A Visualization Framework for Traffic Data exploration and Scan Detection

Mai El-Shehaly
Faculty of Computers and Informatics
Suez Canal University
Egypt
Email: maya70@vt.edu

Denis Gracanin
Virginia Tech,
USA
Email: gracanin@vt.edu

Ayman Abdel-Hamid
Arab Academy
for Science and Technology,
Egypt

Kresimir Matkovic
VRvis Research Center,
Austria
Email: matkovic@vrvis.at

***Abstract*—Network packet traces, despite having a lot of noise, contain priceless information, especially for investigating security incidents. However, given the gigabytes of flow crossing a typical medium sized enterprise network every day, spotting malicious activity and analyzing trends in network behavior becomes a tedious task. Computational mechanisms for analyzing such data usually take substantial time to reach interesting patterns and often mislead the analyst into reaching false positives or false negatives. Therefore, the appropriate representation of network traffic data to the human user has been an issue of concern recently. Much of the focus, however, has been on visualizing TCP traffic alone while adapting visualization techniques for the fields that are relevant to this protocol's traffic, rather than on the multivariate nature of network security data, in general, and the fact that forensic analysis, in order to be fast and effective, has to take into consideration different parameters for each protocol. In this paper, we bring together two powerful tools: SiLK (System for Internet-Level Knowledge), for command-based network trace analysis; and ComVis, a generic information visualization tool. We integrate the powers of both tools by aiding simplified interaction between them, using a simple GUI, for the purpose of visualizing network traces, characterizing interesting patterns, and fingerprinting related activity. We applied the visualizations on anonymized packet traces from Lawrence Berkley National Laboratory, captured on selected hours across three months. We used a sliding window approach in visually examining traces for two transport-layer protocols: ICMP and UDP. The main contribution of this research is a protocol-specific framework of visualization for ICMP and UDP traffic data.**

I. Introduction

A broad overview of the network traffic analysis problem includes *data collection, storage and management, trend analysis,* feature detection, event characterization, and timely response to a particular incident [1]. **SiLK** [3], the system for Internet-Level Knowledge, is a collection of traffic analysis tools developed by the CERT Network Situational Awareness Team (CERT NetSA) to facilitate security analysis of large networks [2]. It supports the efficient *collection, storage,* and *analysis* of network flow data. The SiLK tool suite provides the ability to rapidly query large historical traffic data sets and, if properly installed on the backbone or border of a large, distributed enterprise or mid-sized ISP, SiLK gives great powers to the security analyst. However, these powers are faced with the limitations of the human analyst's ability to extract useful information from textual representations of SiLK's output. In addition, SiLK supports a large number of commands and flag combinations. This is a double sided coin, as much power as it adds to a typical analysis scenario, it can be very confusing to even an experienced user. In order to focus on the analytical reasoning task at hand, a security analyst should be spared from at least part of the hard coding tasks that SiLK requires. Therefore, flow records and other statistical information that are output from different SiLK tools need to be presented using visualization techniques that suit the multivariate nature of network traffic data and, hence, truly enhance the human's ability to identify trends, outliers and to recognize familiar fingerprints of well known attack tools. Information visualization can be of great importance to the field of network security because: *(1)* Many attack tools and their host operating systems, can be identified by their visual signatures. *(2)* Many visualization techniques can have fixed memory requirements despite the high network traffic volumes, which makes them more resistant to overload and resource consumption that can incapacitate the traditional IDS. *(3)* False positive and false negatives are significantly reduced.

ComVis [4] is a coordinated multiple views system that implements a number of interactive multivariate visualization techniques. Developed by the VRVis research team, ComVis is an attractive option for visualizing SiLK data for the following reasons: (i) the support for multiple views gives flexibility for exploring the multidimensional nature of network traffic data; (ii) brushing and linking capabilities cut a long trip short in detecting malicious activity and creating visual fingerprints for them; (iii) the support of parallel coordinates is an important feature in multivariate visualization tools; (iv) a data view allows close inspection of the details of a spotted pattern; and (v) the tool's interoperability with other applications, through its support of command line parameters and of a common data format, comma separated value (csv), made it a powerful choice for the seamless interaction with the SiLK tool suite.

This paper describes a framework of interaction between SiLK on one side, working as a background engine to query and analyze anonymized network traces from the Lawrence Berkley National Laboratory (LBNL), and ComVis on the other side as an information visualization interface. Section II gives an overview of background and related work. Section III discusses the interaction technique. Most importantly, Section IV describes a case study that tests the proposed

framework to aid the main tasks of a security analyst: trend analysis, event characterization, and scan fingerprinting. Finally, Section V concludes the paper. The main contribution of this research is a protocol-specific framework of visualization for ICMP and UDP data. The resulting views led us to a number of guidelines that can be vital in the creation of "smart books" describing best practices in using visualization and interaction techniques to maintain network security.

II. Related Work

Network intrusion and scan detection techniques can be categorized into *misuse detection*, which detect previously encountered attacks, through matching them with patterns of well-known attacks (e.g. IDIOT [14] and STAT [12]); and *anomaly detection systems* [23], which maintain a number of established normal usage profiles. Any significant deviation from these profiles is flagged as an anomaly, which raises an alert for a possible intrusion. Examples of anomaly based detection methods are IDES [17] and the architecture proposed by Zhang et. al. [24]. A major drawback of these algorithmic systems is that they rely solely upon machine-detected signatures and statistical anomalies for the purpose of spotting malicious activity. Each of these techniques has its own shortcomings when it comes to automatically detecting network intrusions without human interference. For example, misuse detection systems cannot detect new intrusions if their behavior does not match any of the known patterns, and anomaly detection systems can be trained slowly over time to overlook malicious activity. Another drawback of traditional IDS is that they have a binary status of the network, either there is intrusion alarm or not, they do not maintain state information of the network.

These shortcomings call for human interference to obtain accurate judgements that can resolve time-critical situations. However, maintaining network situational awareness for the human expert to aid such judgements is a necessary and arduous task. Efficient information visualization techniques are needed to display network traffic information in a way that maintains this awareness without misleading the human analyst. Some of the most famous network traffic visualization tools include NVisionIP [16], and VisFlowConnect [22], and FlowScan [21]. A problem with these tools is that they provide no flexibility in changing the displayed attributes. Also color encoding is of little relevance. Other problems include the fact that Important attributes are shown only for a single machine at a time, and trends in traffic volumes across time are not visualized.

Conti et al. used parallel coordinates to examine the visual fingerprints left by the attack tool in use in [5] and [6]. This provides insight into the attacker's methodology and aids law enforcement forensics Other tools for NetFlow data visualization and processing include NetFlow Sensor (NFSen) [11], Stager [18] a web based application for presenting most types of network statistics, the time-based Network traffic Visualizer (TNV) [8], BLINd Classification (BLINC) [13], and NFlowVIs developed by Fischer et al. [10]. More recent techniques include Secure Decisions' VIAssist [7], and the Dalhousie FloVis tools [9].

III. Interaction Methodology

The SiLK tool suite provides great flexibility in the collection, storage, and analysis of network flow data. This comes at the price of supporting a large number of command-line tools and numerous parameters for each. Memorizing the exact syntax of such commands and the required parameters for each becomes a tedious task for the analyst especially during time-critical forensic analysis procedures. An analyst investigating an attack incident for the purpose of preventing further damage to the local network, will have too little time to check the manual for the appropriate parameters to rwfilter, to obtain the relevant data set from the repository. Therefore, having an appropriate visual interface to a vital SiLK tool such as rwfilter, and linking its output directly to the visualization tool, helps the analyst focus on the analysis task at hand instead of spending their time and effort in memorizing hard-coded commands.

During the course of our analysis, we have come to an understanding of the most commonly used SiLK tools and their most often utilized parameters and we have included these in a simple, yet efficient, graphical user interface (GUI), that can be easily integrated in any information visualization tool. The advantages of this GUI are several, including the fact that the analyst is spared from most of the command-line confusion and from memorizing the order in which parameters need to be entered.

The GUI will walk the analyst through a series of steps to specify some of these parameters, acting only as a reminder for the order in which they need to be specified and sparing the analyst from memorizing the exact name of each flag (e.g. –proto rather than –protocol to specify the desired protocol number). The analyst still has to be aware of the format of SiLK commands in order to include the more advanced parameters in the analysis. In that case the GUI gives her the ability to review the generated SiLK command and edit it before it is passed to the SSH form and executed on the UNIX machine. Another advantage is that the GUI spares the analyst from switching between different platforms when the visualization tool is Windows-based, as is the case with ComVis. Figure 1 is inspired from [2], with the modification that it depicts our proposed framework instead of rwfilter's normal flow.

IV. Case Study

An enormous amount of traffic is captured daily on a typical enterprise network. The task of isolating productive traffic from background radiation [20] is a challenging one. We break this task down into three processes which we test ComVis' ability to perform: (i) trend analysis, (ii) event characterization, and (iii) fingerprint generation. The novelty of the proposed security visualization technique relies in the use of linked 2D time-based scatter plots along with the famous parallel coordinates. The greatest advantage of these

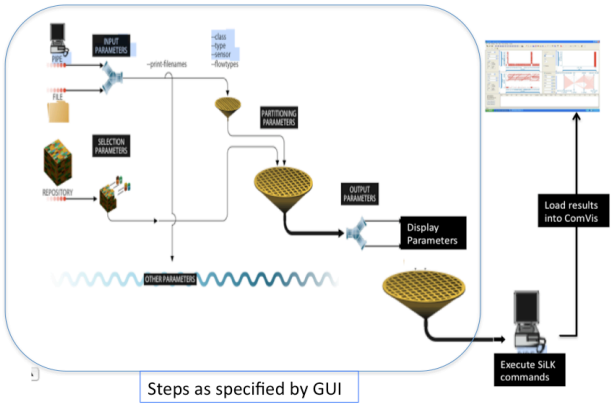

Fig. 1. The order of steps in which the GUI guides the user to interact with the underlying SiLK repository and to load data into ComVis. A tabbed window is used in which the first tab prompts for rwfilter's input and selection parameters, then partitioning parameters are entered using a second tab, followed by a third tab for display parameters used in rwcut command. The last tab lets the analyst edit the generated command and execute it.

TABLE I
PART OF SILK OUTPUT DESCRIBING TRAFFIC CAPTURED ON DECEMBER 15TH, 2004.

Date	Records	Bytes	Packets
2004/12/15T08:00:00	17971.30	121098598.67	237319.61
2004/12/15T09:00:00	18728.46	117956277.81	215969.81
2004/12/15T10:00:00	6621.93	30439031.58	70462.56
2004/12/15T11:00:00	10553.46	95623170.15	151840.06
2004/12/15T12:00:00	18881.72	30495363.68	114856.27

time-based 2D plots relies in their ability to narrate how different security parameters varied across time. A sample trend analysis scenario is described in section IV-A. Linking these variations to other parameters, further characterizes the nature of each security event as explained in section IV-B; while linking them to the parallel coordinates view leads to an identifiable visual fingerprint for the attack as shown in section IV-C.

A. Trend Analysis

An analysis session typically starts with an overview of trends in the data set; in order to give a contextual feel of which particular time slots witnessed relevant security incidents. When did traffic peaks occur? How long did they last? How much of these peaks was incoming traffic? how much of it came from sources in foreign countries? are some of the questions that need to be readily answered by such overview. We take a sample day from the available traces and examine ways to extract such information from the data set.

SiLK supports a number of aggregation tools that can provide statistical overview of a trace. The closest tool to answer the above questions is rwcount which calculates volumes over time samples. Table I displays sample results of an rwcount call with bin-size equals one hour for a day worth of traffic. Applying traditional visualization techniques on the information presented in Table I can improve its readability, but it still comes short in answering most of the questions sought by the specialist. Alternatively, we loaded all traffic encountered on that day in ComVis. Incoming traffic is selected using a brush, therefore, is displayed in red. The plot shows that the peak time slots witnessed greater variability in flow duration as shown in the top left scatterplot. Peak times are very clearly delineated in this particular view which conveys their exact start and end times. The top right view plots destination port on its y-axis. Horizontal lines represent a number of flows being received at a particular port consecutively. Brushing such patterns resulted in fan shapes on the parallel coordinates view. These are most likely to be scanning activity that went undetected by the data set providers who recorded these flows as non-scanning traffic [15]. The question of foreign countries contributing to incoming traffic can be calculated using SiLK's rwuniq tool; or, more efficiently, we can add a fifth view in ComVis that displays a histogram of source country codes. The brush automatically appears in the new histogram and our questions are immediately answered. Once such overview is conveyed to the analyst, zooming in on a smaller range of ports, IP addresses, or time frames, is a natural next step.

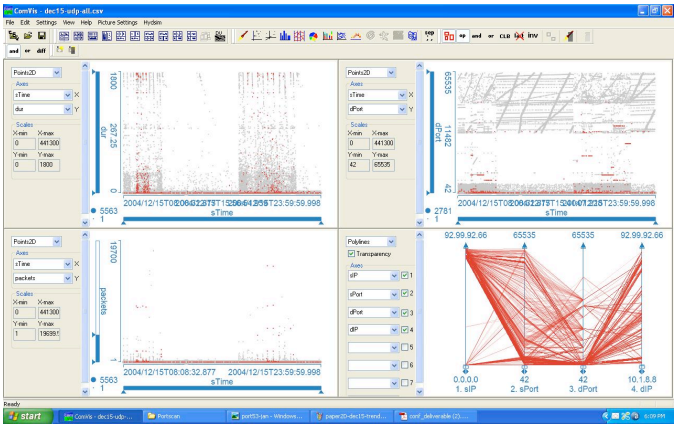

Fig. 2. ComVis visualizing SiLK flows captured on one day. A brush is used to highlight incoming traffic.

The output of the trend analysis stage is a relevant subset of data that the analyst loads in ComVis to zoom in on intriguing events. From there, the most important target for the visualization tool is to convey that an event indeed happened within the given time frame; and to provide as much characterizing details as possible about the event to aid forensic analysis.

B. Event Characterization

In this section, we test the ability of our visualization framework to provide visually identifiable patterns for different types of scans. Certain trends were found in the data sets containing scanning activity that makes use of UDP flows. For instance, several traces have shown great imbalance between inbound and outbound UDP traffic; with the amount of outgoing traffic sometimes exceeding double the amount of incoming traffic. This significant increase in outgoing traffic

should be considered an indication of a possible worm infection on the home network and must be investigated. Despite the suspicion raised by such trends, their absence in a data set does not rule out the possibility of major security incidents. Therefore, after an appropriate overview of trends is properly displayed, more focused detection methods are necessary to spot and isolate instances of malicious activity. In the LBNL data set at hand, Pang et al. [19] used a heuristic to isolate scanning behavior, by looking for hosts that visited more than 20 distinct IP addresses from which at least 16 were strictly in ascending or descending order. Of course this heuristic misses out all scanning action that targeted less than 20 hosts on the local network. In this section, we describe how visual patterns were found in the same traces that can be used to characterize and isolate different scans.

Despite the strengths of parallel coordinates in fingerprinting individual scanning tools as noted in [5], dealing with large amounts of trace records can greatly clutter the parallel view and scanning action becomes impossible to isolate from benign traffic. This is particularly where the brushing and linking capabilities of ComVis come handy, as we have found that port, and sometimes IP address, usage over time gives a less cluttered view of security events. Once such an event is spotted in a 2-dimensional view, a brush is used to link this finding to the parallel coordinates view in which a unique visual fingerprint is obtained. The generation of such fingerprints is covered in more detail in the next section. For now, we give a few sample findings in the data set that were readily obtained by inspection of the 2D time plots. Figure 3 depicts an example. A straight line is observed on the 2D plot with the source IP address at the y-axis and time at the x-axis. This conveys the fact that a single external IP address, issued a large number of flows within a small time frame, using four source ports and targeting one destination port 5000 on a range of internal hosts. The scan used single packet flows, each was 29 bytes long and had zero duration. This pattern was brushed using the 2D scatterplot and linked to the parallel coordinates plot, which was characteristic of this particular scan's behavior; it showed us that the suspected host targeted a large number of internal hosts hitting the same destination port on each, namely port 5000.

The above example showed how capable ComVis is of alerting the user that an incoming attack took place at a particular instance of time.

C. Scan Fingerprinting

1) ICMP Probes: Since ICMP traffic does not use any values for source and destination ports, the most relevant flow fields of interest, when fingerprinting ICMP probes, are ICMP type and code fields. All ICMP probes that originated from external hosts, reaching internal destinations, had a number of common characteristics that we summarize as follows:

- ICMP type field = 8.
- Percentage from total incoming ICMP traffic on that day is above 75%.

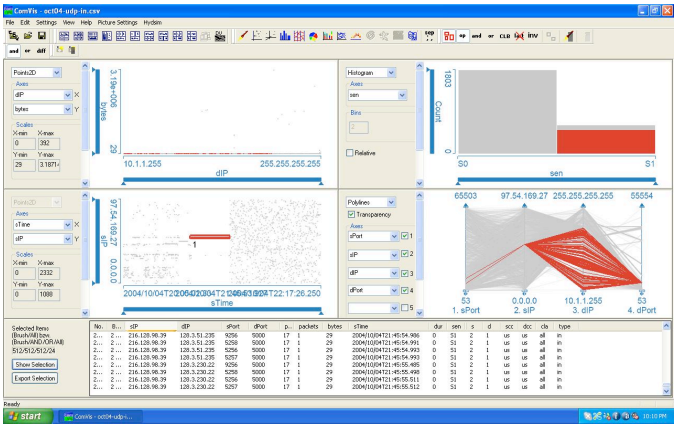

Fig. 3. Incoming UDP scan detected on October 4th. A straight line appears in the 2D plot (bottom left) which, when brushed, highlights a fan on the parallel plot (bottom right). The histogram (top right) is used to verify that no false positives are included and the 2D plot (top left) shows that all flows have the same byte length.

- Visual fingerprint on parallel plots appears as two fans coming from sources at opposite ends of the IP address range and targeting the same destination subrange.
- Byte length = 33 and 37 were used as a main selection criteria to rule out false positives. i.e. When selected, these values narrowed down the brushed records to only those involved in scanning activity.
- Source country codes include Korea and China as the major contributers at all times.
- Visual fingerprints on 2D plots using start time as the x-axis and source IP address at the y-axis appear very similar.

These similarities have led us to create a unified visual fingerprint among the ICMP daily traffic sets as depicted in Figure 4. We always included a parallel view, which was found to be the most characterizing fingerprint of ICMP probes, along with a histogram depicting the ICMP type of these probes, and two scatter plots (2-dimensional each) which plot the source IP's and destination IP's behavior across time.

It appears from the visual fingerprints that the sources and destination are the same. To test this hypothesis, we created three IP sets containing subnet masks sourcing these probes during months: October, December, and January. We assume that these monthly sets will contain a number of common subnet masks for the attacker IP's, assuming a 24-bit masking. The results show that there was a total of 633 distinct attacking subnets in the data set. Only 2 of which were common across all three months, 8 common subnets in October and December alone, 39 in December and January, and 14 in October and January. Despite these common values, it is quite obvious that the greater amount of values in each set were distinct. Therefore, the similarity of the visual fingerprints can be attributed to the type of scan and the tool in use rather than the use of particular IP addresses.

2) UDP PortScan: The Internet Assigned Numbers Authority (IANA) maintains official assignments of port numbers

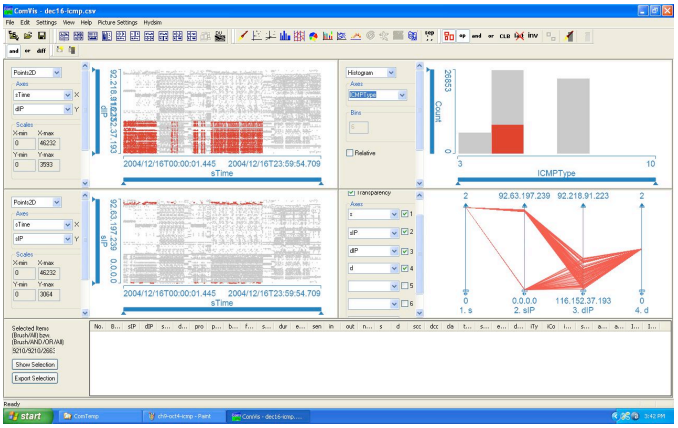

Fig. 4. Visual fingerprints of incoming ICMP probes in December 2004.

UDP port numbers for specific uses. Despite the fact that IANA does not enforce adherence to these assignments, port numbers can act as guiding clues about the involved activity in a captured trace. Table II lists the 10 most popular UDP ports that were targeted by incoming traces in our data set. The total number of flows targeted at such ports, their percentage, whether or not these flows involve scanning traffic (according to the sensor field), and the associated application or service for each port are all listed in Table II. Given these guidelines from SiLK results, we decided to take a look at each of these port's activity in ComVis to test the tool's ability to create visual fingerprints for activity on relevant ports.

Port 1037 on a single host (131.243.63.32) received the largest amount of traffic in the available traces. This port was not targeted by any flows in the 2004 traces. All of its incoming traffic was received during the three days of January 2005 that we have in our traces. Loading all incoming UDP traffic on these three days in ComVis and using a brush to select out port 1037 traffic, we obtained the view shown in Figure 5. Note the patterns in source IP usage across the three days. Two source ports were used in the attack (port 53 and port 9052) as seen in the parallel plot. examining each of the three days worth of traffic alone in ComVis resulted in the exact same fingerprint in the parallel plot. A reasonable explanation here can be the assumption that the attack has been issued by the same IP addresses, and targeted the same port on the same machine.

However, this assumption is rejected by creating IP sets from the subnet masks of the attacking sources (assuming /24 CIDR blocks), we found little similarities in the subnets used. For instance, 5902 subnets were found on January 6th, and 8486 subnets on January 7th. The intersection set between the two days constitutes only 55.8% of the january 6th set and 38.8% of the January 7th set. Given the variability of sources used, the visual fingerprint created by ComVis must be attributed to source and destination ports and the attacking tool's behavior in diversifying its sources across time.

Port 9875 witnessed longer lasting activity that spanned all three months. Visual snapshots were taken for each month

TABLE II
MOST COMMONLY TARGETED PORTS IN THE LBNL NETWORK.

Port No.	Flows	of Total	Scanning	Application
1037	33,869	49.3%	Yes	AMS
53	18,702	27.2%	Yes	DNS
9875	4,996	7.3%	No	unknown[1]
5000	2,624	3.8%	Yes	Sockets de Troie (Trojan)
123	2,145	3.1%	Both	Network Time Protocol [2]
42659	1,024	1.5%	No	
10002	596	0.9%	No	rscs2
10003	484	0.7%	No	rscs3
1027	341	0.5%	No	calendar access protocol
1026	336	0.5%	No	win-rpc, calendar access protocol

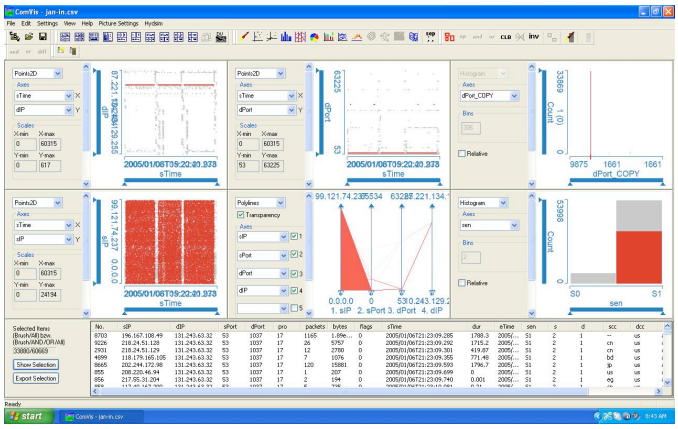

Fig. 5. Visual fingerprint of port 1037 activity.

separately. Figure **??** depicts the visual fingerprint of this port's activity during October 2004 (a), December 2004 (b), and January 2005 (c). Despite the variation of trends of traffic in the three time frames (see the histograms in the lower right) and the interleaving scanning action involving other ports on other hosts (top and bottom left views), the visual fingerprint of port 9875 activity remains the same in the parallel plot view, and source IP and port usage exhibit similar patterns in all three shots.

One may argue that the similar fingerprints may signify a set of attacks issued by the same hosts and therefore are depicted similarly by ComVis. To rule out this possibility, we created sets of sourcing IP addresses, using SiLK set tools, one set for each month's activity; and examined the amount of commonality among them. Results have shown that 100 distinct IP addresses issued the flows hitting port 9875 across the entire data set; only 31% of which were common across the three sets.

Other ports in our traces have exhibited similar fingerprints using ComVis visualizations as was the case with the above examples. For the sake of compactness, we only list a few ports here and include only the parallel plot view for each in Figure 6

V. CONCLUSION

The proposed approach contributes to the existing research efforts in visualizing network security traces in terms of the

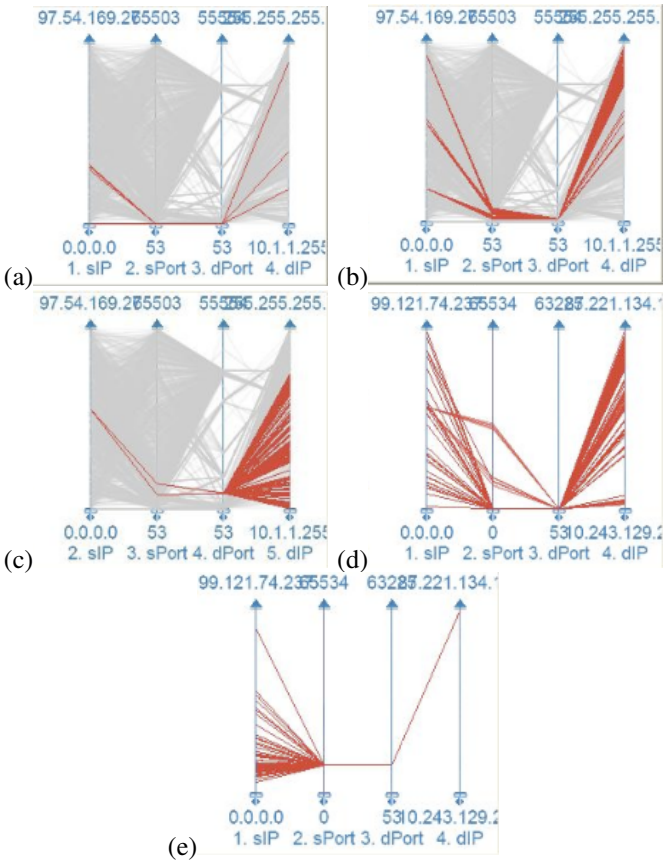

Fig. 6. Visual fingerprints of: (a)port 137, (b)port 1026, (c)port 5000, (d)123, and (e)10002. These fingerprints were unique for each port across the data set.

flexible integration of a variety of visualization techniques with parallel coordinates, which is so far considered to be the most effective technique in fingerprinting network security incidents. In addition, the back-end interconnection scheme between ComVis and SiLK provides support for querying the repository and obtaining fast analytical results. These advantages, combined with the ComVis features such as composite brushing, make the proposed approach and its implementation appealing to network forensic analysis. Future work will focus on using this approach on large resolution displays to take the full advantage of visual and analytical capabilities.

ACKNOWLEDGMENT

This work was supported in part by a grant from The Arab Academy for Science and Technology.

REFERENCES

[1] E. Bethel, S. Campbell, E. Dart, K. Stockinger, and K. Wu. Accelerating Network Traffic Analytics Using Query-Driven Visualization. *IEEE Symposium on Visual Analytics Science and Technology*, 2006.

[2] T. S. Micheal Collins, Andrew Kompanek. *Analysts Handbook: Using SiLK for Network Traffic Analysis*. CERT Network Situational Awareness Group, Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA 15213-3890, for silk version 0.10.3 edition, November 2006.

[3] CarnegieMellon. http://tools.netsa.cert.org/silk/.

[4] K. Matkovic, W. Freiler, D. Gracanin, and H. Hauser. ComVis: A Coordinated Multiple Views System for Prototyping New Visualization Technology. In *Information Visualisation, 2008. IV'08. 12th International Conference*, pages 215–220, 2008.

[5] G. Conti and K. Abdullah. Passive visual fingerprinting of network attack tools. In *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 45–54. ACM New York, NY, USA, 2004.

[6] K. Abdullah, C. Lee, G. Conti, and J. Copeland. Visualizing network data for intrusion detection. In *Systems, Man and Cybernetics (SMC) Information Assurance Workshop, 2005. Proceedings from the Sixth Annual IEEE*, pages 100–108, 2005.

[7] J.R. Goodall and D.R. Tesone. Visual Analytics for Network Flow Analysis. In *Proceedings of the 2009 Cybersecurity Applications & Technology Conference for Homeland Security-Volume 00*, pages 199–204. IEEE Computer Society, 2009.

[8] J. Goodall, W. Lutters, P. Rheingans, and A. Komlodi. Preserving the Big Picture: Visual Network Traffic Analysis with TN. *Visualization for Computer Security, IEEE Workshops on*, pages 6–6, 2005.

[9] T. Taylor, D. Paterson, J. Glanfield, C. Gates, S. Brooks, and J. McHugh FloVis: Flow Visualization System.

[10] F. Fischer, F. Mansmann, D. Keim, S. Pietzko, and M. Waldvogel. Large-Scale Network Monitoring for Visual Analysis of Attacks. In *Visualization for Computer Security: 5th International Workshop, Vizsec 2008, Cambridge, Ma, USA, September 15, 2008, Proceedings*, page 111. Springer, 2008.

[11] P. Haag. Nfsen: Netflow sensor. nfsen. sourceforge. net, 2008.

[12] K. Ilgun, R. Kemmerer, and P. Porras. State Transition Analysis: A Rule-Based Intrusion Detection Approach. *IEEE Transactions on Software Engineering*, pages 181–199, 1995.

[13] T. Karagiannis, K. Papagiannaki, and M. Faloutsos. BLINC: multilevel traffic classification in the dark. In *Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 229–240. ACM New York, NY, USA, 2005.

[14] S. Kumar and E. Spaord. A Software Architecture to support Misuse Intrusion Detection. In *National Information Systems Security'95 (18th) Proceedings: Making Security Real*. DIANE Publishing, 1996.

[15] L. B. N. Laboratory and ICSI. http://www.icir.org/enterprise-tracing/overview.html. Technical report.

[16] K. Lakkaraju, W. Yurcik, and A. Lee. NVisionIP: netflow visualizations of system state for security situational awareness. In *Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 65–72. ACM New York, NY, USA, 2004.

[17] T. Lunt, A. Tamaru, F. Gilham, R. Jagannathan, C. Jalali, H. Javitz, A. Valdes, P. Neumann, and T. Garvey. *A Real-time Intrusion-detection Expert System (IDES)*. SRI International, Computer Science Laboratory, 1992.

[18] A. Oslebo. Stager A Web Based Application for Presenting Network Statistics. In *Network Operations and Management Symposium, 2006. NOMS 2006. 10th IEEE/IFIP*, pages 1–15, 2006.

[19] R. Pang, M. Allman, V. Paxson, and J. Lee. The devil and packet trace anonymization. *ACM SIGCOMM Computer Communication Review*, 36(1):29–38, 2006.

[20] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson. Characteristics of internet background radiation. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 27–40. ACM New York, NY, USA, 2004.

[21] D. Plonka. FlowScan: A Network Traffic Flow Reporting and Visualization Tool, November 2000 University of Wisconsin-Madison http://net. doit. wisc. edu/plonka/lisa. *LISA 2000 Conference Proceedings, Dec*, 2000.

[22] X. Yin, W. Yurcik, M. Treaster, Y. Li, and K. Lakkaraju. VisFlowConnect: NetFlow Visualizations of Link Relations for Security Situational Awareness. In *Internet Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security (VizSEC/DMSEC-2004)*, 2004.

[23] Y. Zhang and W. Lee. Intrusion detection in wireless ad-hoc networks. In *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 275–283, New York, NY, USA, 2000. ACM.

[24] Y. Zhang, W. Lee, and Y. Huang. Intrusion Detection Techniques for Mobile Wireless Networks. *Wireless Networks*, 9(5):545–556, 2003.