledge and deeper insight from data—sight is the major sense in brain development. Interpreting images is perceptually a parallel operation in contrast to textual perusal which is an inherently serial process. Large information transfer per image reduces the amount of mental context switching and facilitates more efficient communications. Considerably more information can be presented via a static image (estimated to be about 150 Mbytes of information per computer screen) than a comparable volume of static text (estimated to be about 100 Kbytes per computer screen).

Thus visualization is a more intuitive and faster way for humans to understand large and multidimensional data sets. Our approach seeks to integrate innate human exploration abilities with the processing power of state-of-the-art computers to form a powerful knowledge discovery environment that capitalizes on the best of both tools.

2.1. Intrusion Detection Systems

Despite recent discussion that IDSs as an isolated protection device are obsolete, they remain an important component of a multi-level defense. Accordingly, there have been many research and commercial efforts attempting to improve IDSs. Currently there are two main approaches to intrusion detection: misuse detection and anomaly detection. Misuse detection finds intrusions by directly matching known attack patterns. The major drawback of this rule-based approach is that it is only effective at finding known attacks with predefined signatures [10, 12, 21, 17]. In anomaly detection, the expected behavior of a system is stored as profile. Any statistically significant deviations from this profile are reported as possible attacks, but these alarms may also be legitimate but atypical system activity.

For this reason and because attack traces are relatively rare within large volumes of data (the base rate fallacy), current anomaly detection IDSs suffer from unacceptable false positive alarm rates. However, anomaly detection does have the capability to detect so-called "zero-day" attacks not recognized by misuse detection tools [20, 18].

The novel approach we propose for computer network security is radically different from current IDS binary alarms from not-matching/matching specific traffic signatures or typical/atypical system patterns. Rather, we visually present the context of all traffic flows on a network to a human operator for evaluation. Of course some traffic flows will always indicate a malicious security event regardless of any context information, but a significant portion of suspicious or undetermined events can be most effectively and efficiently assessed if visualized in an IP space representation over time—network traffic flows may either be legitimate or malicious depending on situational context and this categorization is made easier with visualization as we will show.

2.2. Visualizing Security

Previous work on visualizing networks has been motivated by network management and analysis of bandwidth characteristics [9, 14, 6, 7, 11], but little work has been done on visualizing security. [16] presents a tool named Network Vulnerability Tool (NVT) that visually depicts a network topology and generates a vulnerability database.

In [22] the authors present a visualization of network routing information that can be used to detect inter-domain routing attacks and routing misconfigurations. In [24, 23] they explore further in this field and propose different ways for visualizing routing data in order to detect intrusions. An approach for comprehensively visualizing computer network security is presented in [15], where Erbacher et al. visualize the the overall behavioral characteristics of users for intrusion detection. Erbacher et al. represent the substantial characteristics of user behavior with a visual attribute mapping capable of identifying intrusions that otherwise would be obscured. However, the host representation employed in [15] is not scalable in terms of the number of hosts and traffic volume, generating overlapping images that are too dense to visually process.

Linkages among different hosts and events in a computer network contain important information for traffic analysis and intrusion detection. Approaches for link analysis are proposed in [4, 8, 25]. [4] and [25] focus on visualizing linkages in a network, and [8] focuses on detecting attacks based on fingerprints. Link analysis can illustrate interactions of different hosts either inside or outside a network system, thus provide abundant information for detecting intrusions.

Another closely related work is [3] that also focuses on log visualization using parallel axes. However, in [3] the visualized log data is from a single machine (web server). Since the plotted parameters resemble trellis plots for identifying trends in high dimensional data, the visualization is not very intuitive to a human viewer. In contrast, (1) VisFlowConnect-IP is able to view all traffic in a large network (Class B address space) in real-time, thus solving a key problem of scalability acknowledged as a limitation within [3], and (2) the parallel axes visualization of VisFlowConnect-IP is more intuitive and allows an operator to interact with the data and make immediate inferences about events of possible security significance. A unique feature found in [3] but not in VisFlowConnect-IP is the aggregation of multiple parallel axes plots into a small multiple view for visual comparison between different plots to quickly identify similarities and differences. This is similar to the small multiple view of NVisionIP [19], which serves the same purpose of visual comparison.

2.3. NVisionIP

In [30] and [19] the authors present *NVisionIP*, a security visualization tool that shows network traffic flows from a host-centric view. The visualization approach behind NVisionIP is to present an overview first, zoom and filter capabilities next, and then details on demand. In the overview/galaxy view, NVisionIP represents an entire class B IP address space 64,000 hosts on one screen with attributes highlighted such as traffic volume, number of connections, and port or protocol activity—color and shape representations can be chosen by users on demand. Drill-down views to a subnet small-multiple view and an individual machine view allow operators to make analytical observations of flow activity. Currently, NVisionIP is in the final stages of development and testing with security engineers.

The military term "situational awareness" refers to a commander knowing where his troops are, their readiness and capabilities [5], and most importantly intelligence on the location of enemy troops, their readiness and capabilities. For information assurance on large and complex networks there is a "fog of war" similar to a battlefield in that there is too much data but not enough situational awareness about what is actually happening and its ramifications for security. Both NVisionIP and now VisFlowConnect are prototype tools designed to bridge this uncertainty gap by providing information humans can use for network security decision-making in real-time. VisFlowConnect complements NVisionIP by showing flow connectivity to discover patterns and otherwise obscured relationships. The ultimate goal is to couple the host-centric view of NVisionIP with the network-centric view of VisFlowConnect for comprehensive situational awareness.

3. System Architecture

The general system architecture of VisFlowConnect-IP is shown in Figure 1. VisFlowConnect-IP has three main components: (1) An agent that reads in NetFlow records. (2) A NetFlow analyzer that analyzers the raw data and stores important statistics. (3) A visualizer that converts the statistics into animations. In this section we describe, in detail, the design and implementation of each component of VisFlowConnect-IP.

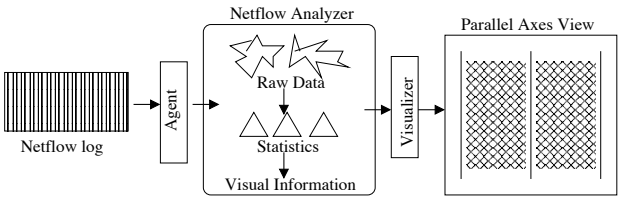


Figure 1. General System Architecture of VisFlowConnect-IP

3.1. NetFlow Source Data

The source data used by VisFlowConnect-IP is derived from Cisco NetFlow data. Other router vendors followed Cisco in implementing this functionality, but NetFlows are now available independent of router access via open source PC software.[1] For this NetFlow application, a distinct flow is defined as either a unidirectional TCP connection (where a sequence of packets take the same path) or individual UDP packets separated beyond a short time threshold (UDP packets closer than the threshold are considered as one flow if they have the same IP address/port number associations).

The input to VisFlowConnect-IP is a stream of NetFlow records either from a log file or a streaming socket. A NetFlow agent is used to retrieve the NetFlow records and feed them into VisFlowConnect-IP. Each record contains the following information: (1) the IP addresses and ports of the source and destination, (2) the number of bytes and packets, (3) start and end times, and (4) the protocol type.

3.2. NetFlow Agent

The NetFlow agent is responsible for reading in NetFlow records and sending them in chronological order to VisFlowConnect-IP. This is reflected by the agent's two major subcomponents: (1) the record filter, and (2) the sorter.

The agent has filtering capabilities based on different properties of the records. It can filter based on the IP addresses, protocols, ports, and flow sizes. Only records having the selected properties are kept, and all others are discarded. Default filtering includes all ports, protocols, flow sizes, and IP addresses.

A troublesome problem in the NetFlow data is that the records are not strictly sorted by chronological order. The records are somewhat sorted by their end time in the macro sense, which means that if the end time of record 1 is much earlier than record 2, record 1 will appear before record 2. However, they are not sorted in the micro sense, which means that if the end time of two records are close to each other, either one may appear first. In order to visualize network traffic with time, VisFlowConnect-IP has to retrieve NetFlow records in chronological order. Thus the agent must be able to reorder the records according to their time stamps.

To handle this temporal disorder problem, a buffer is used in the agent to hold records of the most recent period (e.g. ten minutes). All records in this buffer are sorted by end time. When a new record comes, it is inserted into the appropriate position in the buffer. If this record is too old, it is considered to be out of date and is ignored. When the system requests the next record from the agent, the oldest one is sent out. The working procedure of the buffer is shown in Figure 2.

Figure 2. Record Buffer of NetFlow Agent

3.3. Internal Data Storage of VisFlowConnect-IP

The input to the NetFlow analyzer is raw NetFlow records. In order to visualize the network connectivity by animation with respect to time, the following information is needed.

- *Requirement 1*: the traffic volume between each domain outside our network (or each host in a certain domain) and each host inside our network.
- *Requirement 2*: the traffic volume between each pair of hosts in our network.

1 ARGUS open source software for generating NetFlows <http://www.qosient.com/argus/>

- *Requirement 3*: if traffic of a certain port is being monitored, we need the above information for traffic to that port.

VisFlowConnect-IP uses a traffic statistics repository, which contains traffic information for both traffic inside our local network, and traffic between hosts outside our network and those inside. For traffic between outside and inside hosts, the repository maintains traffic statistics of every outside domain, which contains its total traffic volume to and from each host in our network. For traffic inside our local network, the repository maintains traffic statistics of every sender, which contains its total traffic volume to every receiver.

To enable efficient visualization, we have the following requirements for data storage.

- *Requirement 1*: all information about a certain host (or domain) can be found efficiently, because such information needs to be repeatedly drawn on the screen.
- *Requirement 2*: the hosts (or domains) should be sorted by their IP addresses because they will be drawn by lexicographical order of IP addresses.
- *Requirement 3*: when a new record comes into the system, the stored information can be updated efficiently in nearly constant time.

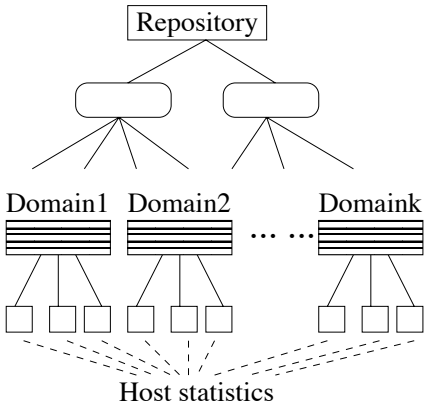


Figure 3. Data structure of traffic statistics repository

To meet the above requirements, we use the data structure shown in Figure 3 for the traffic repository. All domains (or hosts) are stored in a search tree, which sorts all data nodes according to their IP addresses to enable fast look up. Each leaf node of the tree stores the traffic statistics involving a certain domain, including the volumes of traffic from this domain (or host) to every host in our network or those in reverse directions. This information is stored in a hash-table which enables fast look up. When visualizing traffic, the information about each domain (or host) can be either retrieved in blocks of IP addresses or individually by a specified address. The statistics of each domain (or host) can also be retrieved efficiently.

Another traffic statistics repository with a similar data structure is used for traffic inside our network. The only difference is that domains are replaced by senders of traffic.

When a new record comes into the system, the node for the domain (or host) can be found efficiently via the search tree, and the relevant statistics are updated. In this node, the entry for a particular host can also be found efficiently by use of the hash-table, and relevant statistics are also updated. Likewise, when an old record leaves the time window, the relevant information is updated in the same way.

The network administrator may often be interested in traffic of a particular type. Therefore, besides monitoring the overall traffic, VisFlowConnect-IP is also capable of monitoring traffic on specific ports. When the user specify a port to be monitored, VisFlowConnect-IP will create a new traffic statistics repository for traffic on that port. However, since there is much less traffic on only a single port, the repository for just that port is usually much smaller than that of all traffic. An example data structure is shown in Figure 4.

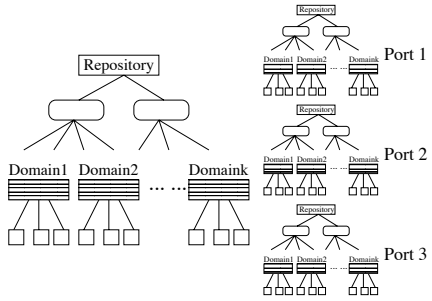


Figure 4. Data structure of traffic statistics repository of three ports

3.4. Sliding Time Window

Usually the user is only interested in monitoring recent traffic instead of cumulative traffic. Consequently, a *Time Window* is used in VisFlowConnect-IP, which is set by the user. For example, if the time window is ten minutes, then only traffic during the last ten minutes is present in the

repository. In this way, the user can easily see the changes in the traffic between different pair of hosts (or domains). An illustration of the time window is shown in Figure 5.

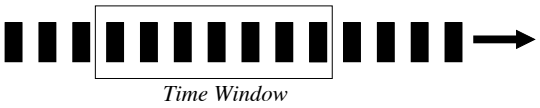


Figure 5. Time Window

VisFlowConnect-IP uses a buffer of NetFlow records for each time window. This buffer is a queue that contains all records in the time window. When a new record comes in, it is put to the end of the buffer. At the same time, the head of the buffer is checked and all old records not within the time window are removed from the buffer. The buffer is also checked periodically regardless of whether new records arrive. When a record comes into the buffer or is removed from the buffer, the corresponding traffic statistics are updated in the repository. With the use of a sliding time window, information about traffic will not accumulate inside VisFlowConnect-IP. This enables the user to see latest information and VisFlowConnect-IP to run continuously for long periods of time.

3.5. Creating Animation

The traffic statistics in the repository are converted into animations by the visualizer. The information in repository is updated in real-time. The visualizer creates a new image after each time interval (e.g. 0.5 seconds) or after reading a certain number of NetFlow records. In this way the user will see animation that reflects current traffic status.

Sample images created by the visualizer are shown in Figure 8 and Figure 9. To create an image according to traffic statistics in the repository, the visualizer goes through all hosts/domains in the repository and draws a line between host-pairs (or external-domain↔internal-host pairs) based on traffic flows and their intensities. Because of the limited resolution of the monitor, at most several hundred of hosts/domains can be distinguished, together with traffic between them. This is for typical NCSA traffic volumes, but it may be a constraint for another network. However, filtering may be used to adjust the amount of data visualized during periods of overload.

Because of the limited resolution of computer monitors, VisFlowConnect-IP can usually display a few hundred of domains or hosts on a vertical axis. If they are allocated at fixed positions that are computed from their IPs, there are usually many domains (or hosts) allocated to a very small region, making it very hard for human to distinguish them. For example, suppose the domain being monitored in 10.9.x.x, which contains 65536 IPs. If all these 65536 IPs are distributed uniformly, with each IP at a fiexed position, then it is quite likely that there are many servers in the range from 10.9.1.1 to 10.9.1.255, which are put into such a small region that no one can distinguish them. On the other hand, if domains or hosts are randomly allocated, then a certain domain (or host) will appear in randomly positions from time to time, as new domains (or hosts) keep coming into the view and old ones going out of it. In that case it will be impossible for human to track a certain domain (or host) and observe its long-term behavior.

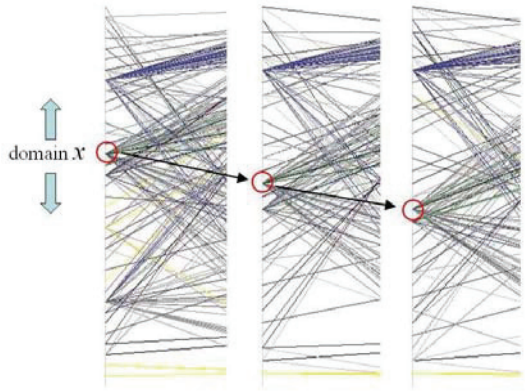


Figure 6. Domains or Hosts can move up and down along vertical axis

In VisFlowConnect-IP we use a simple but effective approach to allocate the domains or hosts on a vertical axis. All domains (or hosts) are ordered by their IP addresses and distributed uniformly on the axis. As animation goes on, new domains (or hosts) may come into the view and old ones go out. Domains (or hosts) may move up and down along the vertical axis (as shown in Figure 6. But the position of each domain (or host) remains fairly stable, and the inter-connection of different domains (or hosts) remains stable. Figure 6 shows three frames, with five minutes time interval between each two of them. It can be seen that the highlighted domain moves down gradually, and it is very easy to human to track it because its connections with other domains (or hosts) remain stable.

3.6. System Analysis

3.6.1. Efficiency VisFlowConnect-IP can handle high-speed stream data. When a NetFlow record is input into the system, it is stored in the record buffer queue, and its

data is used to update the traffic statistics in the repository. When a record gets old and is outside of the time window, it is removed from the record buffer queue, and the traffic statistics are updated. Therefore, the time consumed for processing each record is very small. To visualize traffic statistics, the visualizer just creates images which contain the same information as the traffic repository. The efficiency of this approach allows VisFlowConnect-IP to execute responsively on all laptops we have tested, without any report of problems to date.

3.6.2. Scalability Because a sliding time window is used, only NetFlow records within the time window is stored in the repository, which makes VisFlowConnect-IP capable of handling stream data for long periods of time. For larger time windows, more records need to be stored in the repository, and the memory consumption is greater. However, the time required to process each record is still the same; thus the processing speed is not affected.

VisFlowConnect-IP can only visualize a limited number of domains or hosts because of the limited resolution of computer monitors and limited resources for computation and storage. An overload detector is added to VisFlowConnect-IP, which filters out unimportant domains (or hosts) when the system is overloaded. In details, when VisFlowConnect-IP is unable to show all domains (or hosts) on one axis, the domains (or hosts) with low total traffic volumes will be filtered out by the overload detector, and only those with high traffic volumes will be visualized.

4. Visualizing Network Flows

VisFlowConnect-IP provides an excellent framework for visualizing network traffic, but it can be used for different types of visualization as well. In this section, we describe the visualization of the VisFlowConnect-IP environment.

In general, traffic flow visualization is most suited to show specific security events, such as a host that has an unusually high traffic volume or drastic changes in traffic volume within a time window. To detect potentially suspicious behaviors, a visualization system must have the following features: (1) The traffic volume between each pair of hosts is visualized and may be highlighted for more detailed information, (2) the traffic flows on all ports can be visualized including queries for more detailed information on specific ports, (3) the traffic flows on all protocols can be visualized including queries for more detailed information on specific protocols, and (4) the traffic flows are visualized dynamically over time using animation to highlight changes in network attributes.

Figure 7 is the VisFlowConnect-IP visual interface with features labeled. Based on the visualization principles we have enumerated, we use *the Parallel Axes View* as a basis for human-computer interaction. In more detail we see that:

1. Three vertical parallel axes are used to indicate traffic between external domains (on the Internet) and internal hosts within an organizational network (see Figure 8). Points on the left vertical axis represent external domains that are sourcing flows inbound to the internal network. Points on the right vertical axis represent external domains that are receiving flows outbound from the internal network. The hosts on these two outer vertical axes are placed symmetrically. Points on the middle vertical axis represents internal hosts. All points are sorted according to their IP addresses, so that each point will remain at a relatively stable position for user to track.
2. Figure 9 shows traffic just between internal hosts (source and destination are both internal hosts). The points on the left axis represent the source of traffic flows, and the points on the right axis represent the destination of traffic flows. The user may switch between two views by clicking on the button “Show Inside/Outside”.

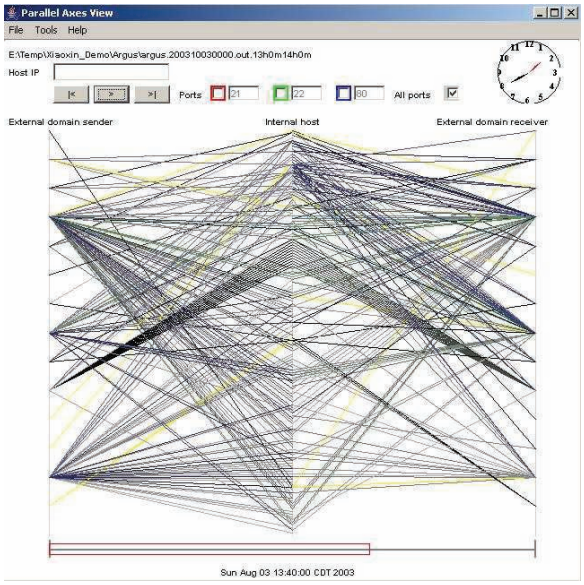


Figure 8. Parallel Axes External View (Default View)

3. The darkness of a line between two points is proportional to the logarithm of traffic volume between the hosts. The user may select any host by clicking on it,

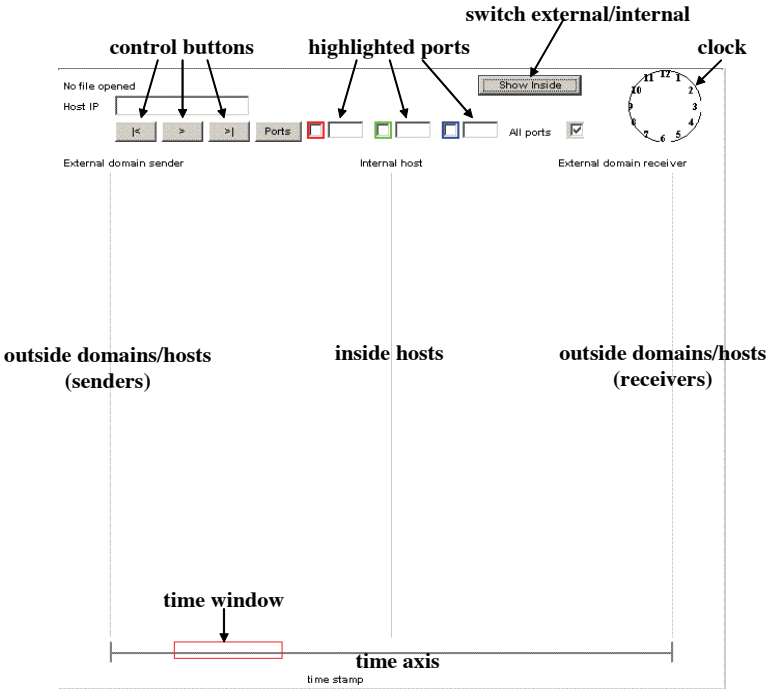


Figure 7. VisFlowConnect-IP: Annotated Parallel Axes View

and all traffic involving that host will be highlighted in pink.

4. Port activity can be visually highlighted by entering up to three different port numbers. Traffic on specified ports is shown in red, green, or blue (see Figure 8). The user may also click on the check box to show only traffic on a particular port.
5. Animation is used to visualize flow changes over time. The clocks show the current time. Only flows within a specified time window are shown (e.g. default 30 minutes from current time). A sliding rectangle along a horizontal time axis is shown at the bottom of the view to indicate the subset of flows being visualize. A user can control the animation with the three buttons at the top of the Parallel Axes View: ($|\ <$) rewind back to start, ($>$) play forward a defined time unit (e.g. default 5 minutes), and ($> |$) play forward to the end of the data set.

As shown in figure 10, we incorporate a drill-down Domain View within VisFlowConnect-IP so a user can visualize traffic between hosts in a specific external network domain to/from hosts in the internal network. The Domain View shows all flows from individual hosts within the corresponding external network domain to/from the internal network. The Domain View also uses animation for visualization and supports functions such as port specification.

During the initialization of VisFlowConnect-IP a user may specifically configure VisFlowConnect-IP by specifying parameters within a setting dialog box as shown in Figure 11. For example, a user may specify the protocol (TCP, UDP, ICMP, other), the traffic threshold, and the period of the time window. Default values are used if users does not make a selection.

Figure 12 shows VisFlowConnect-IP's ability to show details-on-demand. A user may want a detailed inspection of flows involving a particular host. VisFlowConnect-IP provides a function that shows a cumulative summary of flow data between a particular host (either external or internal) and all other hosts.

5. Experiments

Experiments were performed to show both the scalability and the effectiveness of VisFlowConnect-IP. We ran VisFlowConnect-IP on a RedHat 9.0 PC with 2.4GHz Pentium 4 CPU and 1GB memory.

5.1. Scalability

To test the scalability of VisFlowConnect-IP, we tested it with a NetFlow log file containing 1.8 million records (a 24 hour period for our network). We first tested time and space

Figure 9. Parallel Axes Internal View

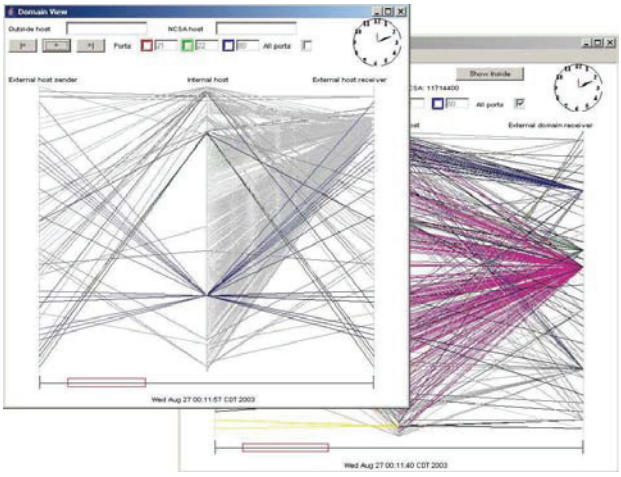


Figure 10. Domain View (from within a Parallel Axes View)

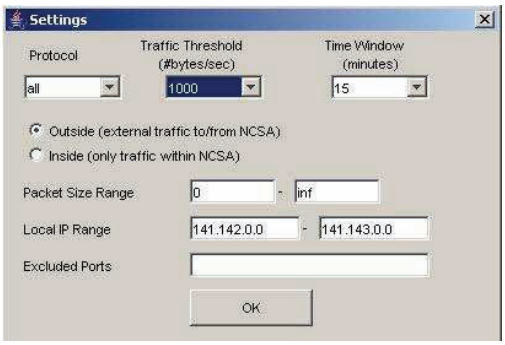


Figure 11. VisFlowConnect-IP Setting Dialog Box

scalability with respect to the number of records, as shown in Figure 13. The time window is set to 60 minutes. Variance is not explicitly indicated in the Figure 12 since the 95% confidence intervals were tight to the values.

It can be seen that VisFlowConnect-IP scales linearly with respect to the number of records, and consumes constant memory. The memory consumption is small for the first 300k records because the log file starts from midnight, and there is less traffic in each time window during the early morning.

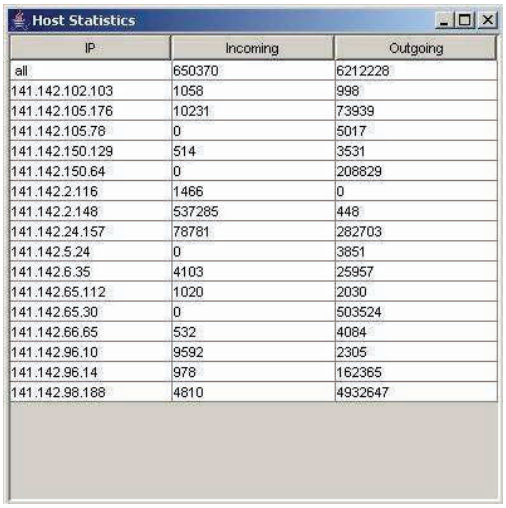
Figure 14 shows the time and space scalability with respect to the size of time window for VisFlowConnect-IP. It can be seen that the memory consumption increases with the time window size, while the running time decreases. This happens because when time window is small, hosts (or domains) keep entering and leaving the traffic statistics repository—a time consuming operation. Variance is not explicitly indicated in the Figure 13 since the 95% confidence intervals were tight to the values.

VisFlowConnect-IP is capable of monitoring traffic upon user specified ports. Because there is usually little traffic on a single port, monitoring a specific a port does not increase the cost significantly. Figure 15 shows the scalability with respect to the number of specially monitored ports. Variance is not explicitly indicated in the Figure 13 since the 95% confidence intervals were tight to the values. The three ports specified in tests were port 80 (HTTP), port 22 (SSH), and port 21 (ftp).

5.2. Visualization

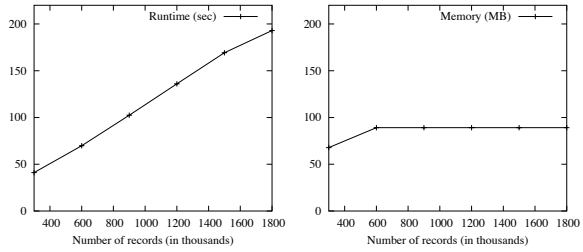
Through the visual interface of VisFlowConnect-IP, the network administrator can identify interesting traffic patterns. Abnormal network events usually have peculiar corresponding patterns that can be identified using VisFlowConnect-IP.

5.2.1. Access Patterns of Clusters Clusters have different NetFlow patterns than individual hosts—an emergent property where the whole is more the sum of its parts. We studied the traffic flows among hosts in two large Linux clusters at the NCSA. Figure 16 shows cluster flows within the Domain View. Even without colors to accentuate the



IP	Incoming	Outgoing
all	650370	6212228
141.142.102.103	1058	998
141.142.105.176	10231	73939
141.142.105.78	0	5017
141.142.150.129	514	3531
141.142.150.64	0	208829
141.142.2.116	1466	0
141.142.2.148	537285	448
141.142.24.157	78781	282703
141.142.5.24	0	3851
141.142.6.35	4103	25957
141.142.65.112	1020	2030
141.142.65.30	0	503524
141.142.66.65	532	4084
141.142.96.10	9592	2305
141.142.96.14	978	162365
141.142.98.188	4810	4932647

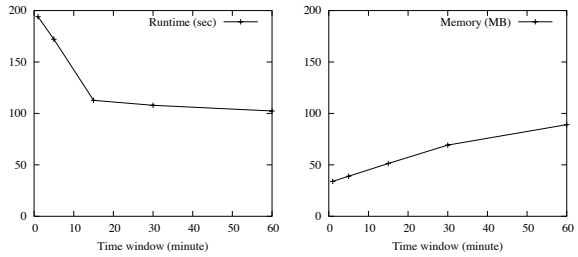
Figure 12. Host Flow Details on Demand



a) Processing vs. Record Volume

b) Memory Space vs Record Volume

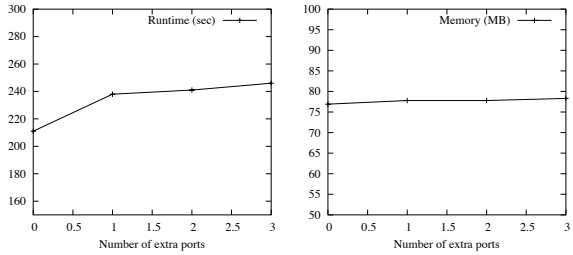
Figure 13. Processing and Memory vs. Record Volume



a) Processing vs. Time Window

b) Memory Space vs. Time Window

Figure 14. Processing and Memory vs. Time Window Size



a) Processing vs. Ports Monitored

b) Memory Space vs Ports Monitored

Figure 15. Processing and Memory vs. Number of Extra Ports

flows you can still discern a small number of external hosts connected to a large number of internal hosts in a fan-out pattern. Without situational context, such a pattern is likely to be attributed to reconnaissance port scanning or IP mapping often used as a precursor to an attack. After verifying the IP addresses, we confirmed that this was actually legitimate traffic between external hosts and an NCSA cluster on the internal network.

Figure 17 shows a different flow pattern between external and internal hosts within a Domain View. We found that that external domain contained a set of hosts with sequential IP addresses communicating directly with an NCSA cluster. These two sets of hosts have one-to-one communications between the two clusters (cluster-to-cluster or Grid communications).

Thus VisFlowConnect-IP shows us that cluster traffic flow patterns are quite different from the patterns of individual hosts, and knowing the context of flows as part of cluster communication can help distinguish legitimate from malicious traffic. This is also very useful to cluster security since they are vulnerable to class break attacks—one machine compromises an entire cluster.

5.2.2. Blaster Attacks The blaster virus was a very effective virus that infected many computers in our network as well as around the world. Once infected, a computer will send out packets to all other computers in certain domains, and the traffic signatures to different destination are very similar. These kind of patterns can be easily detected by VisFlowConnect-IP. In Figure 18 one can see that there is one domain who connects to almost every host in our network, with nearly uniform traffic volume to every destination. This indicates that some computers in that domain might be infected by blaster viruses.

In Figure 19 we focus our inquiry with the domain view. Now the infected computers can be easily found. Further, we find records involving those computers from the log file, and notice that one feature of the blaster virus is that the in-

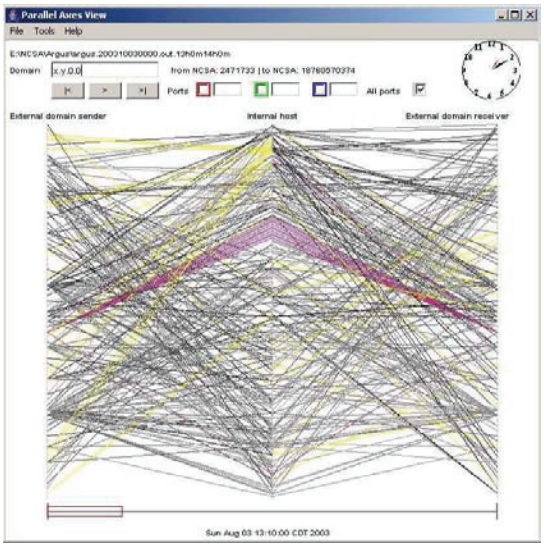


Figure 16. Parallel Axes View: Grid Patterns

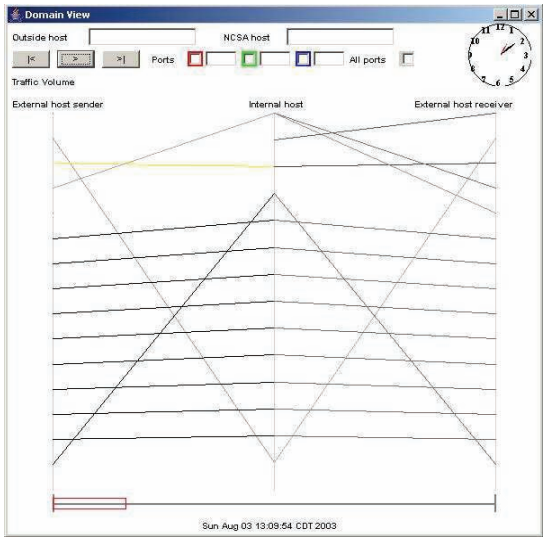


Figure 17. Domain View: A Specific Grid Pattern

fected machines send out an inordinate number of packets 92 bytes long. This feature can greatly help us in identifying infected machines.

6. Conclusions

In this paper we have presented the design of VisFlowConnect-IP, a powerful tool for visualizing network traffic flows. It uses an efficient algorithm for storing dynamic statistics of network traffic, which makes it highly scalable. By the usage of a sliding time window, VisFlowConnect-IP consumes constant memory and is able to run continuously for long periods of time.

VisFlowConnect-IP visualizes network traffic flows by animation, and provides an overall network view as well as drill-down functions that help users find more detailed information. Using VisFlowConnect-IP, users visually train on normal flow behavior in order to distinguish it from abnormal flow behavior, focus on abnormal flow behavior signatures, and/or determine the root causes of security events. We have shown several experiments in which flow behaviors were quickly inferred using VisFlowConnect-IP. These preliminary results show that VisFlowConnect-IP is a promising new tool for providing security situational awareness.

VisFlowConnect-IP is available for download at http://security.ncsa.uiuc.edu/distribution/VisFlowConnectDownLoad.html. A sample NetFlow file is currently available for demonstration purpose at this time of publication, and we plan to make more NetFlow files available in future.

7. Acknowledgments

We would like to first thank our SIFT research group colleague Yifan Li for his help in processing NetFlow log data for us to visualize. We would next like to thank the remaining members of the SIFT research group (in alphabetical order): Cristina Abad, Ratna Bearavolu, Kiran Lakkaraju, Adam Lee, Ramona Su, and Michael Treaster for their invaluable help in implementing and improving VisFlowConnect-IP. Lastly we thank the anonymous reviewers for improving presentation of this work.

References

[1] C. Abad, Y. Li, K. Lakkaraju, X. Yin, W. Yurcik. Correlation Between NetFlow System and Network Views for Intrusion Detection. *Workshop on Link Analysis, Counter-terrorism, and Privacy held in conjunction with SDM 2004*, 2004.

[2] C. Abad, J. Taylor, C. Sengul, W. Yurcik, Y. Zhou, K. Rowe. Log Correlation for Intrusion Detection: A Proof of Concept. *Annual Computer Security Applications Conf. (ACSAC)*, 2003.

[3] S. Axelsson. Visualisation for Intrusion Detection - Hooking the Worm. *Eighth European Symposium on Research in Computer Security (ESORICS)*, Lecture Notes in Computer Science (LNCS) 2808 , Springer, 2003.

[4] R. Ball, G. A. Fink, C. North. Home-Centric Visualization of Network Traffic for Security Administration. *CCS Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC)*, 2004.

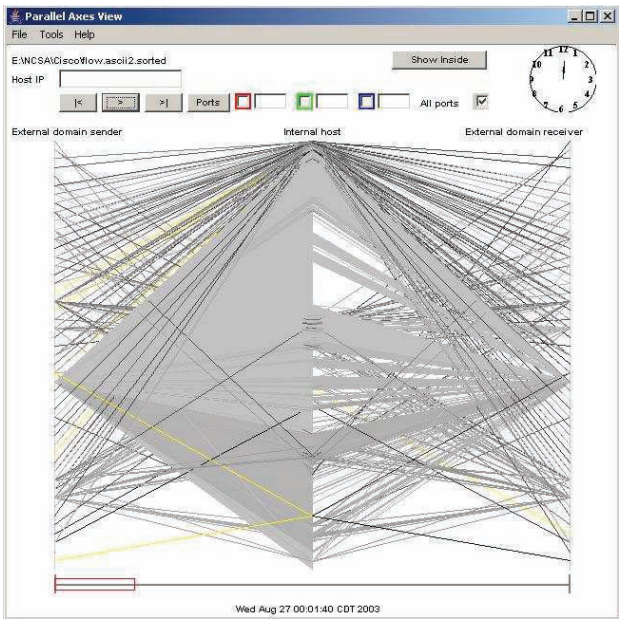


Figure 18. External view of blaster attacks

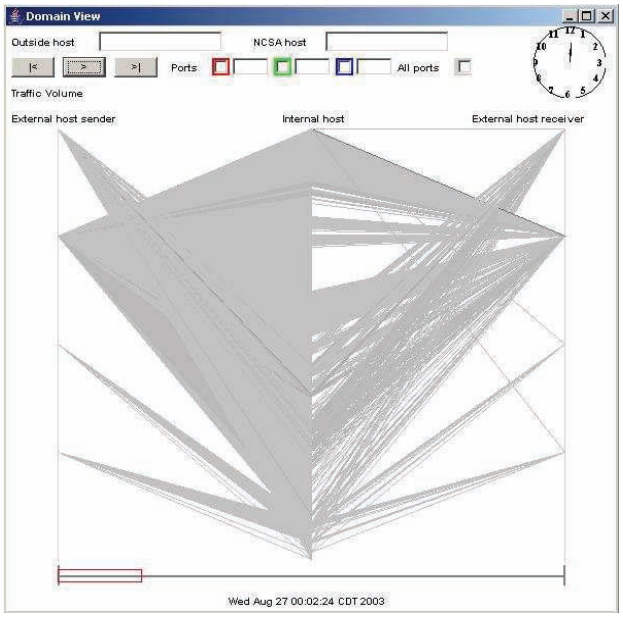


Figure 19. Domain view of blaster attacks

[5] R. Bearavolu, K. Lakkaraju, W. Yurcik, H. Raje. A Visualization Tool for Situational Awareness of Tactical and Strategic Security Events on Large and Complex Computer Networks. *IEEE Milcom*, 2003.

[6] R. Becker, S. Eick, A. Wilks. Visualizing network data. *Readings in Information Visualization: Using Vision to Think*, 1999.

[7] T. Bray. Measuring the Web. *Readings in Information Visualization: Using Vision to Think*, 1999.

[8] G. Conti, K. Abdullah. Passive Visual Fingerprinting of Network Attack Tools. *CCS Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC)*, 2004.

[9] K. Cox, S. Eick, T. He. 3D Geographic Network Displays. *ACM SIGMOD Record*, 25(4):50–54, 1996.

[10] F. Cuppens, A. Miege. Alert Correlation in a Cooperative Intrusion Detection Framework. *IEEE Symp. on Security and Privacy*, 2002.

[11] C. Davidson. What Your Database Hides Away. *New Scientist*, 1993.

[12] H. Debar, A. Wespi. Aggregation and Correlation of Intrusion Detection Alerts. *RAID*, 2001.

[13] M. Dodge, R. Kitchin. Atlas of Cyberspace. Addison-Wesley, 2001.

[14] S. Eick, G. Wills. Navigating Large Networks with Hierarchies. *IEEE Visualization*, 1993.

[15] R. Erbacher, K. Walker, D. Frincke. Intrusion and Misuse Detection in Large-Scale Systems. *IEEE Comp. Graphics and Applications*, 22(1):38–48, 2002.

[16] R. Henning, K. Fox. The Network Vulnerability Tool (NVT) – A System Vulnerability Visualization Architecture. *NISSC*, 2000.

[17] C. Krugel, T. Toth, C. Kerer. Decentralized Event Correlation for Intrusion Detection. *Intl. Conf. on Info. Sec. and Cryptology (ICISC)*, 2001.

[18] C. Krugel, T. Toth, E. Kirda. Service Specific Anomaly Detection for Network Intrusion Detection. *ACM Sym. on Applied Computing*, 2002.

[19] K. Lakkaraju, W. Yurcik, A. J. Lee, R. Bearavolu, Y. Li, X. Yin. NVisionIP: NetFlow Visualizations of System State for Security Situational Awareness" *CCS Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC)*, 2004.

[20] W. Lee, D. Xiang. Information-Theoretic Measures for Anomaly Detection. *IEEE Sym. on Sec. and Privacy*, 2001.

[21] W. Lee, S. J. Stolfo, K. W. Mok. A Data Mining Framework for Building Intrusion Detection Models. *IEEE Sym. on Sec. and Privacy*, 1999.

[22] S. T. Teoh et al. Elisha: a Visual-based Anomaly Detection System. *RAID*, 2002.

[23] S. T. Teoh, K. Ma, S. F. Wu. A Visual Exploration Process for the Analysis of Internet Routing Data. *IEEE Visualization*, 2003.

[24] S. T. Teoh, K. Ma, S. F. Wu, X. Zhao. Case Study: Internet Visualization for Internet Security. *IEEE Visualization*, 2002.

[25] S. T. Teoh, K. Zhang, S. Tseng, K. Ma, S. F. Wu. Combining Visual and Automated Data Mining for Near-Real-Time Anomaly Detection and Analysis in BGP. *CCS Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC)*, 2004.

[26] A. Valdes, K. Skinner. Probabilistic Alert Correlation. *RAID*, 2001.

[27] X. Yin, K. Lakkaraju, Y. Li, W. Yurcik. Selecting Log Data Sources to Correlate Attack Traces for Computer Network Security: Preliminary Results. *11th Intl. Conf. on Telecom. Systems*, 2003.

[28] X. Yin, W. Yurcik, Y. Li, K. Lakkaraju, C. Abad. VisFlowConnect: Providing Security Situational Awareness by Visualizing Network Traffic Flows. *Workshop on Information Assurance (WIA 04) held in conjunction with IPCCC 2004*, 2004.

[29] X. Yin, W. Yurcik, M. Treaster, Y. Li, K. Lakkaraju. VisFlowConnect: NetFlow Visualization of Link Relationships for Security Situational Awareness. *VizSEC/DMSEC*, 2004.

[30] W. Yurcik, J. Barlow, K. Lakkaraju, M. Haberman. Two Visual Computer Network Security Monitoring Tools Incorporating Operator Interface Requirements. *ACM CHI Workshop on Human-Computer Interaction and Security Systems (HCISEC)*, 2003.