

A Visualization Tool for Exploring Multi-scale Network Traffic Anomalies

Romain Fontugne

The Graduate University for Advanced Studies
Tokyo, Japan

Toshio Hirotsu

Hosei University
Tokyo, Japan

Kensuke Fukuda

National Institute of Informatics / PRESTO, JST
Tokyo, Japan

Abstract—Visualization is an intuitive and powerful way of understanding the evolution of huge amounts of network traffic in terms of characterizing network anomalies. We propose an interactive tool to display, explore, and understand network traffic focusing on anomalies. It displays traffic on different temporal and spatial (address and port) scales and lets users navigate network data by using a simple interface. Different graphical representations are used to highlight anomalies quickly, and textual packet information about corresponding plotted points are provided. The proposed tool provides good support for understanding traffic behavior and for evaluating the effectiveness of anomaly detection method. The tool directly reads dump files and uses no intermediate database in daily operations. This paper demonstrates several examples emphasizing specific patterns for various anomalies.

I. INTRODUCTION

Many attractive network services are currently available for ordinary users through the Internet. Consequently, Internet traffic is still growing at a fast rate, and therefore has increasingly more anomalies such as misconfigurations, failures, and attacks. These improper uses of networks consume bandwidth and adversely affect the performance of networks. Thus, these anomalies penalize legitimate applications from using optimal resources. Since Internet backbones are particularly altered by anomalous traffic, the quick and accurate detection of anomalies in backbone traffic has been a hot topic in this field (e.g., [1], [2], [3], [4]). However, due to the lack of labeled backbone traffic traces, researchers have had to manually investigate dump files or flow records to check and then validate their results. This can be a baffling problem because it is laborious to have to identify a few thousand harmful packets from millions of innocuous one.

Dump files are usually investigated intuitively by using graphical representations of traffic. Visualization tools display varied information in a way that highlights important features of the traffic; thus, their main characteristics can be understood at a glance. Many levels of aggregate traffic providing several degrees of information through analysis, can be achieved by visualizing network traffic. For example, although, a representation with a time series is useful for showing a huge number of flows, it can only display one feature and the time, whereas a graphlet [5] can depict several dimensions of a few flows.

This paper proposes a tool for visualizing and exploring network traffic on all temporal and spatial (address and port) scales. It aims at helping researchers to validate their results

with an anomaly detector. Our tool assists researchers in inspecting traffic with five basic features. (1) It can display network traffic at different resolutions, and allows zooming in/out along the time axis or address/port space. (2) 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). All these graphical representations can easily be understood because they are limited to two dimensions. To enable traffic displayed in one of these graphical representations to be better understood, the tool allows two figures to be compared simultaneously. (3) By analyzing traffic, researchers have to focus on particular time and space regions. The tool allows network traffic to be easily navigated. (4) Also, the tool allows us to inspect traffic in detail by displaying an accurate graphlet and retrieving original data from plotted points. (5) This tool can run on different platforms on a daily basis, it uses no intermediate database, and it directly reads dump files (pcap form [6]).

II. RELATED WORK

Many visualization tools have assisted researchers and network operators to monitor network traffic. For example, Fischer et al. [7] and Goodall et al. [8] presented two interesting tools that focused on the detection of anomalies. The former [7] monitored a local network using a TreeMap visualization with remote hosts at the border, and linked hosts by splines. The tool was used to check alarms reported by IDS, and find large-scale attacks. The latter, tnv [8], highlighted connections between hosts (sorted within a matrix) with a color for each protocol. Traffic between local and remote hosts was clearly displayed, and all information about packets was accessible. However, these tools could only display a limited number of hosts (e.g., about 100 hosts for tnv on a 1280x1024 display), and their home-centric view could not be applied to backbone traffic where local and remote hosts are meaningless.

InetVis [9] is a three-dimensional scatter plot visualization for network traffic. It represents network traffic in a three-dimensional way and highlights anomalies. Traffic traces are mapped in a cube [10] wherein each axis provides information. InetVis displays specific patterns for particular anomalies, but it is only useful with extracted traffic (e.g., using IDS [11]), since figures generated from heavy traffic are difficult to read. Textual information about plotted points cannot be obtained with this tool, whereas, information like port numbers, IP ad-

addresses or TCP flags is usually required to identify anomalies. NVisionIP [12] displays traffic from large networks at several levels of aggregation, and provides detailed statistics on hosts. However, packet headers cannot be retrieved, which restrict thorough inspections of traffic.

Similar to our tool, IDGraphs [13] only displays two-dimensional visualizations based on the 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 the asymmetry of backbone traffic and the small number of complete flows captured at a single point, the proposed feature is irrelevant for our purposes.

III. DESIGN AND FEATURES

A. Goals

Our main aim was to provide an interactive tool, to intuitively understand backbone traffic at different temporal or spatial resolutions, and to validate and characterize anomalies reported by an anomaly detector. 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. Our visualization tool has to focus on significant traffic features to show network traffic behavior and highlight anomalies in a way that is intelligible to users. Due to the large variety of anomalies, anomalous traffic can be identified by exploring network traffic on different scales, and a largest variety of anomalies can be displayed using several graphical representations. Only a particular subset can be analyzed by filtering the entire amount of traffic. A clear understanding of monitored traffic has to be gained by displaying original header information and accurate graphs from selected plots. Since this tool is interactive, it has to display figures sufficiently fast, and to provide these on different platforms. Script languages or interpreted languages have to be avoided reasons of performance. As the tool has to be quickly operational on several files, it needs to read data directly from dump files and should not use an intermediate database.

B. Graphical representations

The manner in which a large data set highlighting unusual behavior is clearly displayed is one of the main assets of the proposed tool. Obviously, three-dimensional views provide additional information compared to those that are two dimensional. However, to observe such views we have to project them down onto a two-dimensional visual aid (e.g., a screen or paper). Two main issues are raised by this reduced dimensionality, namely disorientation and occlusion [14]. Disorientation means that the position of the plotted data is not clear and values corresponding to certain plots are not retrieved. Occlusion occurs when plots are hiding one another, so information is omitted from the view. These two problems are well-known in computer vision, and a common solution is to display several two-dimensional projections instead of a single three-dimension view.

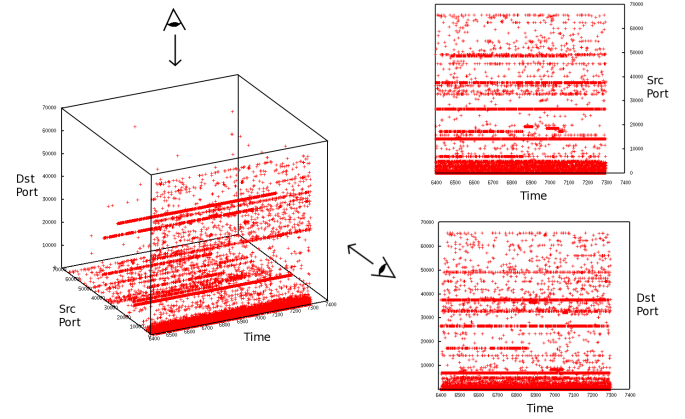


Fig. 1. Hard-to-read three-dimensional view and two projections helping to identify values

Figure 1 shows an example of three-dimensional 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 only one representation. However, the exact position of each point is difficult to determine and it can be confusing to look at this scatter plot. 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. By projecting data onto the faces of a cube surrounding traffic, we obtain an accurate view of the traffic in two dimensions. The two scatter plots on the right of Fig. 1 represent the same traffic; the top one is drawn in function of the source port and time, while the one at the bottom visualizes traffic with regard to the destination port and time. These sub-figures are more readily understood than a three-dimensional representation and allow us to accurately identify the main ports that are used.

We used the same type of two-dimensional scatter plot to monitor traffic in the present work. Thereby, our tool still displays understandable visualizations, even though we have taken five dimensions (source port, destination port, source address, destination address and the time) into consideration. Network traffic is represented in a five-dimensional space and projected onto several two-dimensional planes, where the horizontal axis always indicates the time, but the vertical axis represents different traffic features. The following constitutes a list of all possible ways of representing network traffic with the tool. The first four scatter plots use a color convention where a plotted point is green when it stands for few packets and this becomes progressively redder as the number of packets it represents increases. On the other hand, next three plots are time series with their own color convention. Another graphical representation is discussed in Section III-F for small data set.

1) *Destination IP address space*: This representation exposes anomalies through their targets. It highlights anomalies aimed 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 targets of DoS attacks, or heavy hitters.

2) *Destination port number*: This representation emphasizes services used in the traffic observed. Obviously, busy services and port scans are highlighted in the resulting scatter plots. The first occurs as horizontal “lines” and the second as oblique.

3) *Source IP address space*: This representation highlights the origins of traffic. Anomalies generating heavy traffic from a single host appear as a horizontal line in the resulting scatter plots.

4) *Source port number*: This representation reveals the port used by hosts to communicate. Many anomalies are based on flooding, and create as many connections as possible. The source port number in these attacks increases linearly. This is translated here as vertical or oblique “lines”. This graphical representation is helpful to expose various kinds of DoSs and remote exploit attacks.

5) *Number of packets*: Here, the displayed figures are the time series for 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 others protocols. This representation highlights misuse of a protocol. For example, a misconfiguration or a flood generates a considerable number of packets using a particular protocol, which can easily be identified as significant variations 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 abuse of a particular application, because 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. Overview of the tool

Figure 2 has a snapshot of our tool, which is composed of three panels, i.e., 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. Although the tool only displays a two-dimensional graphical representation based on a single traffic feature, the two detailed panels (W1 and W2 in Fig.2) allow two traffic features to be monitored simultaneously. Users are able to choose which representation (e.g., destination address or destination port) has to be depicted in each panel. Available representations are listed in Section III-B. Anomalies are generally revealed through unusual uses of one or two traffic features [2]. As the tool does not display all considered traffic features at the same time, confusion caused by irrelevant information is avoided. For example, a network scan can easily be identified by only analyzing the destination address and destination port. Depending on the way

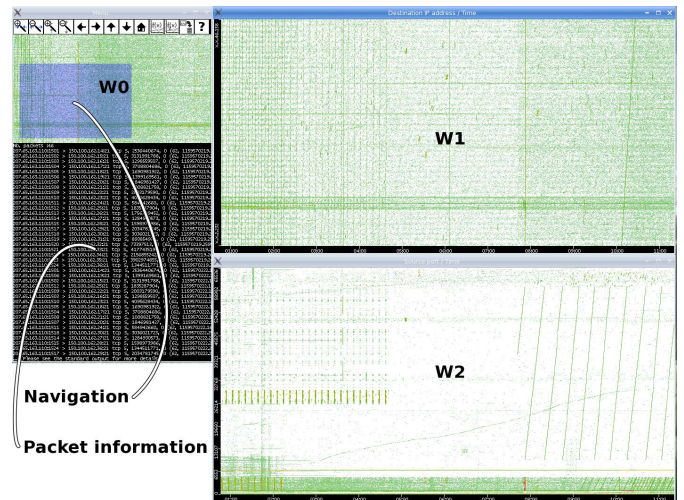


Fig. 2. Overview of the tool.

the traffic is represented we can recognize different patterns that characterize anomalies.

Sections III-D and III-E explain several operations to navigate in W1. Depending on these operations, W2 is automatically updated, providing more information about traffic displayed in W1. W1 acts as a filter for W2, since W2 only displays packets shown in the view of W1. For example, W1 in Fig. 2 displays a scatter plot of destination addresses, whereas W2 displays a scatter plot of 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 view is observed from all traffic traces. W0 also provides raw packet information on a selected point, and informs users about timestamps, IP addresses, port numbers and protocol details.

D. Multi-scale

Anomalies have an impact at different temporal or spatial scales. They can last for long or short periods (from an order of seconds to several hours). Also, anomalies can aim at a single or multiple targets, on one or several ports. The tool allows us to zoom in/out independently on each axis. The length of time and the address space (for example) can be adjusted at any time. This is simply done with the mouse wheel, or corresponding buttons. Long-duration anomalies as well as those of short duration are easy to observe, and attacks spread can also be estimated.

E. Easy navigation

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

F. Packet information

The characterization of anomalies is a complicated task. Several anomalies can only be characterized by 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 plotted pixels. A right click on a point in a figure show up a zoomed view of the clicked zone, and a particular point can be selected to check corresponding packet headers, and learn more about the traffic that is displayed. The tool also represents pointed data as a graphlet similar to those presented in BLINC [5]. These graphlets (or parallel coordinates [15]) allow us to simultaneously visualize more than two dimensions, and intuitively highlight communication patterns. Our 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. Instead of using a database, the tool reads directly from dump files, like those produced by tcpdump. The tool is able to directly read from compressed files, since dump files are usually compressed to save disk space. Several files can be given as input; thereby, the resulting figures are drawn as all corresponding files are merged.

H. Portability

Our tool was designed for users utilizing different platforms. We avoided script and interpreted languages for performance purposes. We implemented this application in C++ using only portable libraries to make it available to most users and for it to run sufficiently fast. The tool can currently be compiled and executed on different platforms: Unix (Linux and BSD), MacOS, and Windows.

I. Option

The tool can be customized to better fit the needs of users. These adjustments can be done through the command line interface. Many options are available and one of the most important permits filters to be specified to select particular sub-traffic from the traffic trace being analyzed. That is, the given filter should have the same syntax as pcap's filters (the same as those used in tcpdump). Filters are based on any field of the packet header, and they allow specific sub-traffic to be accurately selected. For example, this option helps investigations into anomalous traffic by only displaying traffic from a suspicious host on certain ports, or by only selecting SYN packets to highlight probing processes and SYN flood.

J. Screenshot

It is useful for users to save pictures of traces previously observed to visually compare or characterize traffic behaviors. They can save a screenshot of a particular figure at any time. 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 with only one command line.

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:seconds)
With "seek structure"	6.00	0.64	00:23.28
Without "seek structure"	10.25	1.43	00:58.42

IV. RESULTS

A. Performance

The tool is implemented in C++, and it generates visualizations using the CImg library [16]. It needs to be rapidly responsive to one's actions, i.e., the user needs to be comfortable with interactive navigation and manual inspection of traffic traces. Since the tool directly reads pcap files, some performance issues have to be addressed. The main problem is how to directly access a subset of packets from the entire traffic trace. In practice, we need to seek packets in dump files corresponding to a given time. Libpcap does not offer that possibility, and this improvement would allow us to browse long traces. We consider a dump file to be several parts of the same duration, and the first packets of these time slices are called "key packets". Our implementation consists of a data structure that retains information about "key packets", such as packet timestamps and packet offsets in the trace file. That is, this data structure provides several timestamps and helps us to directly access the "key packet" corresponding to the timestamp. Thus, "key packets" for a given timestamp are useful for quickly seeking packets in dump files. For example, to read a packet at a particular time, t_0 , the data structure helps us to jump to the "key packet" preceding t_0 , thereby avoiding having to read unwanted packets. 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 time space on an uncompressed trace of about 800 MB. The measurements were done on a Linux system with the *time* command, and the machine had 2GB of RAM and an Intel Core 2 Duo operating at 2.6 GHz. This improvement permitted comfortable multi-scale navigation in large traffic traces.

B. Darknet data

Figure 3 shows an example of scatter plots generated from darknet traces taken from a /18 sub-network. As described by Pang et al. [17], darknet (or background radiation) means nonproductive traffic sent to unused address spaces. Darknet data are usually analyzed to characterize anomalies, and we used them here to demonstrate the efficiency of the tool. The vertical axis in the first panel stands for destination addresses, whereas this axis represents the source port number in the second panel.

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

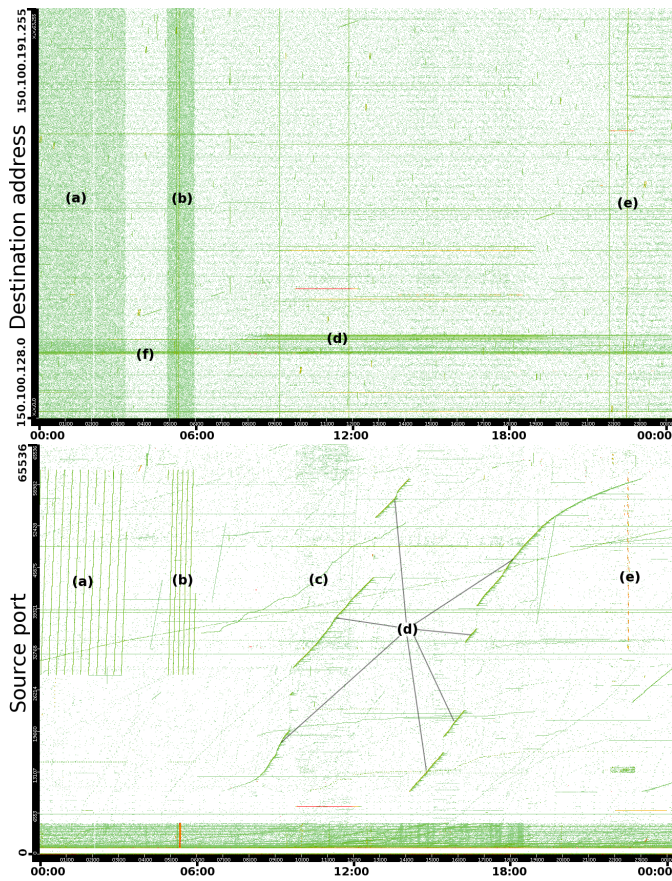


Fig. 3. Scatter plots representing darknet data.

We can see other kinds of anomalies in the second panel, and we obtained more information about those found in the previous scatter plot. Here the vertical “lines” or oblique “lines” mean 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 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 can be interpreted as a process trying to access most computers of a sub-network within a short duration. These are typically exploit or worm behaviors. Checking the headers information, note that all these packets are directed to port 445. Windows has vulnerabilities in protocols using this port. Several worms spread by exploiting these vulnerabilities. The vertical “line” (e) behaves in the same way, but within a shorter time frame. In fact, the packet information for (e) informs us about an exploit on ssh. We also analyzed the oblique curves (see (c) and (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). The ports concerned are 80 for (c) and 161 for (d). These services have

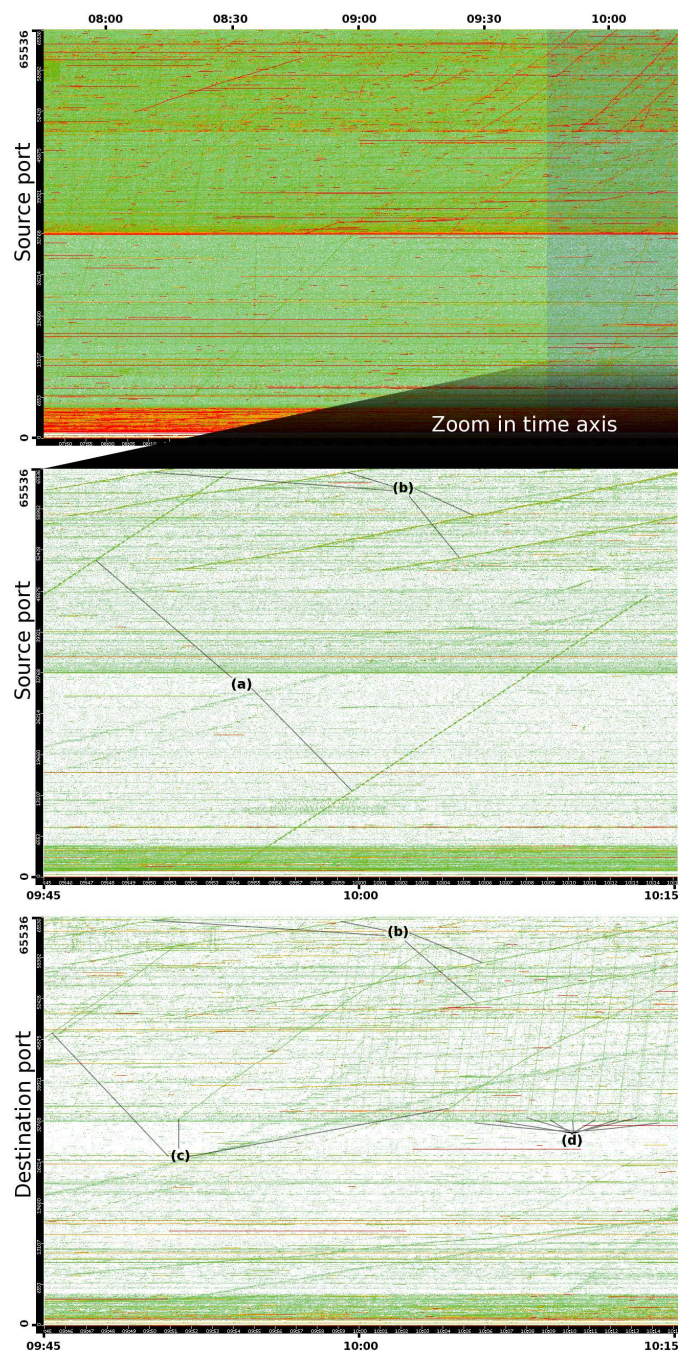


Fig. 4. Samplepoint-F from MAWI Working Group Traffic Archive, 2007/01/09

well known anomalies driving DoS or buffer overflows. (d) aim 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 a trans-Pacific link

Figure 3 depicts the shapes of anomalies using darknet data. Here, let us present another example with anomalies in large and complex traffic. We analyzed a traffic trace from the MAWI archive [18], [19], 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 payload data with both addresses anonymized. Also, the time duration of each trace is fifteen minutes.

Figure 4 shows the graphical representations that were generated, from ten consecutive files from 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 previous example. However, we can still distinguish several red “lines” from the entire amount of traffic. In the following, we have focused on the right part of this figure. Consequently, the next scatter plot results from zooming in on the time axis.

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) represents 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 destination ports. We can see similar “lines” to those found in the previous panel, (b). They stand for the servers’s reactions to the DDoS attacks previously observed. Also, we can see two kinds of “lines” repeated several times (see (c) and (d)). Both of these are DoS attacks of ACK packets from two distinct hosts against different targets.

D. Manual inspection

The proposed tool provides valuable assistance to researchers who are designing anomaly detection method, and manually validating their results. The labeling of anomalies in traffic traces, such as WebClass [20] does, is beyond the goals of the present work. However, the two tools are a good combination to validate and label anomalies detected by anomaly detectors.

Due to the lack of labeled backbone traffic traces, validating results from anomaly detection methods involve manual inspection of dump files. Anomalous traffic is characterized by extracting and analyzing a sub-set of data corresponding to particular IP addresses and/or port numbers at specific times. The tool helps sub-traffic to be visualized based on various criteria by filtering the entire data before they are plotted. The given filters are similar to those in tcpdump allowing powerful data extraction. Using filters, the tool is also useful for creating visualizations reported with anomalies to provide intuitive characteristics of anomalies.

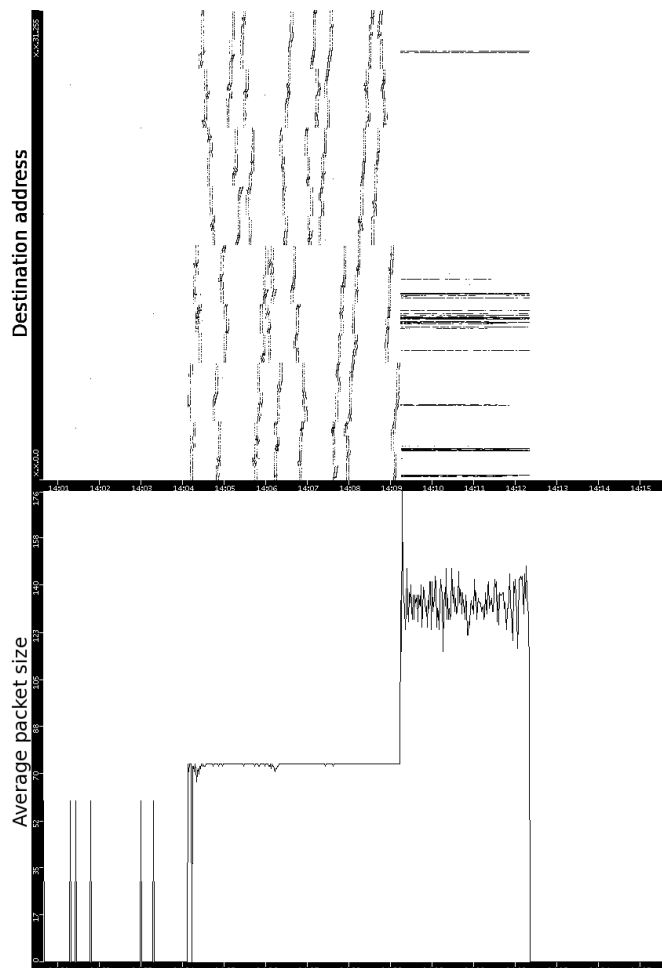


Fig. 5. Exploit on port 515. Top: destination address vs. time. Bottom: average packet size vs. time (MAWI Archive, 2001/04/14)

For example, an algorithm for detecting anomalies [4] reported an anomaly on port 515. As this is not a typical target for attacks, we decided to investigate the abnormal behavior that was detected for this port. We monitored the traffic for port 515 (see Fig. 5) with the filtering option of our tool. We can distinguish two different traffic behaviors in the upper panel of Fig. 5. The left side of the scatter plot shows many short communications dispersed over numerous destination hosts. However, the right 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 where security holes can be exploited, 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 in traffic behavior. It clearly indicates that the size of packets during the first half of the analyzed traffic is abnormally constant while the second half is more typical and fluctuating. The average size of packets in the first phase is particularly small due to the lack of packet payload used during the probing

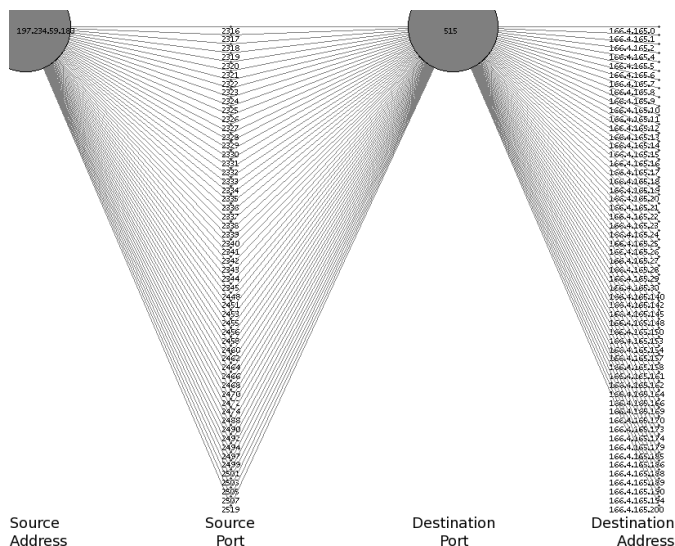


Fig. 6. Header information corresponding to a few pixels representing traffic from MAWI Archive (2004/10/14)

process. However, the following communications have packet payloads increasing considerably the average packet size.

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

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

E. Temporal-Spatial patterns in anomalous traffic

We observed particular patterns that represented for different kinds of anomalies during our experiments. A pattern can exhibit important properties of an anomaly such as its range of targets and sources, its speed in operation and its duration. It also provides certain information on mechanisms employed by anomalies, particularly the uses of source ports.

Figure 7 has representations of different anomalies observed on the same day (2004/10/14). The vertical axis represents destination addresses in scatter plots at the top of the figure and source ports for those at the bottom. There are three different anomalies in this figure.

The two representation ((A) and (B)) at the left of Fig. 7 stand for an exploit against a Windows service listening at port 445. These were obtained by only displaying 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. We found that all packets corresponding to these connections were directed to port 445 with the TCP SYN flag set by checking the header information. The representation for the source port (B) indicates that traffic was initiated from a limited pool of ports with high numbers (< 1024). This traffic behavior is clearly malicious and corresponds to a process of quickly looking for victims.

The two scatter plots labeled (C) and (D) in Fig. 7 stand for network activity from a single host that lasts throughout the entire traffic trace. The upper scatter plot displays long oblique lines, meaning that this traffic also corresponded to a probing process. However, the inclination of the lines indicates that this searching process is slower 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), at the right side of Fig. 7 correspond to the spread of the Sasser worm. The traffic from different hosts are displayed in these. The vertical structures in the upper picture represent the probing procedure done by the worm, and we can see different speeds at which the worm spreads. The scatter plot for source ports indicates that this implementation of the Sasser worm only uses ports with low numbers that are linearly increasing. The shape and height of observed “lines” provide a signature for this variant of the worm that can easily be identified in other traffic traces.

V. DISCUSSION

Analysis of network traffic is complicated because of the number of dimensions involved, and the huge volume that needs to be handled. Our work was motivated by the difficulty of investigating traffic traces with only time series and log files. Understanding network traffic with only textual information is a complicated task, whereas complex traffic can be rapidly and intuitively understood from visualizations. Our tool takes advantage of visualization to reduce the complexity of traffic and displays important quantity of data in two views.

The main contribution of our tool is that it displays a global view of traffic as well as a detailed graphlet and original data. Traffic is able to be interactively explored by zooming in/out, and moving in on traffic traces. These actions help to uncover interesting traffic behavior. Detailed information is also provided to enable data to be thoroughly investigated. Consequently, a particular shape can be identified from visualizations and accurate characteristics of traffic can be obtained with packet-header information. Traffic from specific hosts or services are inspected by using the filtering mechanism. Thus, particular traffic can be investigated by extracting it without surrounding noise.

Anomalies are highlighted by featuring their abnormal distributions of traffic. For example, networks or port scans are monitored as vertical lines in scatter plots. Anomalous traffic is also emphasized through its unusual uses of source ports (see Figs. 3 and 7). Mostly linear patterns similar to those discussed in [4], can be identified from visualizations.

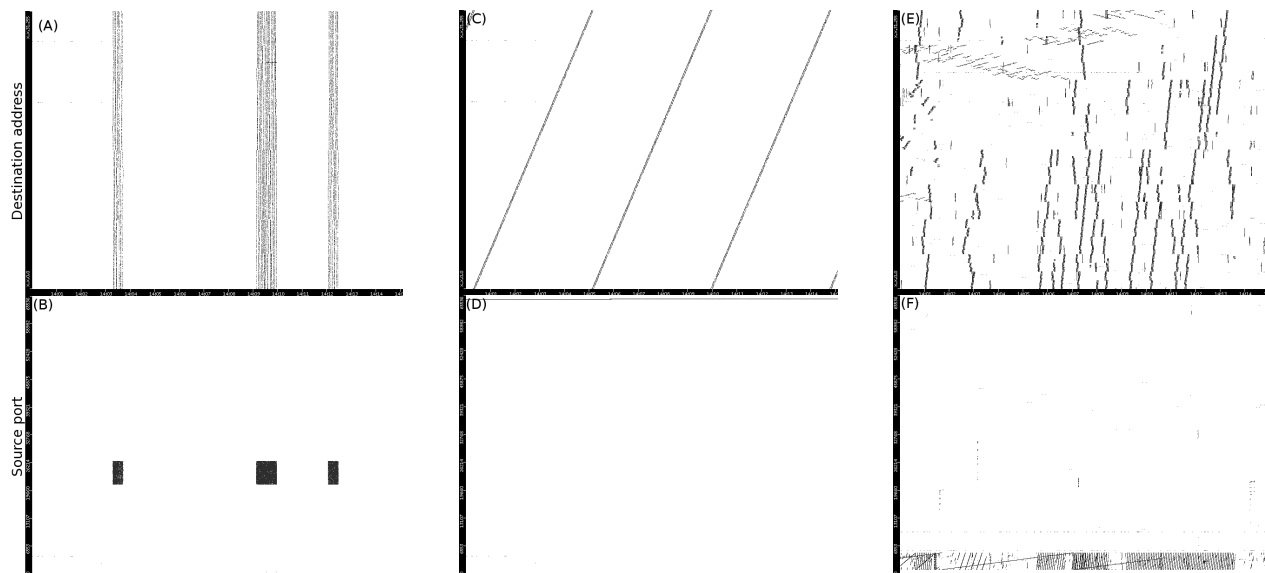


Fig. 7. Different patterns observed in the same traffic trace (MAWI Archive, 2004/10/14). Top: destination IP vs. time, Bottom: source port vs. time.

VI. CONCLUSION AND FUTURE WORK

First, we highlighted the need for identifying anomalies, and understanding network traffic behavior on all temporal and spatial scales. To achieve these purposes, we designed and implemented a tool that can graphically represent network traffic with five main features. (1) The tool displays traffic traces on different temporal/spatial scales. (2) In addition, different graphical representations are displayed on two panels to compare key features (e.g., destination addresses and source ports), and anomalies are highlighted with several patterns. (3) Detailed investigations are possible by easily navigating traffic traces, and (4) by checking the packet headers of all plotted points. (5) The tool reads pcap files directly and does not use an intermediate database. Furthermore, the tool can be executed on different platforms and is freely downloadable¹. Finally, we verified the usefulness of our tool by evaluating it on several traffic traces. A darknet trace highlighted several patterns for different anomalies, and a trace was taken from a backbone in which anomalies in heavy traffic were still highlighted. We proceeded to a manual inspection of anomalous traffic and introduced several patterns representing anomalies.

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. Also, the views of traffic provided by our tool would be useful for displaying results from anomaly detection methods, thus, facilitating the validation of those methods.

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," *LSAD '07*, pp. 145–152, 2007.
- [4] R. Fontugne, T. Hirotsu, and K. Fukuda, "An image processing approach to traffic anomaly detection," *AINTEC '08*, pp. 17–26, 2008.
- [5] T. Karagiannis, K. Papagiannaki, and M. Faloutsos, "Blinc: multilevel traffic classification in the dark," *SIGCOMM '05*, vol. 35, no. 4, pp. 229–240, 2005.
- [6] Tcpdump and libpcap, <http://www.tcpdump.org/>.
- [7] 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.
- [8] 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.
- [9] J.-P. van Riel and B. Irwin, "Inetvis, a visual tool for network telescope traffic analysis," *Afrigraph '06*, pp. 85–89, 2006.
- [10] S. Lau, "The spinning cube of potential doom," *Commun. ACM*, vol. 47, no. 6, pp. 25–26, 2004.
- [11] 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.
- [12] 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.
- [13] P. Ren, Y. Gao, Z. Li, Y. Chen, and B. Watson, "Idgraphs: Intrusion detection and analysis using histograms," *VizSEC '05*, 2005.
- [14] R. Marty, *Applied Security Visualization*, 1st ed. Addison-Wesley Professional, August 2008.
- [15] A. Inselberg, "The plane with parallel coordinates," *The Visual Computer*, vol. V1, no. 4, pp. 69–91, December 1985.
- [16] The c++ template image processing library. The CImg Library : <http://cimg.sourceforge.net>.
- [17] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson, "Characteristics of internet background radiation," *IMC '04*, pp. 27–40, 2004.
- [18] Mawi traffic archive. WIDE Project, <http://mawi.wide.ad.jp/mawi/>.
- [19] K. Cho, K. Mitsuya, and A. Kato, "Traffic data repository at the WIDE project," in *USENIX 2000 Annual Technical Conference: FREENIX Track*, Jun. 2000, pp. 263–270.
- [20] H. Ringberg, A. Soule, and J. Rexford, "Webclass: adding rigor to manual labeling of traffic anomalies," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 1, pp. 35–38, 2008.

¹The source code of the tool is available at <http://www.fukuda-lab.org/~romain>