A Visualization Tool for Exploring Multi-scale Network Traffic Anomalies

Romain Fontugne[1], Toshio Hirotsu[2], Kensuke Fukuda[3]
[1]The Graduate University for Advanced Studies, Tokyo, Japan
[2]Hosei University, Tokyo, Japan
[3]National Institute of Informatics / PRESTO, JST, Tokyo, Japan

***Abstract*— Since anomaly detection in Internet traffic is a crucial and unmet challenge, many anomaly detectors for backbone traffic have recently been proposed. However, evaluating anomaly detectors is a complicated task due to the lack of ground truth data. Our goal is to provide a good level of support for rapidly understanding traffic behaviors and assisting researchers in evaluating the effectiveness of anomaly detectors. This article presents an interactive tool that takes advantage of several graphical representations highlighting the different aspects of network traffic and anomalies. The proposed tool allows for exploration of network traffic at any temporal and/or spatial (address and port) scales. In addition, an accurate description of any sub-traffic is available in the form of textual packet information, enabling complete understanding of the monitored traffic. We exhibit the effectiveness of the proposed tool by analyzing darknet traffic, backbone traffic, and anomalies reported by an anomaly detector. We illustrate a manual validation of the anomalous traffic reported by anomaly detectors, and inspect a recent and sophisticated threat: the Conficker worm. We also state several typical patterns that stand for different kinds of anomalies.**

I. Introduction

The Internet has become a common medium for communication and information exchange providing many attractive services for ordinary users. A victim of its own success, Internet traffic is still growing at a fast rate and contains an increasing amount of anomalies such as misconfigurations, failures, and attacks. These improper uses of network resources consume bandwidth and adversely affect the performances of networks. Thus, these anomalies penalize legitimate applications from using an optimal amount of network resources. Since the core of the Internet is particularly deteriorated by anomalous traffic, quick and accurate detection of anomalies in the backbone traffic has been a hot topic (e.g., [1], [2], [3], [4]). However, due to the lack of ground truth data for backbone traffic, evaluating an anomaly detector is quite challenging and tricky [5]. Therefore, researchers must validate their results from their anomaly detectors by manually investigating the dump files or flow records. This is a baffling problem as it is laborious to identify a few thousand harmful packets from millions of innocuous ones.

Nevertheless, visualizing network traffic is a valuable tool for in investigating dump files. The main advantage of graphical representations is to highlight the significant features of the traffic, thus the main properties of the traffic are understood at a mere glance. Moreover, several degrees of information are retrieved by monitoring the various representations that depict different aggregations of the traffic. For example, a time series is useful for analyzing the time evolution of a single feature for a huge amount of flows. Whereas, a graphlet [6] depicts several features of only a few flows.

In this article, we propose a tool to visualize, explore, and understand network traffic at any temporal and spatial (address and port) scale. Our main contribution is to provide a tool that assists researchers, or network operators, in understanding and validating alarms reported by their anomaly detectors. The proposed tool provides six basic features to help researchers inspect network traffic and evaluate anomaly detectors:

- Network traffic is displayed at different resolutions, and the user is able to zoom in/out along the time axis or address/port space.
- The tool provides different types of scatter plots (corresponding to IP addresses, or port numbers) and time series (e.g., throughput and average packet size). Since these graphical representations are intuitive views, the tool simultaneously displays two views and provide an exhaustive description of the traffic.
- Understanding backbone traffic involves inspecting various sub-traffics, and therefore, the tool allows to easily move along the network traffic in time and space (i.e. address and port number space).
- The tool retrieves all the details concerning the monitored traffic in the form of accurate graphlet and textual data.
- Anomalies identified by anomaly detectors are displayed by this tool, and thus, researchers and network operators are able to easily validate the veracity of the detected anomalies.
- The current implementation runs on different platforms on a daily basis, it uses no intermediate database, and it directly reads dump files (pcap form [7]).

We evaluated the tool on several kinds of traffic; darknet traffic reveals shapes highlighting anomalous traffic, and similar patterns are also observed in the backbone traffic. Furthermore, we demonstrate the help provided by the tool in identifying recent and sophisticated attacks such as the Conficker worm. We also conduct a manual inspection of anomalous traffic reported by anomaly detector, and list several typical patterns highlighting anomalies (in accordance with those reported in [4]).

doi:10.4304/jnw.6.4.577-586

II. Related work

Various visualization tools assist researchers and network operators in monitoring network traffic. For example, Fischer et al. [8] and Goodall et al. [9] presented two interesting tools focusing on anomaly detection. The former [8] monitors traffic related to local hosts based on a TreeMap visualization. It is used to check alarms reported by intrusion detection system (IDS), and to identify large-scale attacks aiming at local hosts. The latter, Time-based Network traffic Visualizer (TNV) [9], highlights the connections between the hosts sorted within a matrix. The traffic between local and remote hosts is clearly displayed, and all the information about the packets is accessible. However, these two tools only display a limited number of hosts (e.g., about 100 hosts for TNV on a 1280x1024 display), and their home-centric view is not suitable for backbone traffic where the terms local and remote hosts are meaningless.

InetVis [10] is a visualization tool used to monitor the network traffic in three-dimensional scatter plots. Traffic is mapped into a cube [11] highlighting the specific patterns for particular anomalies. Although InetVis is adequate enough for monitoring small or extracted traffic (e.g., using IDS [12]), figures generated from heavy traffic (e.g. backbone traffic) are difficult to read and omit a lot of information. Moreover, textual information concerning plotted points cannot be obtained using this tool, whereas, information like port numbers, IP addresses, or TCP flags are usually required to validate anomalies. NVisionIP [13] is another visualization tool that cannot retrieve packet headers — essential to conduct thorough inspections of network traffic — although it is able to display traffic from large networks at several levels of aggregation, and provides detailed statistics on any hosts.

Similar to our work, IDGraphs [14] only displays two-dimensional views based on time. IDGraphs maps an original TCP-flag-based feature (SYN-SYN/ACK values of complete flows) on the vertical axis and emphasizes several patterns for different kind of attacks. However, due to routing policies, the backbone traffic is usually asymmetric and contains numerous incomplete flows, and therefore, the proposed feature based on the TCP flag is irrelevant for analyzing backbone traffic.

Our main contribution is to provide a visualization tool able to display high-volume-traffic from backbone link. Furthermore, global and detailed views of the traffic are available and no assumption on the traffic is required (e.g. complete flows or LAN traffic).

III. Design and Features

A. Goals

Our main goal is to provide an interactive tool, to intuitively understand backbone traffic at different temporal or spatial resolutions, and to validate alarms reported by anomaly detectors. Manually validating results obtained from anomaly detectors is a challenging task because of the multi-dimensionality of network traffic and the large amount of data. Thus, we designed the proposed tool to include the following requirements: the tool has to focus on the significant traffic features to show a network traffic behavior and highlight anomalies in a way that is intelligible to users. It should enable the identification of diverse anomalies by exploring traffic at different scales and in various graphical representations, and permits a particular subset of the whole traffic to be analyzed by filtering the entire set of traffic. A precise understanding of the monitored traffic has to be gained by displaying the original header information and accurate graphs from selected plots. Since this tool is interactive, it has to display figures sufficiently fast, and provide them on different platforms. Script languages or interpreted languages have to be avoided for performance reasons. As the tool has to be quickly operational on several files, it needs to read data directly from the dump files and should not use an intermediate database.

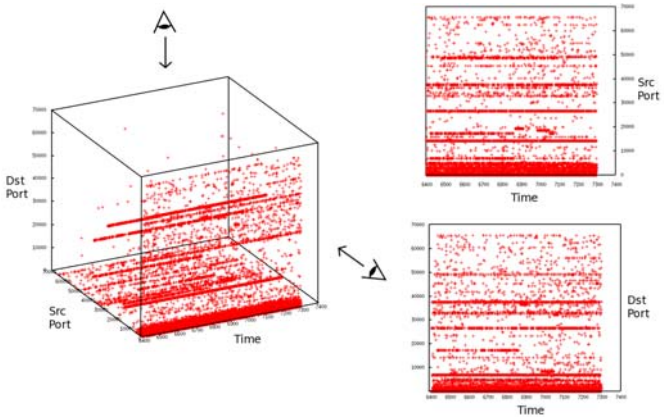


Figure 1. Hard-to-read three-dimensional view and two projections helping to identify values

B. Graphical representations

An asset of the proposed tool is its ability to display a large amount of data and highlight unusual behaviors in two-dimensional views that are easily readable. Obviously, three-dimensional views would provide additional information compared to those that are two-dimensional (hereafter respectively called 3-D and 2-D views). However, to observe such 3-D views we have to project them down onto a 2-D visual aid (e.g., a screen or paper). Two main issues are raised by this dimensionality reduction, namely disorientation and occlusion [15]. Disorientation means that the position of the plotted data is not clear and the values corresponding to the plots are difficult to retrieve. Occlusion occurs when plots hide one another, so information is omitted from view. These two problems are well-known in the field of computer vision, and a common solution is to display several 2-D projections instead of a single 3-D view.

Figure 1 shows an example of a 3-D scatter plot representing network traffic. The three dimensions correspond to the timestamp, source port, and destination port. The main advantage of this representation is to present two traffic features and the time in a single view. Nevertheless, the exact position of each point is difficult to determine and confusing. Also, we need to rotate the cube to verify that plots are not hidden in this

particular view. The occlusion issue is even more important when more data are displayed. However, by projecting data onto the faces of a cube surrounding traffic, we obtain an accurate 2-D view of the traffic. For example, the two scatter plots on the right-hand side of Fig. 1 represent the same traffic; the top one is drawn in the function of the source port and time, while the one at the bottom visualizes the traffic with regard to the destination port and time. These sub-figures are more readily understood than the 3-D representation and allow us to accurately identify the ports numbers corresponding to the plots.

The same type of 2-D scatter plot monitors traffic in the proposed tool, displaying understandable views of the traffic even though we have taken five dimensions into consideration (source port, destination port, source address, destination address, and time). In particular, the network traffic is represented in a five-dimensional space and projected onto several 2-D planes, where the horizontal axis always represents the time, but the vertical axis represents the different traffic features. The following constitutes a list of all the possible ways to represent network traffic using the tool; the first four scatter plots use a color convention where a plotted point is green when it stands for a few packets and becomes progressively redder as the number of packets it represents increases. On the other hand, the next three plots are a time series with their own color convention. Another graphical representation is discussed in Section III-F for a small data set.

1) Destination IP address space: This representation exposes anomalies through their targets. It highlights anomalies that aim at many hosts, or anomalies generating a lot of traffic to a single host/sub-network. The resulting scatter plots have vertical or oblique "lines" (consecutively aligned dots) for anomalies, such as remote exploit attacks, and horizontal "lines" for the targets of DoS attacks, or heavy hitters.

2) Destination port number: This representation emphasizes services targeted in the observed traffic. Obviously, busy services and port scans are highlighted and respectively occur as horizontal and oblique "lines".

3) Source IP address space: This representation highlights the origins of the traffic. Anomalies generating heavy traffic from a single host appear as a horizontal line in the resulting scatter plots. Also, this representation emphasizes various traffics initialized at the same time as DDoS, botnet, or flash crowd.

4) Source port number: This representation reveals the port used by the hosts to communicate. Anomalies based on flooding create as many connections as possible using an increasing source port number. This is translated here as vertical or oblique "lines". This graphical representation is helpful for exposing various kinds of DoSs and remote exploit attacks.

5) Number of packets: Here, the displayed figures are the time series of the number of packets transmitted for each protocol. A red time series is derived for TCP packets, a blue one for UDP, a green one for ICMP, and a black one for other protocols. This representation highlights the misuse of a protocol. For example, a flood generates a considerable number of packets using a particular protocol, easily identifiable as a significant variation in the time series.

6) Number of bytes: Several anomalies cause abnormal variations in the number of bytes. These processes that consume bandwidth are highlighted in this representation as significant variations in the time series.

7) Average packet size: As described by Bardford et al. [1], the average packet size can be taken into consideration to detect anomalies. This reveals the abuse of a particular application, as applications usually use the same packet size for all communications they carry out. This representation is a time series of the average packet size, where anomalies are emphasized by abnormal variations.

C. Tool overview

Figure 2 is an overview of our tool, which is composed of three panels, a small one (W0) with a menu bar and an overview of the traffic, and two larger ones (W1 and W2) displaying the traffic in detail. Since our tool displays only 2-D graphical representation based on a single traffic feature, the two detailed panels (W1 and W2 in Fig. 2) allow the monitoring of two traffic features simultaneously. Users choose which representation has to be displayed in each panel (available representations are listed in Section III-B). Thus, our tool avoids the confusion caused by irrelevant information and focuses on the anomalies as they are generally revealed through unusual uses of one or two traffic features [2]. For example, a network scan can easily be identified by analyzing only the destination address and destination port.

Sections III-D and III-E explain several operations for navigating in W1. Depending on these operations, W2 is automatically updated, providing more information about the traffic displayed in W1 as W2 displays only the packets shown in the W1 view. For example, W1 in Fig. 2 displays a scatter plot of the destination addresses, whereas W2 displays a scatter plot of the source ports. When W1 is zoomed to select a particular sub-network, W2 only presents packets for this sub-network. In W0, the blue rectangle (labeled "Navigation" in Fig. 2) helps us to figure out where the detailed view is located in the entire traffic. W0 also provides a packet header that corresponds to certain points selected by the user.

D. Multi-scale

Anomalies appear at different temporal and spatial scales. Namely, they can last for short or long periods (from an order of seconds to several hours), and they can aim at a single or multiple targets, on one or several ports. The proposed tool allows to zoom in/out independently on each axis. The length of time and feature space (e.g. address space) can be adjusted at any time. This is easily achieved with the mouse wheel, or corresponding buttons. Thus, when long and short-term anomalies are observed, their time duration and their impact in the feature space can easily be estimated.

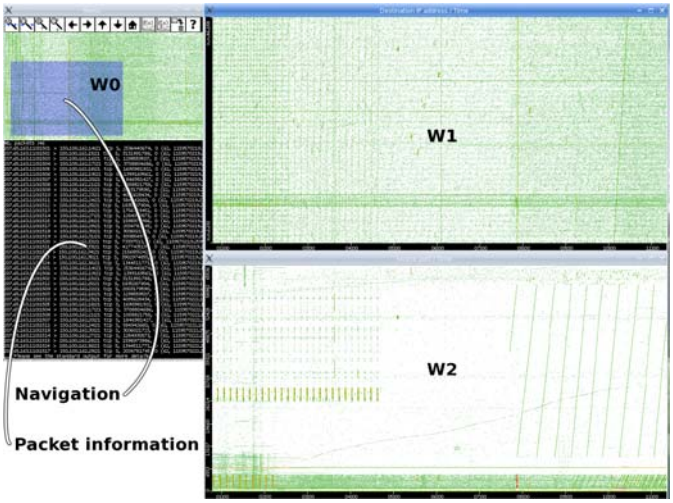


Figure 2. Tool overview.

E. Easy navigation

Inspecting network traffic and thoroughly investigating anomalous traffic requires movement along the traffic trace and a focus on a particular region. The proposed tool lets users conveniently navigate through the analyzed traffic. Only a click on a particular point is required to center the view on that zone.

F. Packet information

Characterizing anomalies is a complicated task, as some of them are only identifiable by inspecting the flags of the packet header. A combination of graphical and textual information is essential for identifying anomalies. Our tool helps users in their investigations by providing useful information about all the plotted pixels. A right click on a point in a figure brings up a zoomed view of the clicked zone, and a particular point can be selected to check the corresponding packets headers, and thus we can learn more about the displayed traffic. The tool also represents the selected data as a graphlet that is similar to those presented in BLINC [6]. These graphlets (or parallel coordinates [16]) allow us to simultaneously visualize more than two dimensions, and intuitively highlight communication patterns. The tool takes advantage of this graphical representation to display only small data sets pointed at by the user (graphlets representing large data sets are too confusing).

G. Input

The tool has to quickly display figures from several input files. Although it would be easier to access data, copying files into an intermediate database is too costly for analyzing daily backbone traffic. Instead of using a database, the tool reads directly from the dump files, like those produced by tcpdump. Also, the tool is able to directly read from compressed files (commonly used to save disk space). Moreover, several files can be given as inputs, and hence, the resulting figures are drawn as all the corresponding files are merged.

TABLE I.
GAIN IN PERFORMANCE DUE TO MECHANISM FOR SEEKING IN PCAP FILES

	User CPU time (clock ticks)	System CPU time (clock ticks)	Time elapsed (minutes:secs)
With “seek structure”	6.00	0.64	00:23.28
Without “seek structure”	10.25	1.43	00:58.42

H. Anomaly description

Reports from anomaly detectors are passed on to the tool in the form of admd files[1], which is a XML schema allowing the annotation of traffic in an easy and flexible way. Thus, anomalies reported by anomaly detectors are quickly identified and inspected as they are displayed in black in all the scatter plots.

I. Portability

Our tool is designed for users utilizing different platforms. We avoided script and interpreted languages for performance purposes, and implemented this application in C++ using only portable libraries to make it available to most users (e.g. views are displayed with the CImg library [17]). Thus, the tool can currently be compiled and executed on different platforms: Unix (Linux and BSD), MacOS, and Windows.

J. Option

The tool is customizable through the command line interface to better fit the needs of the users. One important option from among the many options available permits to filter displayed traffic, thus, the tool monitors only certain sub-traffic from the entire traffic trace. Filters have the same syntax as pcap’s filters (the same as those used in tcpdump) and are based on any field of the packet header. They allow specific sub-traffic to be accurately selected. For example, this option helps investigations into anomalous traffic by displaying only traffic from a suspicious host on certain ports, or by only selecting SYN packets to highlight the probing processes and SYN flood.

K. Snapshot

Saving pictures of traces previously observed is essential for visually comparing or illustrating traffic behaviors. Users can save a snapshot of a particular figure at any time. The snapshots are in PNG format and the size can be specified by the user. The tool can also be used to generate a batch of visualizations from a set of files with the command line interface. For example, visualizations of daily figures from a year of traces can be generated and stored using only one command line[2].

IV. RESULTS

A. Performance

The comfort of navigation and inspection of traffic with our tool is strongly related to its performance and reactivity

[1]Meta-data format and associated tools for the analysis of pcap data: http://admd.sourceforge.net

[2]An example resulting from this feature is the website MAWIviz illustrating all traffic traces of the MAWI archive [18]: http://www.fukuda-lab.org/~romain/MAWIviz/

to one's actions. Since the tool directly reads pcap files, some performance issues are addressed. The main problem is that libpcap does not offer the possibility to directly access a subset of packets corresponding to a given time interval. In practice, the whole traffic trace has to be scanned consuming substantial resources for large traffic traces. Therefore, we consider a dump file to be several parts of the same duration where the first packets of these time slices are called "key packets". Our implementation consists of a data structure that retains information on the "key packets", such as their timestamps and their offsets in the trace file. This data structure helps us to directly access a "key packet" regarding its timestamp. Thus, "key packets" are used as indexes in order to quickly go through the traffic trace. For example, to read a packet at a particular time, t0, the data structure helps us to jump to the "key packet" preceding t0, thereby avoiding having to read numerous unwanted packets prior to this "key packet". Table I lists the gains in performance we obtained with this improvement. The numbers in this table represent the average results from five executions of the same scenario. The scenario consisted of five consecutive zooms in the time space on an uncompressed trace of about 800 MB. The measurements were done on a Linux system with the *time* command, using a computer with 2 GB of RAM and an Intel Core 2 Duo CPU operating at 2.6 GHz. This improvement makes for a comfortable multi-scale navigation through large traffic traces.

B. Darknet data

Figure 3 shows an example of the scatter plots generated from darknet traces taken from a /18 sub-network. As described by Pang et al. [19], darknet (or background radiation) is a type of nonproductive traffic sent to unused address spaces. Darknet data are usually analyzed to characterize anomalies and useful for demonstrating the efficiency of our tool. The vertical axis in the first panel of Fig. 3 stands for the destination addresses, whereas this axis represents the source port numbers in the second panel.

The vertical "lines" in the first panel represent the exploited attacks or any processes using network scans (e.g., (e)). The horizontal "lines" stand for the hosts or sub-networks under heavy attack. They could be the targets of any flood attacks or misconfigurations (e.g., (d) and (f) in the figure).

Other kinds of anomalies are observed in the second panel, and more information about those found in the previous scatter plot are available. Here the vertical "lines" or oblique "lines" represent any procedure using an increasing number of source ports. This is the case in most operating systems when a process opens as many connections as possible. The horizontal "lines" in this panel indicates the constant and heavy traffic from a single port, emphasizing floods, misconfigurations, or heavy-hitters. We can see two sets of consecutive vertical "lines" ((a) and (b) in Fig. 3) appearing at the same time as sudden heavy noise in the first panel. These two behaviors are interpreted as a process trying to access many of the computers of a sub-network within a short time duration (e.g. exploit or worm) as possible. Checking the headers information revealed

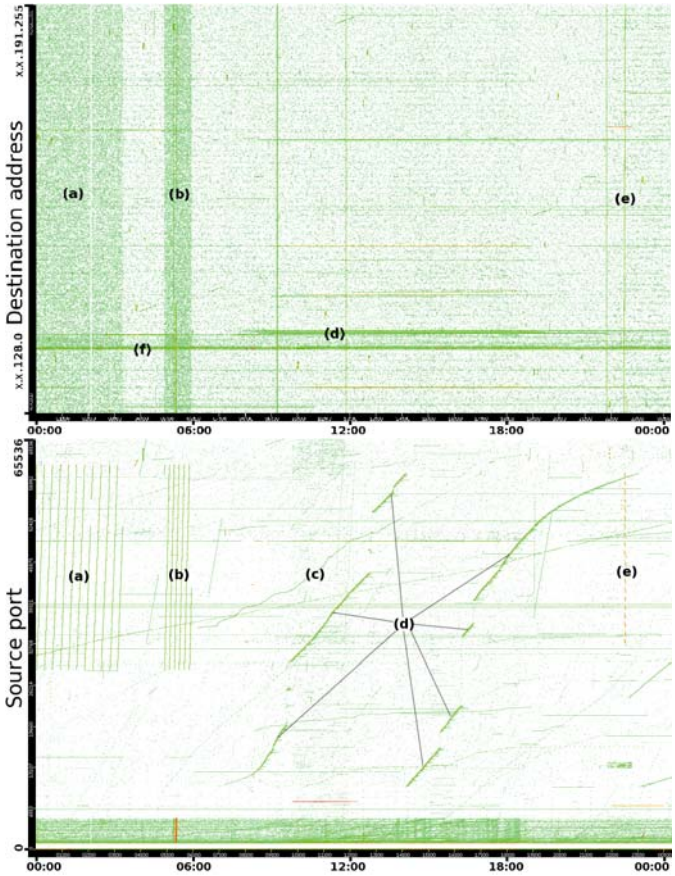


Figure 3. Scatter plots representing darknet data.

that all these packets are directed to port 445. Windows has vulnerabilities in its protocols using this port and several worms have spread through these vulnerabilities. The vertical "line" (e) depicts the same behavior, but within a shorter time frame. Indeed, the packet header information emphasizes an exploit on ssh. We also analyzed the oblique curves (see (c) an (d) in Fig. 3) and detected attacks aimed at services sensitive to attacks. These attacks are not linear because of the variations in time processing or network delays (due to another activity (d) has some jumps in its source port numbers). Checking packet header data reveals that the ports concerned are 80 for (c) and 161 for (d). These services are the targets of well-known attacks driving DoS or buffer overflows. (d) aims at a small sub-network (see (d) in the first panel), whereas (c) is aimed at a single target easily identifiable by zooming in on (f).

C. Network traffic from trans-Pacific link

As an example of anomalies surrounded by legitimate traffic, we analyzed a traffic trace from the MAWI archive [18], which is a set of traffic traces that has been collected by the WIDE Project from 1999. This archive provides large-scale traces taken from trans-Pacific links. The traffic traces are in pcap form without any payload data with both addresses anonymized. Also, the time duration of each trace is fifteen

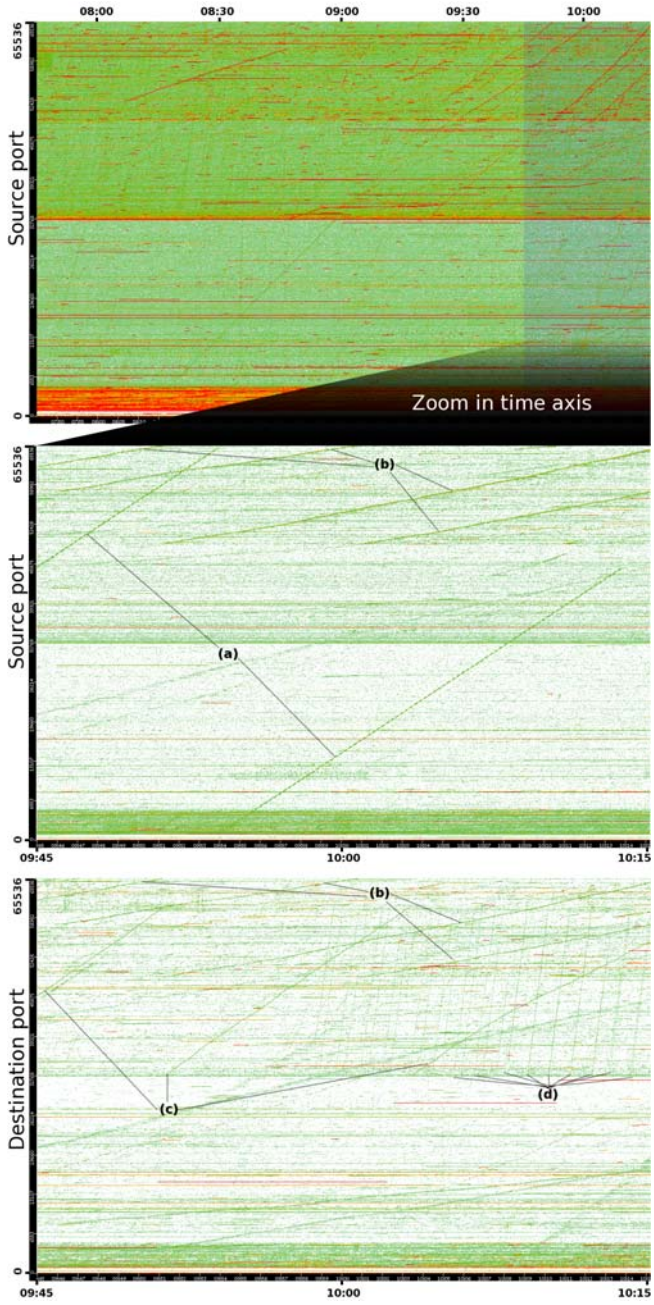


Figure 4. Samplepoint-F from MAWI Working Group Traffic Archive, 2007/01/09

minutes.

Figure 4 depicts views from ten consecutive files of the MAWI database. The total size of these ten files is about 7.6 GB, for a time of 2.5 h and more than 22 million packets. The vertical axis in the first panel stands for source ports. We can easily see that traffic is heavier than in the example presented in previous section. However, we can still distinguish several red "lines" highlighting some intensive uses of network resources. In the following, we focus on the right part of this figure. Consequently, the next scatter plot results from zooming in on the time axis.

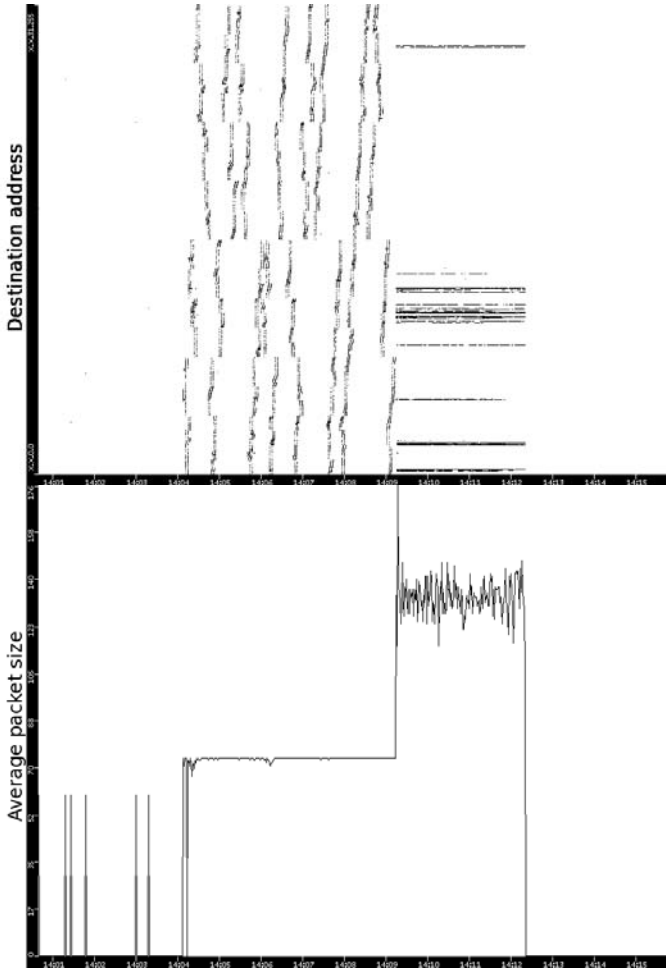


Figure 5. Exploit on port 515. Top: destination address vs. time. Bottom: average packet size vs. time (MAWI archive, 2001/04/14)

The second panel has also been drawn in regard to source ports. Header information helps us to understand plotted pixels; the two oblique "lines" crossing the figure (see (a) in Fig. 4) represent a SYN flood. This is an attack from a single host to several targets, the attacker floods targets on port 443 (usually used for HTTP over SSL). This method is well known and results in buffer overflows in the Private Communications Transport (PCT) protocol implementation in the Microsoft SSL library. The other oblique "lines" represent the same kinds of attacks against other services and from different hosts. In particular, (b) stands for a DDoS attack against a few HTTP servers. The horizontal red "lines" are anomalies consuming bandwidth as in DoS attacks, misconfiguration or heavy-hitters from peer-to-peer networks.

The last panel in Fig. 4 shows the same traffic but in regard to the destination ports. Similar "lines" to those found in the previous panel (b) appear. They stand for the server's reactions to the DDoS attacks previously observed. Also, two kinds of "lines" repeated several times (see (c) and (d)) are highlighted. Both of these are DoS attacks of ACK packets from two distinct hosts against different targets.

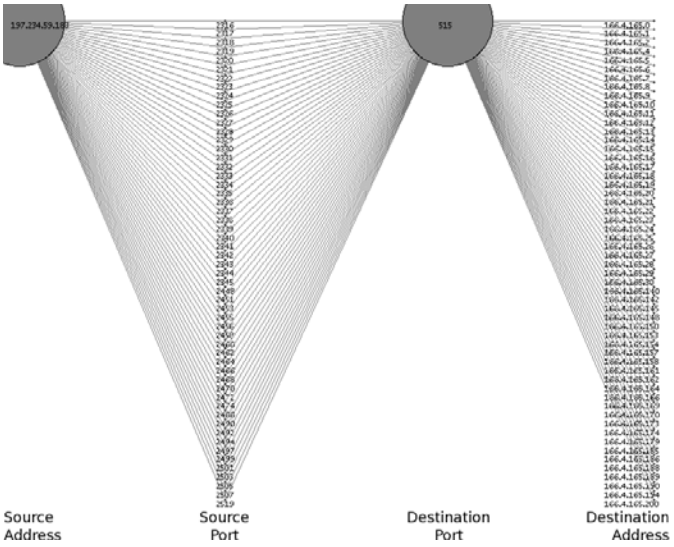
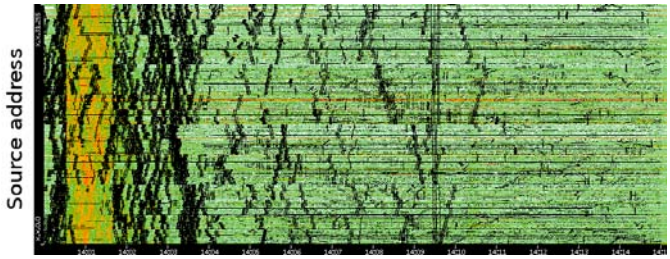
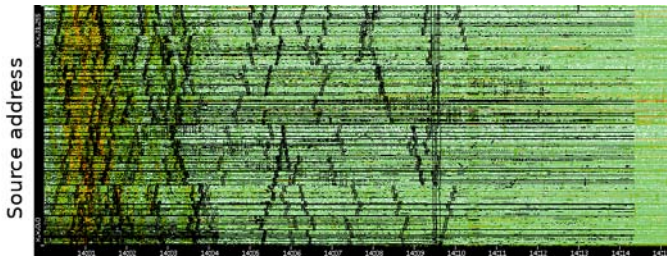


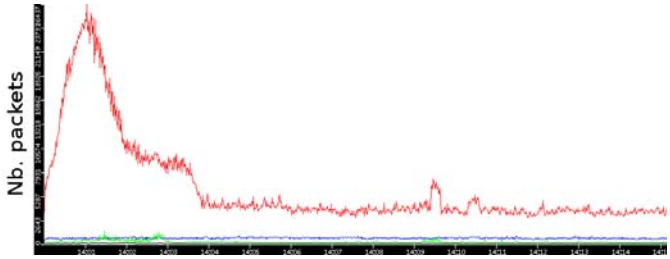
Figure 6. Header information corresponding to several pixels representing traffic from MAWI archive (2004/10/14)



(a) Anomalies reported by anomaly detector based on Hough transform.



(b) Anomalies reported by anomaly detector based on gamma modeling.



(c) Number of packets of analyzed traffic trace.

Figure 7. Highlighting anomalies reported by anomaly detectors in a traffic trace from MAWI archive (2004/08/01) altered by Sasser worm.

D. Manual inspection

1) Inspecting a specific anomaly: The tool helps in inspecting a particular sub-traffic by filtering the entire data before plotting it. The given filters are similar to those in tcpdump allowing for a powerful data extraction. Using filters, the tool is also useful for creating the visualizations of reported anomalies providing additional information in anomaly detector reports. Moreover, filters improve the global performance of the tool as less traffic is displayed.

For example, an anomaly detector [4] reported anomalous traffic on port 515. As this is not a typical target for attacks, we investigated the traffic related to this port. We monitored only the traffic for port 515 (Fig. 5) with the filtering option of our tool. The upper panel of Fig. 5 highlights the destination addresses of the traffic, and depicts two different traffic behaviors; the left-hand side of the scatter plot shows many short communications dispersed over numerous destination hosts, whereas, the right-hand side of the scatter plot displays longer communications concentrated on a few hosts. This can be interpreted as an attacker probing sub-networks to identify hosts with specific security holes, and a few connections are established to compromise detected victims. The bottom part of Fig. 5 represents the average packet size corresponding to the traffic displayed in the scatter plot. This time series also exhibits two different phases; it clearly indicates that the size of the packets during the first half of the analyzed traffic is abnormally constant while the second half is more typically fluctuating. The average size of packets in the first phase is particularly small due to the lack of packet payload used during the probing process. However, the following communications have packet payloads that considerably increase the average packet size.

The traffic behavior can intuitively be understood from Fig. 5, but actual information is still needed to confirm this. The tool supplies header information that corresponds to the displayed plots. Textual header information and a corresponding graphlet are obtained by pointing to a particular plot in the graph.

We retrieved information from several of the plots in Fig. 5 to clearly comprehend the displayed traffic. Figure 6 shows a graphlet corresponding to the header information from various plots selected from the first half of the analyzed traffic. The structure of the graphlet is more interesting than the exact values of the IP addresses or port numbers. It clearly indicates that one host using many ports probes numerous hosts on the same port. The textual data reveals that all packets had a SYN flag set, and confirms that the plotted traffic corresponds to a probing process.

2) Inspecting outputs from anomaly detectors: The proposed tool provides valuable assistance to understand and evaluate anomaly detection methods by displaying their results at any temporal and spatial scales in various views. Indeed, by passing the anomaly detector results and original traffic to the tool, it monitors the reported anomalies and helps in rapidly validating them. Thus, researchers designing anomaly detectors are able to validate at a glance the traffic reported by their anomaly detectors and thoroughly inspect anomalies by retrieving anomalous packet header information.

Two examples of anomalies reported by two distinct

anomaly detectors are depicted in Figure 7, where the anomalous traffics are displayed in black. The two anomaly detectors analyzed a MAWI traffic trace in which the first quarter of the traffic is strongly altered by the spreading of the Sasser worm (see the main peak in Fig. 7(c)). The upper scatter plot (see Fig. 7(a)) depicts 337 anomalies reported by an anomaly detector based on image processing [4]. This view exhibits the inability of this anomaly detector (with the specified parameter set) to detect all Sasser activities during the main outbreak of the worm. This case emphasizes the valuable support provided by the tool as this fact could not be deduced by only inspecting the textual results outputted by the anomaly detector.

The middle scatter plot depicts 332 anomalies obtained with another anomaly detector based on multi-scale gamma modeling [3]. A quick visual comparison of the two views (Fig. 7(a) and Fig. 7(b)) indicates that these two anomaly detectors identified many distinct traffics — particularly during the peak identified in the first quarter of the trace — although they reported a similar amount of anomalies. This comparison is quickly derived from the two views provided by the tool, whereas, similar conclusions are usually deduced from a time-consuming manual analysis of the two anomaly detectors outputs.

E. Temporal-Spatial patterns in anomalous traffic

During our experiments we observed particular patterns that stood for certain kinds of anomalies. These patterns exhibit some important properties of the anomalies such as its range of targets and sources, its operational speed, and its time duration. It also provides certain information on the mechanisms used by the anomalies, particularly the uses of the source ports.

1) Coarse view: At large scales certain anomalies are easily identified as sudden changes in the main traffic behavior or in the usage of a particular protocol. For example, Figure 8 displays three months of darknet traffic recorded while the first two versions of the Conficker worm were released. This figure shows that first, a sharp increase in the number of source IP addresses and number of packets clearly signaling the start of the worm spread (labeled *Conficker.A* in Fig. 8). Second, another growth of these quantities depicts the release of the second version of the worm and its aggressive behavior in terms of the network resources consumption (labeled *Conficker.B* in Fig. 8). The scatter plot of the destination port (see middle scatter plot of Fig. 8) reveals that the first version of the worm is communicating with the other hosts using random port numbers ranging over $(1024, 5120)$. These types of communications disappear after the second release is unveiled, highlighting that different mechanisms are implemented in this new version.

2) Fine view: On smaller scales, we observe other kinds of patterns exhibiting anomalies through their abnormal uses of the traffic features. We emphasize that these patterns are in accordance with those identified by an anomaly detector based on pattern recognition [4]. For example, Figure 9 is composed of different anomalies observed on the same day (2004/10/14). The vertical axis represents the destination addresses for scatter plots at the top of the figure and source ports for those at the bottom. Three different anomalies are emphasized in this figure.

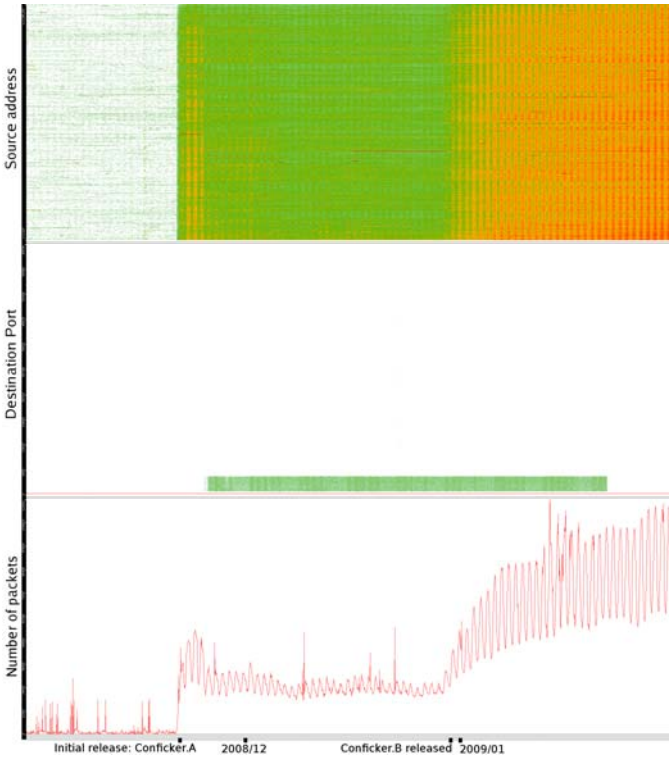


Figure 8. Three months (2008/12, 2009/01-02) of darknet traffic related to port 445 during Conficker outbreak.

The two representations ((A) and (B)) on the left-hand side of Fig. 9 stand for an exploit against a Windows service operating on port 445. These were obtained by displaying only the traffic related to a specific IP address, X. The upper representation (A) shows long vertical lines meaning that X contacted numerous hosts within three short periods of time. The header information revealed that all the packets corresponding to these connections were directed to port 445 with the TCP SYN flag set. The representation of the source port (B) indicates that the traffic was initiated from a limited pool of high number ports (< 1024). This traffic is clearly malicious and corresponds to a probing process looking quickly for victims.

The two scatter plots labeled (C) and (D) in Fig. 9 stand for network traffics from a single host lasting for the entire traffic trace. The upper scatter plot displays long oblique lines, meaning that this traffic also correspond to a probing process. However, the inclination of the lines indicates a slower process than the one previously discussed. Moreover, the lower scatter plot (labeled (D)) shows a horizontal line representing only a couple of source ports.

The two representations, (E) and (F), on the right-hand side of Fig. 9 correspond to a spreading of the Sasser worm. Traffic from different hosts are displayed in these figures. The vertical structures in the upper scatter plot represent the probing proce-

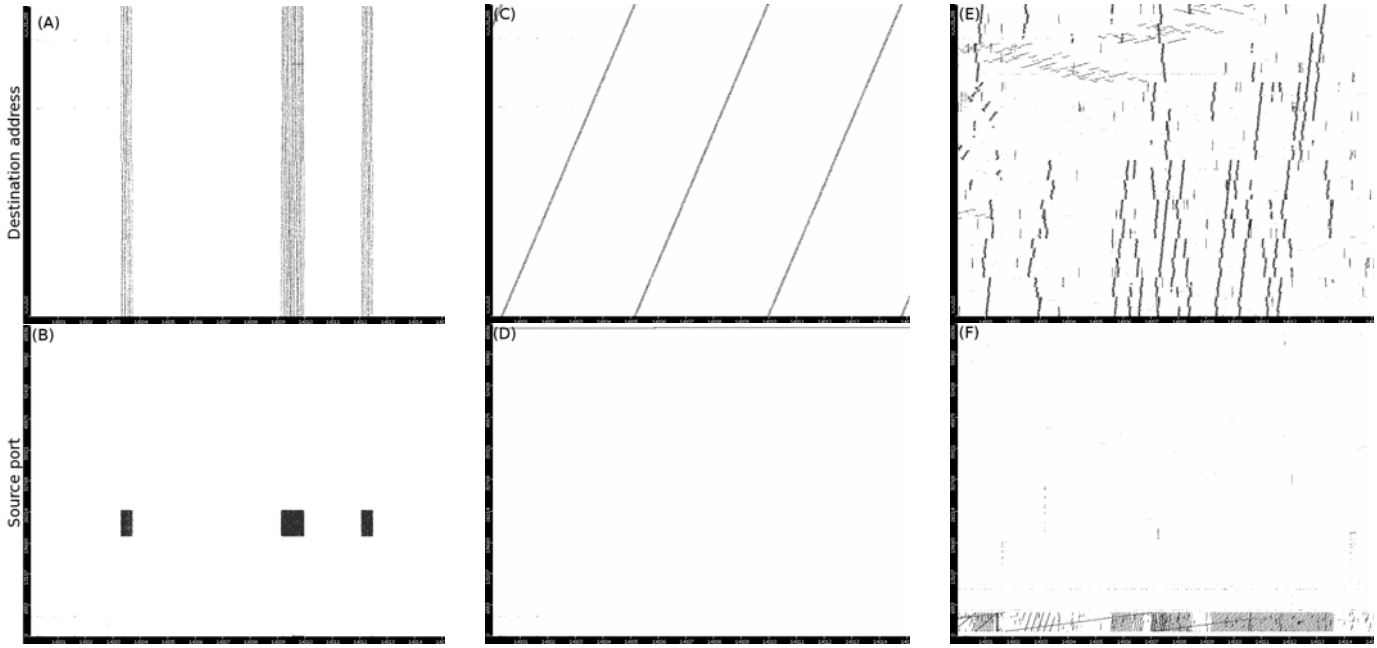


Figure 9. Different patterns observed in same traffic trace (MAWI archive, 2004/10/14). Top: destination IP vs. time, Bottom: source port vs. time.

dure done by the worm, and we noticed that different spreading are observed. The scatter plot representing the source ports (labeled (F)) indicates that this implementation of the Sasser worm generates traffic with only low source ports numbers that are linearly increasing. The shape and height of the observed "lines" provides a signature for this variant of the worm that can be easily identified in other traffic traces.

V. Conclusion and Future Work

We outlined the need for understanding the network traffic behavior and evaluating anomaly detectors. To achieve these purposes, we designed and implemented a tool graphically representing the network traffic on any temporal and spatial scales. The main contribution of this tool is to display global and detailed views of the network traffic focusing on anomalies. Interesting traffic behaviors are uncovered by interactively exploring the traffic traces, and detailed information is also provided to enable data to be thoroughly investigated. Traffic from specific hosts or services is extracted by using a filtering mechanism. Thus, particular types of sub-traffics are displayed without surrounding noise and can easily be investigated. Furthermore, anomalies reported by anomaly detectors are highlighted in full view and their validation can then be facilitated. The tool runs on different platforms, licenced under the GNU General Public License (GPLv3), and is freely downloadable [3]. We verified the usefulness of our tool by evaluating it on several traffic traces; darknet traces highlighting several patterns for different anomalies, and traces taken from a backbone link where anomalies surrounded by heavy noise were still identifiable. Observation of recent threats, such as the Conficker worm, can also be carried out. We conducted manual inspections of the alarms reported by an anomaly detector and visually compared the outputs of two distinct approaches. Also, we listed several patterns standing for distinct anomalies and noticed that they are consistent with those found in [4].

One important project we intend to carry out in the future is to add a capability to process raw packets taken directly from a network interface.

[3]The tool is available at http://www.fukuda-lab.org/~romain/mulot

Acknowledgments

We would like to thank G. Dewaele et al. for having provided us the source code of their anomaly detector. This work is partially supported by MIC SCOPE.

References

[1] P. Barford, J. Kline, D. Plonka, and A. Ron, "A signal analysis of network traffic anomalies," *IMW '02*, pp. 71–82, 2002.

[2] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," *SIGCOMM '05*, pp. 217–228, 2005.

[3] G. Dewaele, K. Fukuda, P. Borgnat, P. Abry, and K. Cho, "Extracting hidden anomalies using sketch and non gaussian multiresolution statistical detection procedures," *SIGCOMM LSAD '07*, pp. 145–152, 2007.

[4] R. Fontugne, Y. Himura, and K. Fukuda, "Evaluation of anomaly detection method based on pattern recognition," *IEICE Trans. on Commun.*, vol. E93-B, no. 2, Feb. 2010 (to appear).

[5] A. S. Haakon Ringberg and J. Rexford, "Webclass: adding rigor to manual labeling of traffic anomalies," *SIGCOMM CCR*, vol. 38, no. 1, pp. 35–38, 2008.

[6] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "Blinc: multilevel traffic classification in the dark," *SIGCOMM '05*, vol. 35, no. 4, 2005.

[7] Tcpdump and libpcap, http://www.tcpdump.org/.

[8] F. Fischer, F. Mansmann, D. A. Keim, S. Pietzko, and M. Waldvogel, "Large-scale network monitoring for visual analysis of attacks," *VizSEC '08*, pp. 111–118, 2008.
[9] J. R. Goodall, W. G. Lutters, P. Rheingans, and A. Komlodi, "Focusing on context in network traffic analysis," *IEEE Comput. Graph. Appl.*, vol. 26, no. 2, pp. 72–80, 2006.
[10] J.-P. van Riel and B. Irwin, "Inetvis, a visual tool for network telescope traffic analysis," *Afrigaph '06*, pp. 85–89, 2006.
[11] S. Lau, "The spinning cube of potential doom," *Commun. ACM*, vol. 47, no. 6, pp. 25–26, 2004.
[12] B. Irwin and J. P. Riel, "Using inetvis to evaluate snort and bro scan detection on a network telescope," *VizSEC '07*, pp. 255–273, 2007.
[13] K. Lakkaraju, R. Bearavolu, A. Slagell, W. Yurcik, and S. North, "Closing-the-loop in nvisionip: Integrating discovery and search in security visualizations," *VizSEC '05*, p. 9, 2005.
[14] P. Ren, Y. Gao, Z. Li, Y. Chen, and B. Watson, "Idgraphs: Intrusion detection and analysis using histographs," *VizSEC '05*, 2005.
[15] R. Marty, *Applied Security Visualization*, 1st ed. Addison-Wesley Professional, August 2008.
[16] A. Inselberg, "The plane with parallel coordinates," *The Visual Computer*, vol. V1, no. 4, pp. 69–91, December 1985.
[17] The c++ template image processing library. The CImg Library : http://cimg.sourceforge.net.
[18] K. Cho, K. Mitsuya, and A. Kato, "Traffic data repository at the WIDE project," in *USENIX 2000 Annual Technical Conference: FREENIX Track*, June 2000, pp. 263–270.
[19] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson, "Characteristics of internet background radiation," *IMC '04*, pp. 27–40, 2004.