# Visualizing NetFlows for Security at Line Speed: The SIFT Tool Suite

*William Yurcik* – National Center for Supercomputing Applications (NCSA)

## ABSTRACT

The first step in improving Internet security is measurement – security events must be made visible. The irony in making this happen is that there is no lack of security measurement data, in fact, quite the opposite. However, making security manifest faces a major challenge: the large volume and multi-dimensional nature of security data typically obscures valuable security events. NCSA has developed a suite of tools that solves this problem and is making this software available to the Internet community.

We present two visualization tools,[1] (1) NVisionIP and (2) VisFlowConnect-IP. Both of these tools have been developed based on system administrator requirements, their design peer-reviewed in security research forums, and usability testing is in process. These tools both present large volume complex data transparently to system administrators in simple intuitive visual interfaces that support human cognitive processes. NVisionIP visually represents the state of all IP addresses on large networks on a single screen window (we use a Class B address space as the default) with capabilities to filter and drill-down to subnets and individual machines for details-on-demand. VisFlowConnect-IP visually represents flows between internal network IP hosts and the Internet showing who is connecting with whom with capabilities to filter and drill-down to subnets and individual machines for details-on-demand. NVisionIP and VisFlowConnect-IP can be used individually or in unison for correlating events. This work is distinguished from others in that these are the first Internet security visualization tools to be freely available on the Internet and deployed in large production environments.

## Introduction

Organizations use computer network infrastructures that hold a vast amount of information for system administrators and security engineers. There are typical logs common to most computer networks, but the systems are often large and dynamic, making it difficult to extract knowledge from the sea of information. Individually, each system log can be massive, causing operator overload. When overload occurs, security events can slide by unnoticed. Overload can also cause operators to disregard alarms due to high false positive rates. Even in homogeneous infrastructures, solutions from a single vendor fail to scale to medium or large networks. However, the problem is compounded because most organizations have network infrastructures from multiple vendors.

We have developed Security Incident Fusion Tools (SIFT) [9], an integrated suite of tools for evaluating the security of an entire computer network on a single screen. We address the need to discover security incidents that currently go undetected by security operations systems. Specifically two SIFT tools, (1) NVisionIP and (2) VisFlowConnect-IP, leverage human visual cognitive abilities to process log data into knowledge for situational awareness of network security. It is estimated that human beings can visually process a screen of information at 150 Mbits per second [10], with the ability to discriminate relatively minor shifts in color, shape, and motion. By presenting network data visually, it can be scanned quickly, patterns in complex data rise to the surface, and inferences become intuitive. Once a security professional becomes familiar with the normal appearance of the network being monitored, it is much easier to spot attacks including new so-called ‘‘zero-day attacks.’’ The tools are designed to give security engineers situational awareness of an entire network in order to help them determine when a network is under attack, what is being attacked, and what form the attack is taking.

The remainder of the paper is organized as follows: The next section discusses NetFlows data management and introduces the first tool in the SIFT suite: CANINE. Subsequently we present the two SIFT visualization tools – NVisionIP and VisFlowConnect-IPand close with a summary and on-going future work.

## Data Management

### NetFlows Source Data

While this paper focuses on visualization, we would be remiss if we did not address data management since it is arguably the greatest obstacle in realizing any scalable visualization system. We address the challenge of processing high-bandwidth data streams by instrumenting networks with distributed NetFlows

[1]Funded in part by grants from the Office of Naval Research (ONR) under the auspices of the Technology Research, Education, and Commercialization Center (TRECC) and the National Center for Advanced Secure Systems Research (NCASSR) both established at NCSA/University of Illinois.

sensors and then combining this sensor data into a unified format. While in the recent past NetFlows were solely router-based, PC-based NetFlow sensors (Argus) make this a feasible solution for most organizations. The first tool in our suite is a NetFlows converter/anonymizer called CANINE which can handle different NetFlows formats so independent implementations can be interoperable with SIFT visualization tools. NetFlow logs have proven to be the appropriate granularity to process heavily loaded networks and high bandwidth connections (Gb/s) in near-real-time (five minute monitoring windows).

A *network flow* is defined as a sequence of packets that are transferred between two endpoints within a certain time interval. The endpoints are identified at the network layer by IP addresses and at the transport layer by port numbers. In addition to data format differences, there are other interoperability problems in practical NetFlows implementations:

- Cisco NetFlows are defined as *unidirectional* and generated through intelligent flow cache management, which contains a set of specialized algorithms [4].
- Argus NetFlows are defined as *bidirectional* containing two distinct sub-flows, one in each direction [2].
- Cisco and Argus NetFlow formats have different fields (e.g., flags etc.) [3, 5].

For a more detailed comparison between different NetFlows formats see [15].

### CANINE

With the increased use of NetFlows for security monitoring and the fact that NetFlows come in different and incompatible formats, we have developed CANINE (Converter and ANonymizer for Investigating Netflow Events) [7, 8] which can be downloaded from http://security.ncsa.uiuc.edu/distribution/CanineDownLoad.html. CANINE allows tools designed for a specific type of NetFlows to be interoperable with any NetFlow format. CANINE consists of the two main modules: (1) the CANINE GUI and (2) the conversion/anonymization engines. For the purposes of this paper we will only discuss the conversion engine (for information about the anonymization engine see [7, 8]). The CANINE GUI accepts user input to identify the NetFlow file for conversion, sends the request to the processing engine which performs the conversion to the newly specified output file, and lastly summarizes the results of the performed actions in a pop-up window. At present CANINE supports conversion to/from Cisco version 5/7, Argus, NFdump, and our own NCSA internal NetFlows format. Future formats to be included in CANINE include Cisco version 9 and the future IETF IPFIX standard.

### Network Instrumentation

With the development of high-speed network infrastructure has also come the need for high-speed security – security at line speed – for current 2005 networks this is 4 GB/s at the edge and higher within the core [15]. Unfortunately, high network bandwidths present special problems for security monitoring.

The first challenge is the streaming nature of security sensors. It is important to note that security sensors generate streaming data and not batch log files. Since streaming analysis is an open research question, security systems typically create batch log files by collecting streaming data over defined time periods. However, depending on the network size and traffic volume these log files can become large and difficult to handle. Tuning is required to determine the best time period of analysis to match the preferred log size to the network size and traffic volume. Creating logs over longer time intervals may risk losing NetFlows records upon high transmission rates from overflow or blocking.

The second challenge is observation point. Security cannot be measured where it is not observed thus sensors need to be placed to cover the entire network space. Typical deployment for NetFlows includes the border router for Internet traffic and Argus sensors for internal network observation. There are blind spots from VLANs and switched networks which do not leave IP (network layer) traces – future sensors based on S-Flows are developing to address this gap.

The third challenge is CPU speed to generate and process NetFlows at line speed. As routers have increased speed, monitoring techniques have shifted to sampling NetFlows. While sampling is statistically sufficient for network planning, it is not a good idea for security analysis. NetFlow records are created by sampling packets (not flows), letting the majority of the packets go unnoticed, which may lead to missing important security events. A possible justification for sampling is that an attack may be high traffic volume, at least part of which may be captured with high probability (such as high-volume denial-of-service attack or indiscriminate scanning by propagating worms and viruses). A preferred approach we recommend for security at line speed is the parallel processing NetFlows in a distributed manner. Instead of instrumenting only the high-speed border router that may only be able generate sampled NetFlows, instead instrument all the routers feeding into the border router. This technique effectively relieves the load on each flow collector so that it will not be over subscribed. The drawback is that multiple flow collectors are required and NetFlows records from different routers must be merged to eliminate duplicate flows (the same flow that passes through multiple routers).

## NetFlows Visualization Tools

### Design By Requirements

We firmly believe that the first step to improve Internet security is by measurement. Measurement

allows one to accurately assess the degree of the problem at a specified time and then further measurements track whether solutions are having the desired effect. However, not all measurements are equal, users have a mental model based on experience and tools should be designed to enhance and augment these mental models for the most effective results [16, 17, 18].

For this work, we did two important things often neglected from security tool design: (1) taking time to work with security engineers in their operational production environment in order to learn their mental models and thus tool requirements and (2) the capability to design new visualization models from scratch to meet these requirements without having to incorporate legacy constructs. The results have been very satisfying in that most security engineers who view our visualization tools for the first time immediately begin inferring hypotheses based on the content displayed.

To briefly summarize the major findings from our requirements analysis there are two primary findings. First, security engineers need to answer questions such as these posed by upper management: What is the state of the network? Is the network being attacked? How is the network being attacked? Who is attacking the network? While these may appear to be basic questions, the answers are not immediately available using current security tools and when available after much analysis the answers are complex. Visualization provides a rich representation to help answer these questions concisely.

Second, security engineers have mental models based on their experience with the network infrastructure, knowledge of people within the organization, and security expertise learned over many years. While most tenets of information visualization design are useful in designing within our specific security domain, we did find that leveraging the mental model of security engineers caused us to break some of these consensus rules (after much consternation). Instances when the security engineer mental model overrides information visualization design best practices are highlighted in our discussion of each of our visualization tools.

**NVisionIP**

Our first and most mature security visualization tool is NVisionIP [1, 6] which we designed to answer the question: What is the state of the network?

Figure 1 shows the Galaxy view of NVisionIP which can be downloaded from http://security.ncsa.uiuc.edu/distribution/NVisionIPDownLoad.html . The Galaxy view represents an entire Class B IP address space (in this single window!) as a matrix with subnets along the horizontal axis and hosts along the vertical axis. Each IP address is represented as a dot (actually four pixels) and the state of each IP address is represented with color or shape as determined by the user in the color and shape legend. Two magnification options are available to see the IP addresses: linear and fisheye.

NVisionIP allows the security engineer is to load one (or multiple) NetFlow files and perform visual queries. NVisionIP has taken all the possible NetFlow database query combinations and hard coded them into the tool as drop-down and point-and-click commands. A user would typically start with primary queries such as how bytes per IP address or how many connections per IP address. A filter then allows the user to select secondary queries to view only source or destination traffic, different protocol (IP, UDP), and different ports (destination or source ports, specific ports or collections of ports) or any combination thereof.

At the Galaxy view, NVisionIP can identify large or small levels of traffic as measured in bytes (based on expectations for the class of machine – laptop or server). This may indicate malware is being served to/from a machine or the machine is involved in a denial-of-service event. Worm and virus scans as indicated by number of connections can also be easily detected based on variance from expected levels.

There are aspects of the Galaxy view design that are contrary to information visualization best practices: the IP address space is laid out logically in matrix space without organizing IP addresses into known classes or enlarging the part of the IP address space with more activity (thus patches of white space or inactive IP address space appears). This design was intentional to retain security engineer knowledge of the IP address space based on logical numbering for subnets/hosts and mental mapping between logical addresses (e.g., cluster compute nodes with contiguous IP addresses) and physical locations (IP subnets are usually physically located in the same area such as a building floor etc.). The white space of inactive IP addresses actually has other advantages and is not wasted space – any traffic activity shown there is anomalous (unallocated address space that should have no legitimate traffic).

While an overall view is important, it is of limited use without the ability to drill down to find more detailed information when something interesting is identified. Figure 2 shows the drill-down levels of NVisionIP which are activated with a mouse click and a drag over a region of interest. These levels are the (1) Small Multiple View and (2) Machine View.

The Small Multiple View allows the user to quickly scan and compare traffic activity across subnets on many machines simultaneously. Each machine is a box with two sets of histograms, an upper set of histograms representing traffic on well-known ports and a lower set of histograms representing traffic on ports over 1024. The well-known ports are color-coded in a user legend. The ports over 1024 are ordered from most active to least active (top N ports). Note that no numbers are shown in the small multiple view, this view is designed for the user to identify activity of interest and then drill-down for raw data details on-demand.

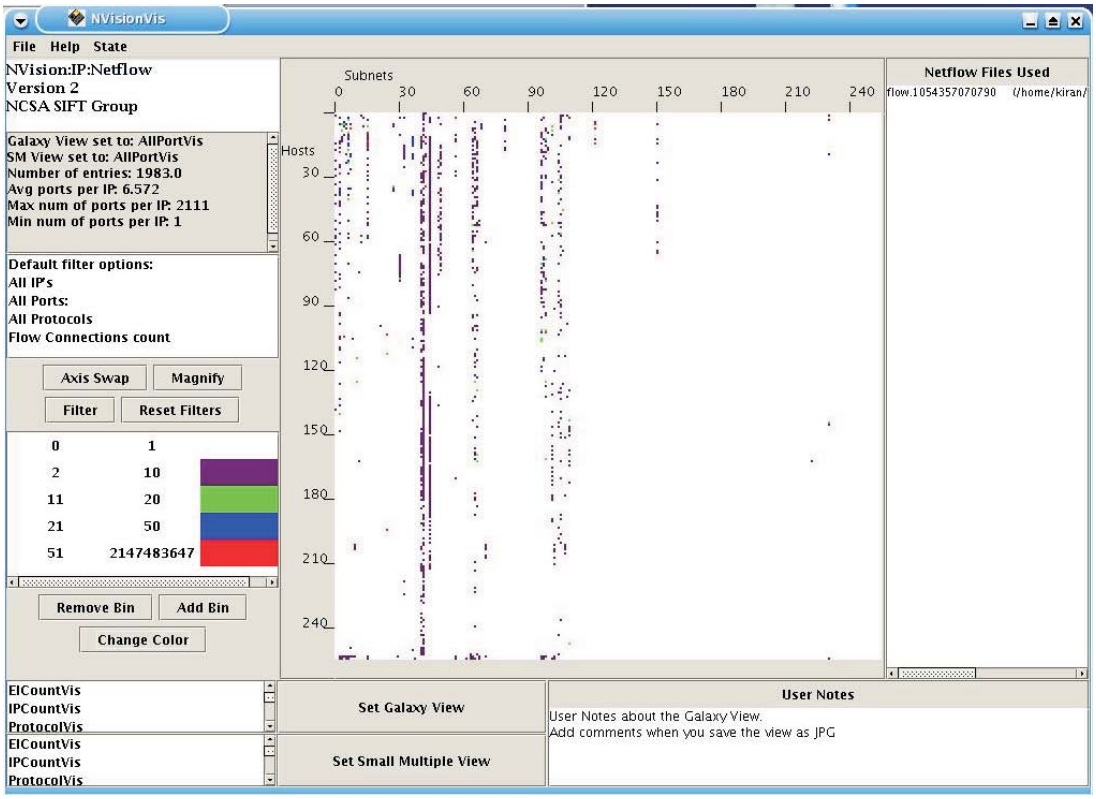


**Figure 1**: NVisionIP Galaxy view of an entire Class B IP address space.

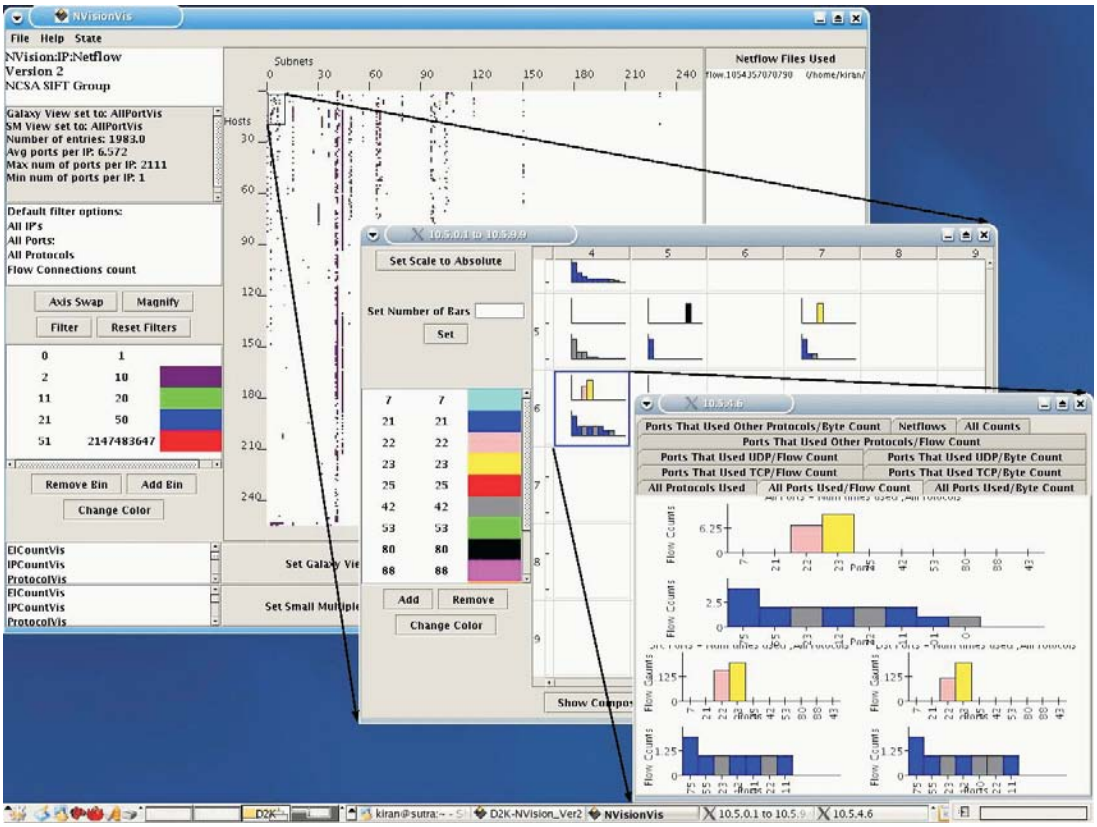


**Figure 2**: The three levels of NVisionIP (top to bottom): (1) Machine view, (2) Small multiple view, and (3) Galaxy view.

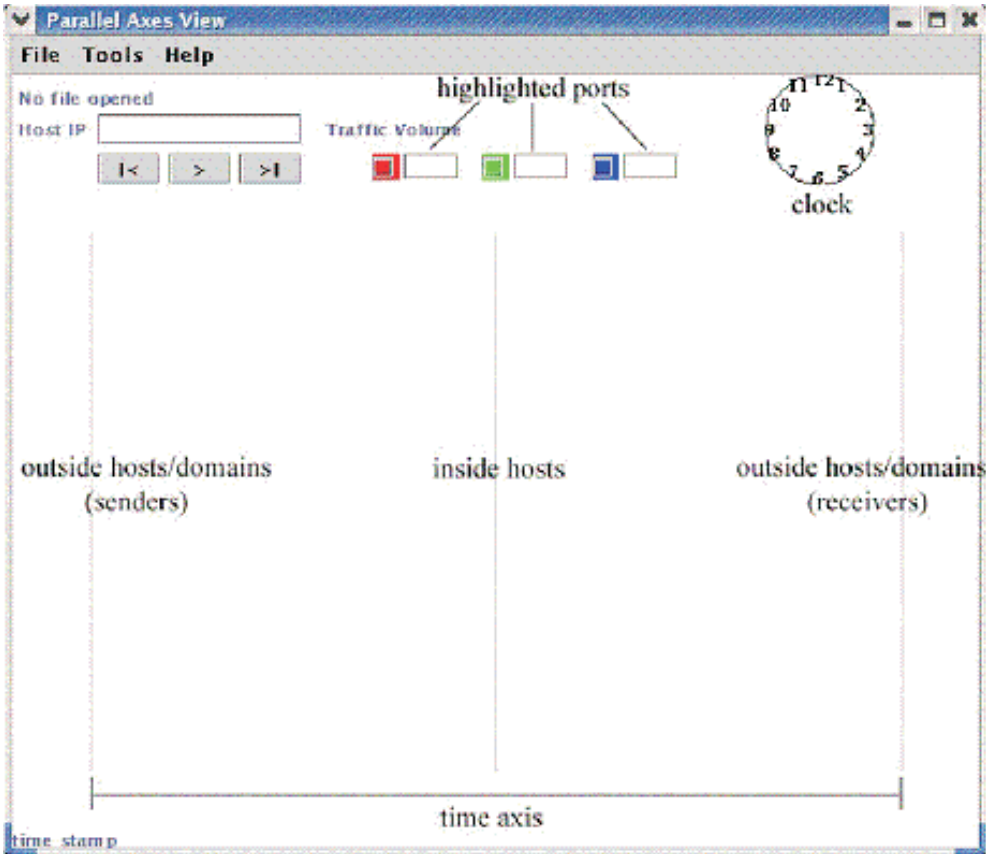


**Figure 3**: VisFlowConnect-IP: Main view.

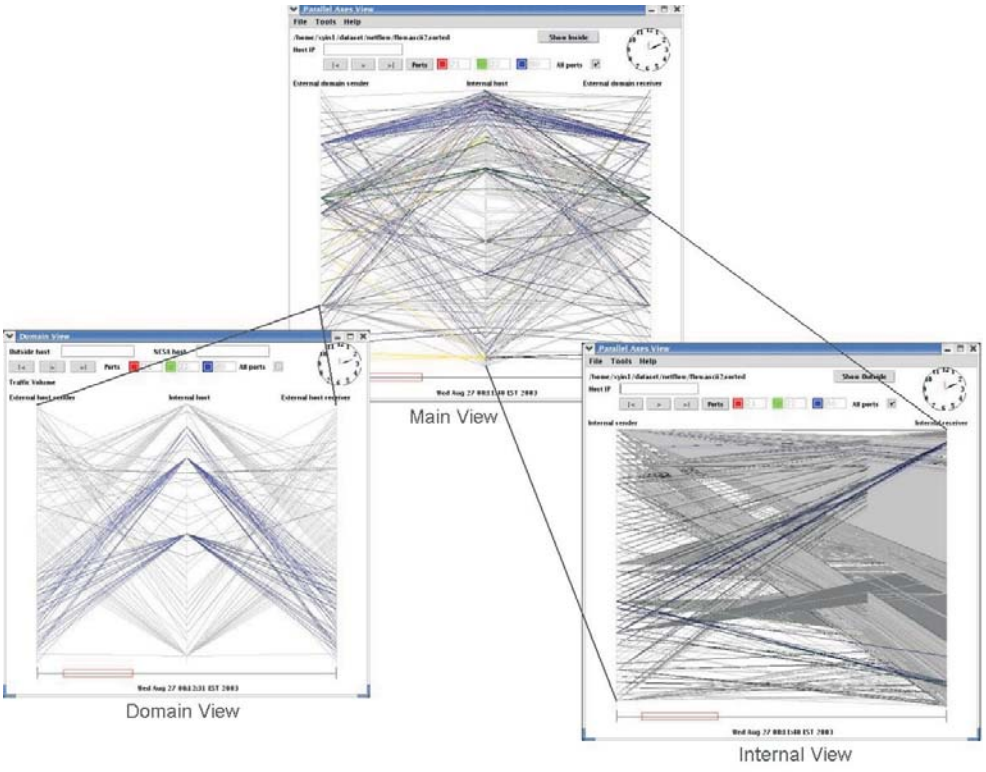


**Figure 4**: Drill-down layers of VisFlowConnect-IP: left Domain View and Bright Internal View.

At the Small Multiple View, NVisionIP has been used to quickly identify anomalous services that violate security policy such as unauthorized services (web or mail server) and exposed services that need to be patched or otherwise protected due to vulnerabilities.

If a user sees a machine with interesting traffic activity in the Small Multiple View, they may click on that block to drill-down to the Machine View. The Machine View organizes all the data from a particular machine in multiple tabs – each tab showing aggregate activity in an upper histogram and source/destination directional flows in two lower histograms. Note in the Machine View each histogram is fully labeled with port numbers and traffic level. At the lowest level, the raw NetFlows data for that machine is available for inspection in this Machine View as a tab. At this lowest level Machine View, details of most security events can be revealed.

The overall effect of using NVisionIP (with its interactive drill-down levels) is that relationships between aggregate network activity and individual machine activity can be more easily discovered and comprehended by human operators – providing situational awareness of network system state. Future work is progressing on optimizing Galaxy View animation to show IP address state changes over time as well as a difference view to visually compare current network traffic versus benchmark network traffic.

**VisFlowConnect-IP**

The second security visualization tool in the SIFT tool suite focuses on answering the question: Who is connecting to whom on the network? This basic question has been attempted in the past with topology-based diagrams based on network infrastructure, however, these results were either (1) not providing information relevant to real-time traffic or (2) not scalable since representing network traffic over time produces obscured lines in relatively short order. We solve both of these problems with VisFlowConnect-IP [12, 13, 14] which is available at http://security.ncsa.uiuc.edu/distribution/VisFlowConnectDownLoad.html .

VisFlowConnect-IP is a security visualization tool based on the parallel axes concept drawn from data mining. It is a complementary tool to NVisionIP since it visualizes the same NetFlows source data – the design similarities will become apparent in the following description. VisFlowConnect-IP allows a user to visually assess the connectivity of large and complex networks (in a single window!) by providing a main view of the network with filter and drill-down views that provide more details on-demand. The three views of VisFlowConnect-IP are: (1) Main, (2) Domain, and (3) Internal. The Main View is shown in Figure 3 with the Domain and Internal views shown in Figure 4.

The VisFlowConnect-IP Main View utilizes the parallel axis view with the left-most and right-most vertical axes representing the external domains and the center vertical axis representing host IP addresses within the internal edge network domain (See Figure 3). Lines connecting external domains and internal hosts represent directional data flows, with line darkness being proportional to the logarithm of the volume of data transferred. VisFlowConnect-IP can filter/highlight flows to certain hosts or traffic on specific ports and protocols using a filter drop-down menu and selection boxes on the main view. Ports indicated in the selection boxes are represented in different colors within the network traffic or may be isolated from network traffic for focused analysis. The overall effect is visualization of traffic into-an-edge-network-from-the-Internet and traffic out-from-an-edge-network-to-the-Internet.

Figure 4 shows the two drill-down views within VisFlowConnect-IP. While we would have liked to represent each individual external host IP address connecting into the internal edge network symmetrically on both the left-most and right-most axes, this is not possible due to scalability. Preliminary measurements of NCSA's network showed over 100,000 different IP addresses commonly appeared in the NetFlow files we wished to visualize and this is too many for the vertical line pixel space of a single window without scrolling. Instead we implemented a drill-down Domain View which is invoked by the user clicking on a drop-down menu while having an external domain highlighted on the vertical external domain axis. The resulting Domain View is a mirror image of the Main View except it only shows traffic within the highlighted external network domain to/from the internal edge network. This has turned out to be very valuable since typically hackers ''own'' entire subnets or even ''own'' entire network domains so it is common to see malicious activity captured within a Domain View.

Figure 4 also shows the drill-down Internal View which is invoked as a toggle button on the Main View. While monitoring for external Internet hacker activity is sexy, we have found this Internal View very useful since it shows only traffic that both sources and sinks within the internal edge network. There are only two vertical lines in this view, internal edge network IP addresses are ordered symmetrically in a mirror image on the left-most and right-most axes (no middle axis). This Internal View has helped security engineers determine important security events like the initial source of a worm infection which infiltrated the edge network from the inside, and the insider attacks from those misusing privileged access.

The VisFlowConnect-IP Main View has a time axis at the bottom which is used to solve the scalability problem we referred to as the major challenge for this tool. The user loads a NetFlow file for visualization and then may select multiple filters to determine how this traffic is to be represented in animation including intensity, byte size, and a sliding time window. The sliding time window provides scalability by only representing traffic within the window and ignoring traffic outside the window. Thus the sliding time window can be adjusted to any size network and any

traffic volume – the general rule for clear viewing is the more traffic the smaller the sliding time window. The window size itself is represented to the user by a red box (where the length of the red box is proportional to window size) that travels along the time axis as the traffic is animated (as shown in the Domain and Internal Views within Figure 4).

VisFlowConnect-IP has also implemented a filter language using real expressions that is beyond the scope of this paper [12]. With this filter language capability, VisFlowConnect-IP can create mechanisms for storing/retrieving filter profiles. These profiles can store customized filters that remove ''uninteresting'' information from view-thus leaving only the more security relevant data to be displayed.

## Summary

Visualization is the future of security monitoring and NetFlows are the source data for high-speed networks. In this paper we marry security visualization with NetFlows by presenting the SIFT suite of tools along with accompanying techniques for security at line speed. The goal is to enable security engineers to go beyond binary/text command line log file analysis toward real-time network security situational awareness. A growing community of researchers has formed on security visualization, see [11] for more information.

The three specific tools of the SIFT suite presented in this paper (CANINE, NVisionIP, and VisFlowConnect-IP) are available for download at the URLs provided in the text. We are currently conducting usability tests with human subjects to quantify the utility of these tools and preliminary results from these tests are very promising. We intend to go open source with these tools after the software is stable, at present we are still developing the software with new versions posted on the corresponding webpages. We enthusiastically invite feedback from users about the use of these tools.

## Author Biography

William (Bill) Yurcik is currently Manager, Security R&D and Senior Systems Security Engineer at NCSA. Prior to this he was Head of Security Operations at NCSA, so he has both a theoretical and practical background in computer network security. Prior to joining NCSA he has 12 years of professional experience as a Network Engineer for large networks (Naval Research Laboratory, NASA, Verizon, and MITRE). He is a graduate of Johns Hopkins University (MS Electrical Engineering 1990, MS Computer Science 1987), the University of Maryland (BS Electrical Engineering 1984), and is Ph.D. ABD from the University of Pittsburgh (1994-99). Bill can be reached at byurcik@ncsa.uiuc.edu .

## References

[1] Bearavolu, Ratna, Kiran Lakkaraju, and William Yurcik, ''NVisionIP: An Animated State Analysis Tool for Visualizing NetFlows,'' *FLOCON,* 2005.

[2] Bullard, Carter, *Argus, the network Audit Record Generation and Utilization System*, http://www.qosient.com/argus/ , accessed 26 September, 2005.

[3] Bullard, Carter, *Argus Record Format*, http://www.qosient.com/argus/argus.5.htm/ , accessed 26 September, 2005.

[4] Cisco Systems, *Cisco NetFlow Services and Applications White Paper*, http://www.cisco.com/warp/public/cc/pd/iosw/ioft/neflct/tech/napps_wp.htm , accessed 26 September, 2005.

[5] Cisco Systems, *NetFlow Overview Presentation,* http://www.cisco.com/application/vnd.mspowerpoint/en/us/guest/tech/tk362/c1482/ccmigration_09186a0080182b50.ppt , accessed 26 September, 2005.

[6] Lakkaraju, Kiran, William Yurcik, Adam J. Lee, Ratna Bearavolu, Yifan Li, and Xiaoxin Yin, ''NVisionIP: NetFlow Visualizations of System State for Security Situational Awareness,'' *CCS Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC)*, 2004.

[7] Li, Yifan, Adam Slagell, Katherine Luo, and William Yurcik, ''CANINE: A Combined Converter and Anonymizer Tool for Processing NetFlows for Security,'' *International Conference on Telecommunication Systems – Modeling and Analysis (ICTSM)*, 2005.

[8] Luo, Katherine, Yifan Li, Adam Slagell, and William Yurcik, ''CANINE: A NetFlows Converter/Anonymizer Tool for Format Interoperability and Secure Sharing,'' *FLOCON*, 2005.

[9] *SIFT Project Webpage*, http://www.ncassr.org/projects/sift/ , accessed 26 September, 2005.

[10] Tufte, Edward, *A One-Day Course: Presenting Data and Information*, Madision WI, (http://www.edwardtufte.com/tufte/courses , accessed 26 September, 2005), August, 2005.

[11] *VizSEC Community Homepage*, http://www.ncassr.org/projects/sift/vizsec/ , accessed 26 September, 2005.

[12] Yin, Xiaoxin, William Yurcik, and Adam Slagell, ''VisFlowConnect-IP: An Animated Link Analysis Tool for Visualizing NetFlows,'' *FLOCON*, 2005.

[13] Yin, Xiaoxin, William Yurcik, and Adam Slagell, ''The Design of VisFlowConnect-IP: a Link Analysis System for IP Security Situational Awareness,'' *Third IEEE International Workshop on Information Assurance (IWIA)*, 2005.

[14] Yin, Xiaoxin, William Yurcik, Michael Treaster, Yifan Li, and Kiran Lakkaraju, ''VisFlowConnect: NetFlow Visualizations of Link Relationships for Security Situational Awareness,'' *CCS Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC)*, 2004.

[15] Yurcik, William and Yifan Li, ‘‘Case Study: Instrumenting a Network for NetFlow Security Visualization Tools,’’ *21st Annual Computer Security Applications Conference (ACSAC)*, 2005.

[16] Yurcik, William, Kiran Lakkaraju, James Barlow, and Jeff Rosendale, ‘‘A Prototype Tool for Visual Data Mining of Network Traffic for Intrusion Detection,’’ *Workshop on Data Mining for Computer Security (DMSEC)*, 2003.

[17] Yurcik, William, James Barlow, and Jeff Rosendale, ‘‘Maintaining Perspective on Who Is The Enemy in the Security Systems Administration of Computer Networks,’’ *ACM CHI Workshop on System Administrators Are Users, Too: Designing Workspaces for Managing Internet-Scale Systems*, 2003.

[18] Yurcik, William, James Barlow, Kiran Lakkaraju, and Mike Haberman, ‘‘Two Visual Computer Network Security Monitoring Tools Incorporating Operator Interface Requirements,’’ *ACM CHI Workshop on Human-Computer Interaction and Security Systems (HCISEC)*, 2003.