VisFlowConnect-IP: An Animated Link Analysis Tool For Visualizing Netflows [*]

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

We present VisFlowConnect-IP, a network flow visualization tool that allows operators to detect and investigate anomalous internal and external network traffic. We model the network on a parallel axes graph with hosts as nodes and traffic flows as lines connecting these nodes. We present an overview of this tool's purpose, as well as a detailed description of its functions.

1   Introduction

Networks are becoming increasingly complex, and the number of different applications running over them is growing proportionally. No longer can a system/network administrator realisitically be aware of every application on every machine under her control. At the same time, the number of network attacks against machines has increased exponentially. These attacks are often concealed among this vast amount of legitimate, and seemingly random, traffic. It is often difficult just to log this traffic, yet alone analyze and detect attacks in real-time with traditional text-based tools.

However, humans excel at processing visual data and identifying abnormal patterns. Visualization tools can translate the myriads of network logs into animations that capture the patterns of network traffic in a succinct way, thus enabling users to quickly identify abnormal patterns that warrant closer examination. Such tools enable network administrators to sift through gigabytes of daily network traffic more effectively than scouring text-based logs.

VisFlowConnect-IP is one such network visualization tool. It visualizes network traffic as a parallel axes graph with hosts as nodes and traffic flows as lines connecting these nodes. These graphs can then be animated over time to reveal trends. VisFlowConnect-IP has the following distinguishing features: (1) it uses animations to visualize network traffic, so that network dynamics can be presented to users in a comprehensible and efficient manner, (2) it pro-

vides both an overview of traffic as well as drill-down views that allow users to dig out detailed information, and (3) it provides filtering capabilities that enables users to remove mundane traffic details from the visualization.

2   System Architecture

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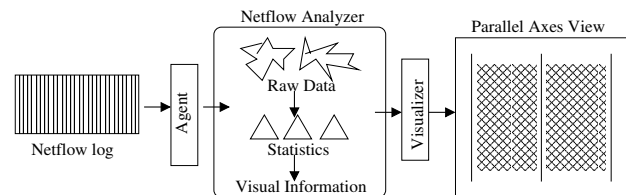

Figure 1. System Overview

2.1   NetFlow Source Data

VisFlowConnect-IP can use the following NetFlow formats: Cisco 5/7 and Arugs[1]. VisFlowConnect-IP works in a batch mode, reading NetFlow records from a log. An agent is used to extract the NetFlow records and feed them into VisFlowConnect-IP. Each record contains the following information: (1) sourcce/destination IP addresses and ports, (2) number of bytes and packets, (3) start and end timestamps, and (4) protocol type.

2.2   Input Filtering Capability

NetFlow logs contain many different types of traffic with distinct properties. While certain traffic patterns are usually a red flag, depending upon the context, they may be quite normal and benign. For example, it is very common that a DNS server has connections with every other

[1]http://www.qosient.com/argus/

host on a network, but on a workstation this may indicate a worm infection. In order to remove noise such as this, VisFlowConnect-IP provides advanced filtering profiles that users can store and load.

Let $F_1, \ldots, F_k$ be a set of user created filters. Table 1 shows filter variables and their value ranges. Each filter has a list of constraints on the variables and a leading label ("+" or "–") that indicates whether to "include" or "exclude" matches. A constraint on a variable takes the form of "$x = v_{min} - v_{max}$", where "x" is a variable and "v_{min}" and "v_{max}" are the lower and upper bounds of "x", and "$=$" is the only operator defined. Records are passed sequentially through each filter and the last match will determine whether or not to include the record. A record that matches no filter rules is dropped. For example, the following set of filters will include all traffic from domain 141.142.x.x with a source port between 1 and 1000, except tcp traffic involving port 80.

+: (SrcIP=141.142.0.0-141.142.255.255), (SrcPort=1-1000)

-: (SrcPort=80, Protocol=tcp)

-: (DstPort=80, Protocol=tcp)

Variables	Value Ranges
SrcIP, DstIP	$0.0.0.0 \Leftrightarrow 255.255.255.255$
SrcPort, DstPort	$0 \Leftrightarrow 65535$
Protocol	tcp, udp, icmp
PacketSize	$0 \Leftrightarrow \infty$

Table 1. Input Filter Language

3   How to Use VisFlowConnect-IP

In this section, we describe the visualize interface of VisFlowConnect-IP—which can be downloaded at <http://security.ncsa.uiuc.edu/distribution/VisFlowConnectDownLoad.html>

In *the Parallel Axes View*, three vertical axes are used to indicate traffic between external domains and internal hosts on the center axis (Figure 3). Points on the left [right] axis represent external domains that are sourcing [receiving] flows to [from] the internal network. Unlike the middle axis where points represent individual hosts, here points represents sets of hosts. The darkness of a line between two points is proportional to the logarithm of traffic volume between the hosts. All points are sorted according to their IP addresses, so that each point will remain at a relatively stable position for a user to track during animation. Figure 2 illustrates the VisFlowConnect-IP GUI with important features labeled.

1. **Menu Bar:** It contains the menu items for operations that are less frequently used, including (1) 'Open': open a NetFlow file, (2)'Load Filters': load a file for input filters, (3) 'Settings': bring up the settings dialog box, (4) 'Show Domain': show the domain view of the selected domain (described below), (5) 'Host
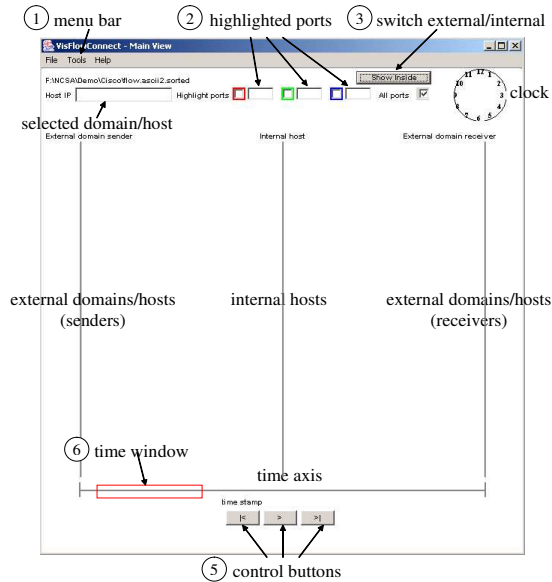

Figure 2. Parallel Axes View

Statistics': show the traffic statistics of the selected host/domain, and (6) 'Save Screen': save a snapshot of the current view.

2. **Highlighted Ports:** The user may specify up to three ports to highlight in special colors: red, green, or blue (see Figure 3). The user may also click on the check box to show traffic only on the highlighted port.

3. **External/Internal Switch:** The internal view (See figure 4) shows traffic between hosts on the internal network. The points on the left [right] axis represent the source [destination] of traffic flows. The user may switch between external and internal views by clicking on the button "Show Inside/Outside".

4. **Domain View:** As shown in Figure 5, VisFlowConnect-IP has a drill-down Domain View that allows a user to visualize traffic between hosts in a specific external network domain to/from hosts in the internal network. The Domain View shows all traffic between individual hosts in the corresponding external network domain and the internal network.

5. **Control Buttons:** A user can control the animation with three buttons: ($|<$) rewind back to start, ($>$) play forward a defined time unit (default is 10 minutes), and ($>|$) play forward to the end of the data set.

6. **Time Window:** Because a user will typically be more interested in recent traffic, only flows within a specified time window are shown as opposed to a cumulative view. A sliding rectangle along a horizontal time axis is shown at the bottom of the GUI to indicate the time window in view.

7. **Settings Dialog:** Figure 6 shows the settings dialog, which allows the user to change the input file format

(Cisco or Argus) and to select protocols of interest (e.g., tcp, udp or icmp). Here, the user may also adjust the traffic threshold, so that only domains whose aggreagate traffic volume is lower than this threshold are ignored. It also allows the user to change the time window, to restrict investigation to flows whose sizes are within a user-defined range, and to ignore flows over certain ports. This is also where the user sets the "Local IP Range" in order to distinguish internal hosts from external hosts.
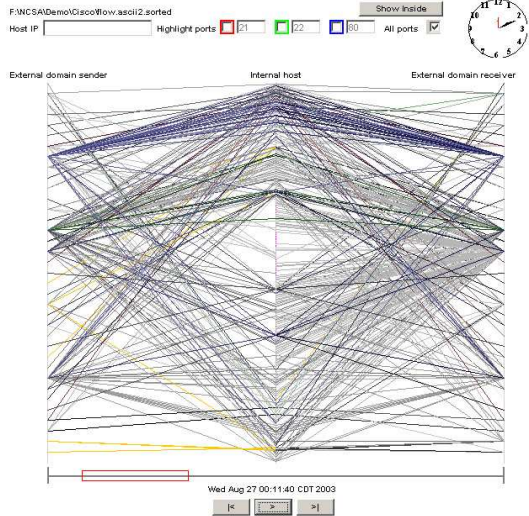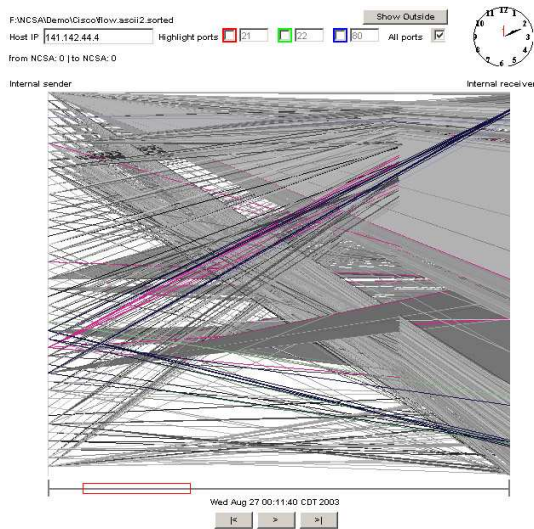

Figure 3. External View

Figure 4. Internal View

4   Related Work

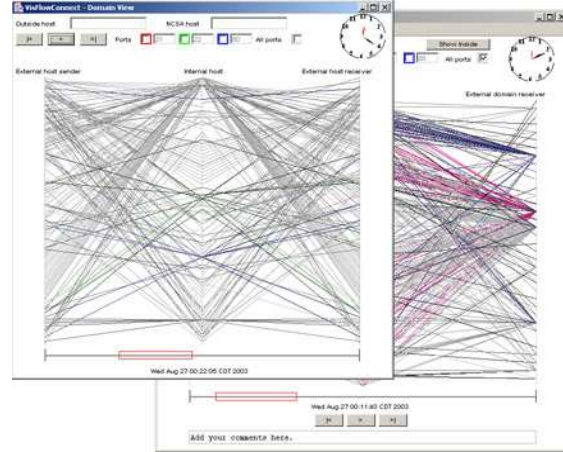In [5] the authors present a tool named NVT (Network Vulnerability Tool) that visually depicts a network topology
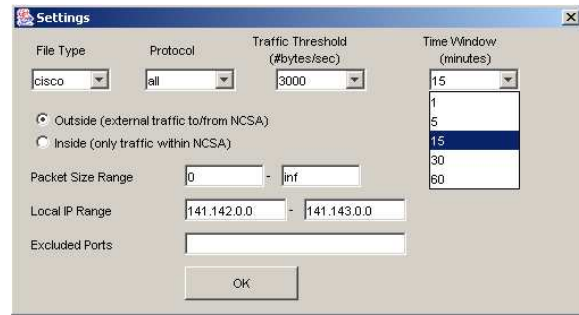

Figure 5. Domain View

Figure 6. Setting Dialog

and generates a vulnerability database. In [6], the authors present a visualization of network routing information that can be used to detect inter-domain routing attacks and routing misconfigurations. In [7], they go further and propose different ways of visualizing routing data in order to detect intrusions. An approach for comprehensively visualizing computer network security is presented in [4], where Erbacher et al. visualize the overall behavioral characteristics of users for intrusion detection. [1] focuses on visualizing log data from a web server in order to identify find patterns of malicious activity caused by worms.

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 [2, 3, 8]. [2] and [8] focus on visualizing linkages in a network, and [3] focuses on detecting attacks based on fingerprints. Link analysis can illustrate interactions between different hosts either inside or outside a network system, thus providing abundant information for detecting intrusions. In previous papers we have introduced the design and implementation of VisFlowConnect-IP [9, 10, 11, 12], an animated tool for visualizing network flows. This paper describes how to use that tool in detail.

5 Example Anomaly Detection

Here, we show an example of how we can detect the blaster virus with VisFlowConnect-IP. The blaster virus spreads quickly and has a common worm characteristic in which infected computers send out packets to an abnormally large number of hosts within a short time period. In Figure 7, one can see that there is one domain which connects to almost every host in the local network. This indicates that some hosts in that domain might be infected by a worm. This is verified when we see the uniform payload size and port usage on all of these flows that match the Blaster signature. At this point we can filter on those characteristics, and by digging deeper with the domain view, we can begin to identify specific hosts that have been infected.
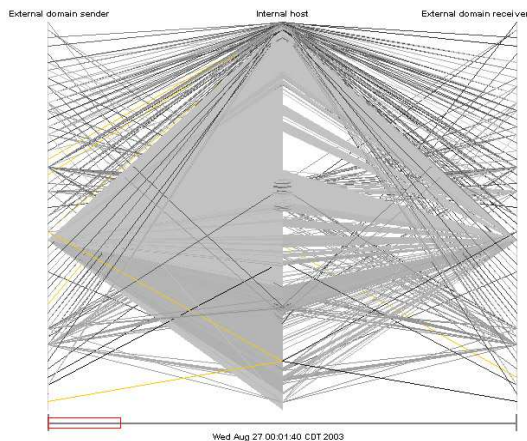

Figure 7. External view of blaster attacks

6 Conclusions

We have presented VisFlowConnect-IP, an approach to visualizing patterns on a network with NetFlow log data. Its purpose is to enhance an administrator's situational awareness by providing an easy-to-use, intuitive view of NetFlow data using link analysis. The central aspect of this interface is the parallel axes view, used to represent the origin and destination of network traffic. A high-level overview of the data is provided first, and the user is provided the capability of drilling down into the data to find additional details. Filtering mechanisms are provided in order to assist the user in extracting interesting or important traffic patterns. The VisFlowConnect visualization framework described in this paper is extensible beyond IP networks, and we are currently modifying it to monitor security in storage systems and high performance cluster computing environments as well.

References

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

[2] 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.

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

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

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

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

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

[8] 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.

[9] X. Yin, W. Yurcik, Y. Li, K. Lakkaraju, C. Abad. VisFlowConnect: Providing Security Situational Awareness by Visualizing Network Traffic Flows. *23rd IEEE Int'l. Performance Computing and Communications Conference (IPCCC)*, 2004.

[10] X. Yin, W. Yurcik, A. Slagell. The Design of VisFlowConnect-IP: a Link Analysis System for IP Security Situational Awareness. *3rd IEEE Int'l. Workshop on Information Assurance (IWIA)*, 2005.

[11] X. Yin, W. Yurcik, M. Treaster, Y. Li, K. Lakkaraju. VisFlowConnect: NetFlow Visualization of Link Relationships for Security Situational Awareness. *CCS Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC)*, 2004.

[12] W. Yurcik. The Design of an Imaging Application for Computer Network Security Based on Visual Information Processing. *SPIE Defense and Security Symposium / Visual Information Processing XIII*, 2004.