A Survey of Visualization Systems for Network Security

Hadi Shiravi, Ali Shiravi, and Ali A. Ghorbani, *Member*, *IEEE*

Abstract—Security Visualization is a very young term. It expresses the idea that common visualization techniques have been designed for use cases that are not supportive of security-related data, demanding novel techniques fine tuned for the purpose of thorough analysis. Significant amount of work has been published in this area, but little work has been done to study this emerging visualization discipline. We offer a comprehensive review of network security visualization and provide a taxonomy in the form of five use-case classes encompassing nearly all recent works in this area. We outline the incorporated visualization techniques and data sources and provide an informative table to display our findings. From the analysis of these systems, we examine issues and concerns regarding network security visualization and provide guidelines and directions for future researchers and visual system developers.

Index Terms—Information visualization, network security visualization, visualization techniques.

1 Introduction

Although the visualization of network security events is the subject of this survey, this paper does not focus on designing and developing a specific visualization system. Instead, we consider network security with respect to information visualization and introduce a collection of use-case classes. In this study, we provide an overview of the increasing relevance of security visualization. We explore a novel classification approach and review the artifacts most commonly associated with security visualization systems. We provide a historical context for this emerging practice and outline its surrounding concerns while providing design guidelines for future developments.

Visual data analysis help to perceive patterns, trends, structures, and exceptions in even the most complex data sources. As the quantity of network audit traces produced each day grows exponentially, communicating with visuals allows for comprehension of these large quantities of data. Visualization allows the audience to identify concepts and relationships that they had not previously realized. Thereby, explicitly revealing properties and relationships inherent and implicit in the underlying data. Identifying patterns and anomalies enlightens the user, provides new knowledge and insight, and provokes further explorations. It is these fascinating capabilities that influence the use of information visualization for network security. Visualization is not only efficient but also very effective at communicating information [1]. A single graph or picture can potentially summarize a month's worth of intrusion alerts (depending on the type of network), possibly showing trends and exceptions, as opposed to scrolling through multiple pages of raw audit data with little sense of the underlying events.

Security Visualization is a very young term [2], [3]. It expresses the idea that common visualization techniques have been designed for use cases that are not supportive of security-related data, demanding novel techniques fine tuned for the purpose of thorough analysis. It may not always be possible to fully predict how an end user will perceive and interpret a design due to the varying nature of the audience's cognitive characteristics. Yet careful consideration of the user's needs, cognitive skills, and abilities can determine the appropriate content and design. Often associated with human-computer interaction, the philosophy of user-centered design places the end user at the center of the design process. Network security is a highly specialized and technical discipline and operation. It deals with packets and flows, intrusion detection and prevention systems, vulnerabilities, exploits, malware, honeypots, and risk management and threat mitigation. The complex, dynamic, and interdependent nature of network security demands extensive research during the development process. Without an in-depth understanding of security operations and extensive hands on experience, developing a security visualization system will not be possible. A design process centered on the needs, behaviors, and expectations of security analysts can greatly influence and impact the usability and practicality of such systems. For best results, security experts and visual designers must thereby collaborate to complement each other's skills and expertise to innovate informative, interactive, and exploratory systems that are technically accurate and aesthetically pleasing.

In this survey, we begin by looking into different categories of data sources incorporated in the design of security visualizations and provide an informative list of sources accessible to the research community. We continue in Section 3 by expressing our main contribution in the classification of network security visualization systems. We provide a detailed description of the proposed taxonomy

- *The authors are with the Information Security Centre of Excellence, Faculty of Computer Science, University of New Brunswick, 540 Windsor Street, Gillin Hall, Room E128 Fredericton, NB E3B 5A3, Canada.*
  E-mail: {hadi.shiravi, ali.shiravi, ghorbani}@unb.ca.

Manuscript received 30 Aug. 2010; revised 26 June 2011; accepted 12 Aug. 2011; published online 23 Aug. 2011.
Recommended for acceptance by K.-L. Ma.
For information on obtaining reprints of this article, please send e-mail to: tvcg@computer.org, and reference IEEECS Log Number TVCG-2010-08-0203.
Digital Object Identifier no. 10.1109/TVCG.2011.144.

1077-2626/12/$31.00 © 2012 IEEE Published by the IEEE Computer Society

TABLE 1
Potential Data Sources for Security Visualizations

Event Type	Data Source	Devices
Network Traces	–Raw Packets	Tcpdump, Tshark
	–Netflow Records	Cisco NefFlow NDE, Cisco NSEL Netflow
Security Events	–Intrusion Detection Systems	Cisco CSA, Cisco IDS, Enterasys Dragon, Fortinet Fortigate,
		Juniper ISG, SNORT, Niksun NetVCR, SourceFire Intrusion Sensor
	–Intrusion Prevention Systems	ForeScout ConterACT, Juniper NetScreen IDP, McAfee Intrushield,
		Radware Defense Pro, FireEye, Tipping Point X, IPAngel
	–Firewalls	Check Point, Linux Iptables, PaloAlto PA, Cisco ACE
	–Virtual Private Networks	ArraySP, Nortel VPN Gateway, Checkpoint VPN-1, Cisco ASA
	–Anti-virus	Mcafee, Sophos, Symantec, Trend Micro
Network Activity Context	–Layer 7 application context	Q1 Labs QFlow, Foundry SFlow, Juniper JFlow, Packeteer FDR
User/Asset Context	–Vulnerability Scanners	NMap, eEye REM, Nessus, Rapid7 NeXpose, SecureScout
		nCircle IP360, Patchlink Scan, Qualys, Saint
	–Identity and Access Management	Microsoft ForeFront Identity Manager, Identity Forge
		Quest Identity Manager One, EmpowerID
Network Events	–Switches	Cisco CatOS, Cisco Catalyst, 3Com 8800 Series
	–Routers	Cisco Routers, Enterasys Router, Juniper Router, Nortel Router
	–Servers	Apache, BlueCoat SG, Cisco Ironport, IIS, Sun Sendmail
	–Hosts	Windows, Linux, Solaris, IBM AIX RACF, HP Tandem
Application Logs	–Application Databases	IBM DB2, SQL Server, Oracle, Imperva SecureSphere, Sybase
	–Workflow	
	–Enterprise Resource Planning	
	–Management Platforms	

together with an analysis of the derived use-case classes. We follow by giving a thorough description of each system as we outline its strengths and weaknesses. An overall assessment of systems in each use-case class in addition to guidelines and directions for future systems is also provided. We summarize the multiple attributes of recent network security visualization systems in a table for better future references. We continue in Section 4 by outlining issues and concerns surrounding security visualization by elaborating on seven potential pitfalls. We conclude this research in Section 5 by summarizing our findings.

Papers studied in this survey were selected based on the following metrics:

1. **Relevance to network security:** As the title of the paper indicates, this study focuses specifically on network security visualization systems. Visualizations of code security, binary files, or visual cryptanalysis are subjects that could span another volume of similar size and are thereby not considered in this study.
2. **Contribution of system and visual techniques:** Due to the chronological study of papers, systems that have utilized a specific visualization technique or method with highly similar characteristics to those of previous systems have not been selected for this survey. Similarly, visualization systems that lack contextual, perceptive, and cognitive considerations are also not considered.
3. **Satisfactoriness of evaluation:** Although most systems surveyed in this paper lack formal evaluation, yet many have been validated through ad hoc use-case attack scenarios. Systems that lack even this basic validation strategy are also not considered in this survey.

We believe these three metrics impact the quantity and quality of papers surveyed in this work to resemble systems that are focused explicitly on network security, are novel in their incorporated visual techniques, and are validated on at least a use-case scenario. Systems that do not adhere to these metrics are thereby not considered in this study.

2 Data Sources

Visualization cannot happen without data or information. Many of the systems surveyed in this paper have been created based on a single source of data. Looking at network events from multiple perspectives by incorporating different data sources into a system can provide an analyst with a richer insight into the underlying events. Therefore, a nonexhaustive list of potential data sources that are available to the research community and may be incorporated in the design of network security visualization systems is given in Table 1. The decision on the type and number of incorporated data sources and the set of extracted features from each data source is a critical act. The data sources mentioned in Table 1 are very generic and in some cases, e.g., network traces, hundreds of features can be extracted from them. The importance of selecting the appropriate features, as a first step in designing a visualization system, has been extensively studied in the fields of statistics, pattern recognition, machine learning, and data mining and the resulting efforts have been applied to the fields of artificial intelligence, text categorization, and also intrusion detection. These studies are of great benefit to security visualization researchers as often the required steps of selecting an optimal subset of features (subset generation, subset evaluation, stopping criterion, and result validation) have been examined extensively before. Based on a particular problem a researcher is facing and the data sources available to him or her, a subset of features may be extracted and incrementally validated until a desired optimality is achieved.

3 Classification Approach

The approach taken in many visualization systems is data driven. In network security for instance, one may take a single data source like packet traces and try to develop a visualization system based on that. The methodology behind the design of visualization systems should be use-case driven. A visualization system should be built to support answering specific questions. In this approach, the system may incorporate one or multiple data sources.

Based on this mindset, we have classified the recent works of network security visualization into five use-case classes. We provide a detailed description of each class, discuss several recent examples of each approach, specify the incorporated visualization techniques of each system, and challenge the applicability of each use-case class in regard to modern day networks. Guidelines for future research, and directions for informative and efficient visuals are also provided for each use-case class at the end of each section.

3.1 Host/Server Monitoring

In this class of visualization, the main display is devoted to the representation of hosts and servers. The intent is to display the current state of a network by visualizing the number of users, system load, status, and unusual or unexpected host or server activities. Systems of this class should also be able to correlate communicating processes of a single host or server with the network traffic. This feature enhances the ability of a user to identify malware as they often manifest themselves in irregular and often anonymous system processes.

The work of Erbacher et al. [4], [5] constitutes one of the earlier works in this class. As illustrated in Fig. 1, hosts are arranged around five concentric circles with the monitored server placed in the center. The ring of a node depicts the difference between its IP address and that of the monitored system, resulting in hosts residing inside the local subnet to appear closest to the monitored system. The position of a host on the circular ring is also recorded to ensure that a specific host always appears in the same position. Multiple visual attributes are assigned to each node as they are depicted using glyphs. For the monitored server, for example, spokes extending from its perimeter represent the number of connected users. As connections are made from hosts to the monitored server and based on the connection type, communication links are shown with different line patterns. These visual illustrations give an analyst an exploratory framework to work with as it strengthens her abilities to detect unknown relationships within the underlying data.

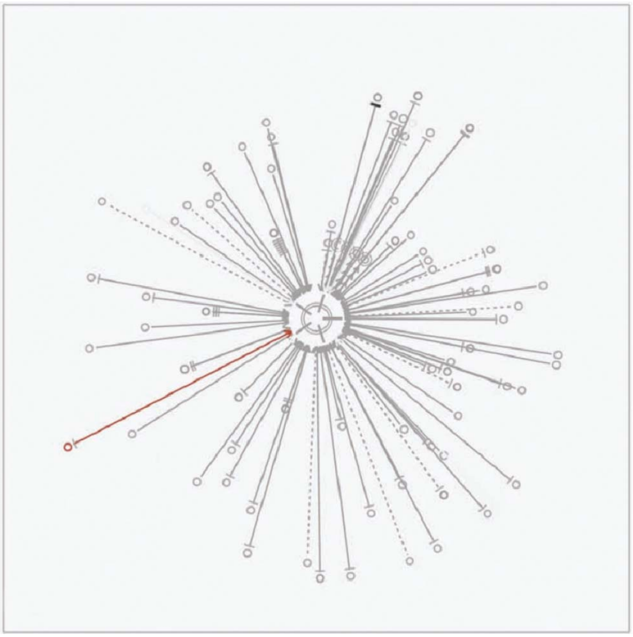
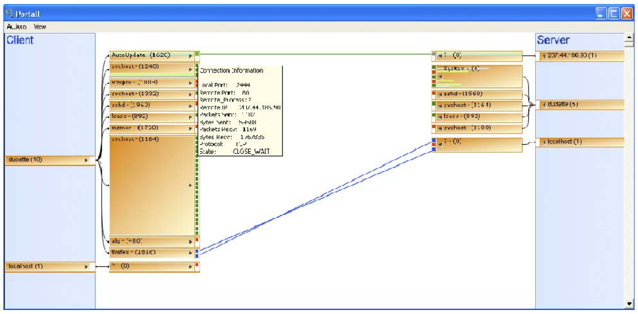


Fig. 1. Basic visual representation of network and system activity in [4].

Tudumi [6] is also one of the earlier systems belonging to this category aimed at monitoring and auditing user behavior on a server. In a 3D visualization, Tudumi visualizes connections using lines and system nodes using 3D glyphs as they are displayed on multilayered concentric disks. Similar to Erbacher's system, Tudumi uses line patterns to encode different access methods including coarse dashed lines to represent a terminal service and thin dashed lines to represent file transfer.

The previous two systems are more concerned with the activities of a single or a limited number of hosts or servers rather than incorporating a larger portion of the network. NVisionIP [7], [8] takes on a different approach. It represents an entire class B IP network on a single 256 × 256 matrix grid with each cell of the matrix representing interactions between the corresponding network hosts. In the galaxy view of the system, all network subnets are listed along the horizontal axis while hosts of each subnet are listed along the vertical axis. As the number of visualized elements increases, inevitably the portion of the screen allocated to each object decreases. NVisionIP uses a magnifier function to allow the user to hover over the display screen. If an analyst is interested in a particular part of the display, she can select it using the magnifier function. A bar graph is then displayed for each host, depicting their activity over common and uncommon port numbers.

Portall [9] digs deeper into the monitored hosts and tends to correlate TCP connections with the host processes that generate them, allowing an end-to-end visualization of communications between distributed processes. As displayed in Fig. 2, the main display consists of two parallel axes with the left side representing clients and the right side representing servers and their respective processes. A line is drawn from a client to a server to depict a TCP connection. Portall is one of the first systems that visually correlates



Connection Information

Local Port: 2444
Remote Port: 80
Remote_Process: ?
Remote IP: 217.44.185.40
Packets Sent: 1102
Bytes Sent: 59508
Packets Recv: 1169
Bytes Recv: 176835
Protocol: TCP
State: CLOSE_WAIT

Fig. 2. A screen shot of Portall [9] with monitored hosts and servers stacked on the left and right sides.

network traffic to host processes, allowing spywares and ad-wares to be easily detected.

Similar in nature to Portall is the Host Network (HoNe) [10] visualization system that also visualizes communicating processes of a host with network traffic. The authors argue that the reason behind not being able to correlate processes to network traffic is inherent in the design of the TCP/IP networking model of modern operating systems. The system visualizes client side hosts and their respective processes and port numbers on the left side of the display while external sources and their respective port numbers are displayed on the right side. Different to Portall, HoNe uses splines rather than simple straight lines to connect processes of a client to external servers.

Perlman and Rheingans [11] extend existing approaches in host/server monitoring by adding and encoding service and temporal information inside the visualized node itself. Each host inside the network is illustrated using a circular glyph node much like a pie chart. Each glyph represents the existence and amount of activity for a particular service. The size of the glyph represents the total amount of activity of the node, measured by the number of packets. Wedge sizes identify what percentage of the total activity belongs to a particular service. A collection of different colors is used to distinguish between different services of a host. The system also incorporates time by using a stacked pie chart approach where the most outer ring represents the most recent time slice. Hosts are laid out in a simple node-link layout with straight lines connecting the communicating hosts together.

The Radial Traffic Analyzer [12] visualizes the distribution of network traffic of a particular host using a radial representation. The system is composed of four concentric circles, each mapped to an attribute of the underlying data. In its default setting, the innermost ring is assigned to source IP addresses, the second ring to destination IP addresses, and the third and fourth rings are mapped to source and destination port numbers, respectively. The notion of assigning port numbers to services and applications, devised in this system, is no longer accurate in modern networks as many applications tend to piggyback or tunnel through common port numbers such as HTTP (80) and HTTPS (443).

The work of Mansmann et al. [13] is one of the recent works in this class. In their proposed visual analytics tool, by incorporating a force directed graph layout, host behavior is monitored and irregular positional changes are flagged as suspicious. The authors believe that change in network traffic over time is well suited for detecting uncommon system behaviors. As illustrated in Fig. 3, in the first step of the visualization, a set of dimension nodes, each representing a network service are laid out using a circular force directed layout. In the second step, the observation nodes representing a particular host are placed on the display and are connected to their corresponding dimension nodes through virtual springs. Node size is calculated based on the sum of transferred bytes using a logarithmic scale. Since the state of a monitored host is displayed through multiple time stamps and due to the large number of visualized elements on the display, depicting multiple hosts without overlap is a challenge for this visual.

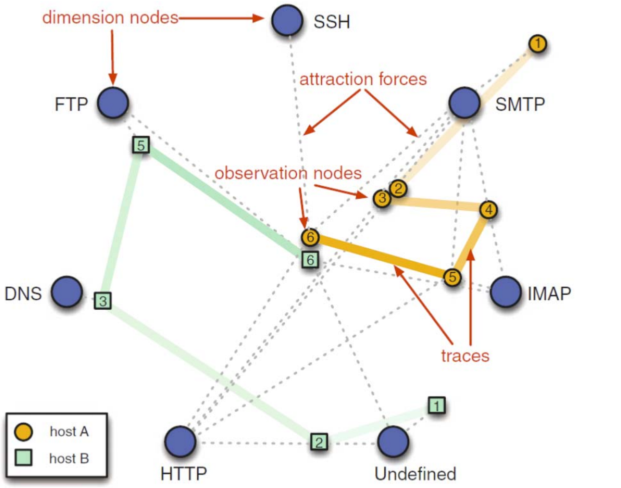


Fig. 3. A sketch showing the coordinate calculation of a host position at a particular point of time as depicted in [13].

Overall assessment of the host/server monitoring class. The ability of the visualization systems of this class in displaying a restrained number of hosts or servers within the monitored network is a perceptible issue. Most, if not all, of the systems of this class are constrained by their incorporated visualization techniques. As networks tend to grow in size and complexity at an exponential rate, there is an unprecedented need to create meaningful contexts. Even the smallest of university campus networks can consist of thousand of hosts, with which the aforementioned systems are less than capable of displaying in a clear and perceivable manner. For an analyst, simpler graphics are easier to understand and interpret than complex ones, since complexity can often influence the ability of the viewer in perceiving and decoding a visual. The overwhelming number of hosts in a monitored network, accompanied by the many hundreds of events generated for each, and the complexity of relations between events limit the cognitive process of situation awareness for analysts. For visualizations of this class to be effective and to clearly convey meaning, it is essential for them to devise an automated process that prioritizes situations and projects critical events. If a visualization system, due to its incorporated visualization technique, is limited in displaying a comprehensible range of hosts, envisioning a situation assessment process is inevitable. In this case, the process of identifying hosts with anomalous behavior and the mechanism of correlating

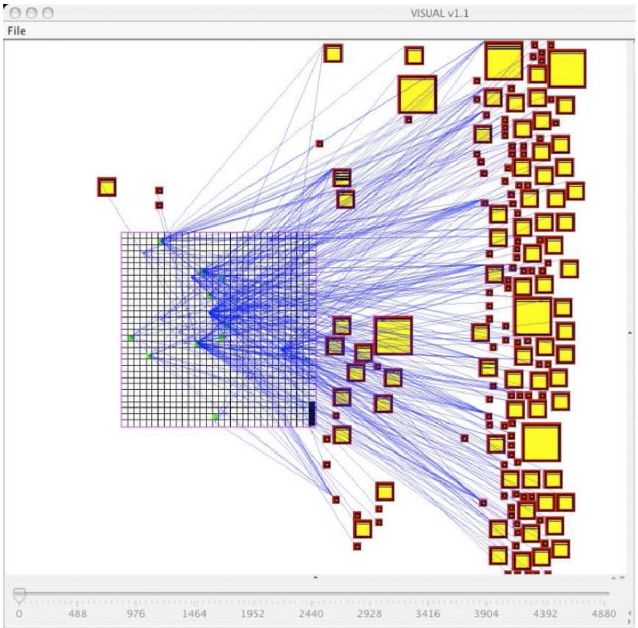


Fig. 4. VISUAL [14] displaying 80 hours of network data on a network of 1,020 hosts.

events is partly undertaken in a separate background component, and the processed results are projected to the visualization system. In this manner, the load on the visualization system is reduced considerably; allowing for a near real time analysis of events and a more responsive system. Packet traces, server logs, and network flows constitute primary data sources for this class of visualizations. Node link graphs, glyphs, and scatter plots are also primary visualization techniques incorporated in this class.

3.2 Internal/External Monitoring

Visualizations of this class are concerned with the interaction of internal hosts with respect to external IPs. Similar to the above-mentioned class, this class of visualization also incorporates a display of internal hosts, but in relation to communicating external IPs. Since the art of displaying internal hosts in a nonoccluding and meaningful manner is by itself a delicate act, adding the burden of displaying hundreds and thousands of external IPs is a nontrivial process for systems of this class.

VISUAL [14] is one of the earliest systems of this class. It is a security visualization system with the goal of allowing an analyst to see communication patterns between an internal network in regard to external sources. As displayed in Fig. 4, the internal network is represented by a grid with each cell depicting one of the internal hosts. External sources are represented as squares outside the internal grid with the square size denoting the level of activity. Simple straight lines are used to represent a connection between internal and external hosts. Multiple filtering mechanisms can be used to filter out internal or external hosts leading to a less cluttered display. Various detailed information regarding a host can also be displayed upon user request.

VizFlowConnect [15] uses parallel axes to display network traffic between internal and external hosts. The goal of the system is to display relationships between communicating machines of a network. The main display consists of

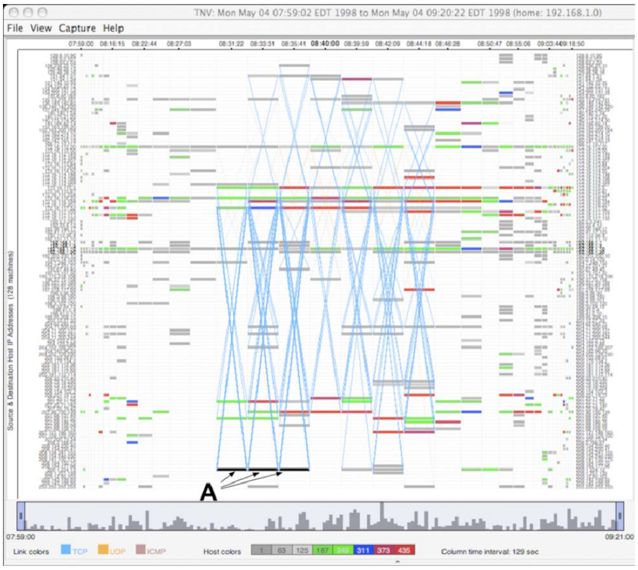

TNV: Mon May 04 07:59:02 EDT 1998 to Mon May 04 09:20:22 EDT 1998 (home: 192.168.1.0)
File View Capture Help
Source & Destination Host IP Addresses (128 machines)
A
07:59:00
09:21:00
Link colors
TCP
UDP
ICMP
Host colors
1
63
125
187
249
311
373
435
Column time interval: 129 sec

Fig. 5. TNV [17] showing 50,000 network packets in a 90 minute time span.

three distinct parallel axes. The center axis represents internal hosts. The left axis corresponds to machines originating network traffic to the internal network while the right axis represents the destination machines of internal traffic. Each point on an axis represents an IP address and connections between points on parallel axes represent network communication. Time is incorporated in the system by using animation and various multiple views allow for further exploratory analysis. VizFlowConnect also shows individual host statistics, but further drill down depth is desired.

Erbacher et al. [16] have come up with a second visualization system; this time aimed at internal/external host monitoring and geared toward filtering unwanted data, allowing focus on more critical events. The visual system incorporates a radial panel design consisting of multiple concentric disks each showing a constant period of time. Local IP addresses are placed around the radial disks while remote hosts are located on the top and bottom of the display. In order to avoid overlapping lines, an IP address located on the top half of the circle is connected to remote hosts located along the top of the display while hosts located on the bottom half of the circle are connected to remote hosts located on the bottom of the display. In the same manner, port numbers are also allocated on the left and right sides. The outer ring of the display shows the most recent period with interior rings displaying previous periods. This feature allows an analyst to see trends and patterns within the communicating hosts. Hosts are identified by dots on the circular rings resulting in difficult user interaction.

In a visual network traffic analysis system, TNV [17], Goodall et al. believe that analysts often lose sight of the big picture while examining low-level details of attacks. In order to prevent this loss of context, they propose TNV with the goal of providing a focused view on packet level data in the high-level network traffic context. As illustrated in Fig. 5, the main visual component of TNV is a matrix displaying network activity of hosts over time, with connections between hosts overlaid on the matrix. TNV is designed based on a focus and context paradigm where the center of the display, the focal area, shows communicating hosts within wider columns. In order to preserve continuity throughout the display, the context area, located to the left and right sides of the display, has gradually decreasing width. Each host inside the matrix is colored according to its level of activity and multiple linked views are used to illustrate port activity and details of raw packets. TNV is one of the few security visualization systems that has been fully implemented and is freely available for download.

Overall assessment of the internal/external monitoring class. Similar to the recommendations mentioned for the host/server monitoring class, the visualizations of this class can greatly benefit from a situation assessment component. This component can be defined in two different styles. One, as a process that automatically identifies and evaluates the impact of underlying events and relates them to assets of the monitored network or two, as an exploratory system that provides the facility for an analyst to validate various hypotheses. In the first style, due to the processing of events in a background component, the visual component can focus better toward richer and more responsive user interfaces. A necessity that is lacking and often overlooked in security visualization systems. In the second style, it is the analyst's job to pose queries, correlate disparate events, and derive insightful meanings from the visualization. These activities impose the need for visual exploration and filtering mechanisms to be implemented. Dynamic queries, details on demand techniques, and linking and brushing interaction techniques are essential concepts that need to be addressed and considered in this class of visualizations. Color maps, radial panels, scatter plots, and parallel coordinates are common visual techniques used in this class. Packet traces and network flows are also used as the main data sources for visualizations of this class.

3.3 Port Activity

Designers of this class of visualization argue that various malicious programs like viruses, Trojans, and worms manifest themselves through unusual and irregular port activity. Visualizations of this class can aid in the detection of malicious software running inside a network. Scaling techniques must be incorporated in the design of visualizations of this class, due to the amount of traffic as well as the large range of possible port numbers and IP addresses.

One of the earlier visualization systems designed specifically for this class is the work of Abdullah et al. [18]. In their developed system, a port-based overview of network activity is presented through stacked histograms of aggregated port activity. The authors believe that port activity can be used to detect zero-day exploits that are not detectable by conventional methods. As displayed in Fig. 6, port numbers are aggregated into multiple groups based on the services provided in the network. Well-known ports ($<1{,}024$) are assigned to major services on a system making them more vulnerable to attacks. For this reason, they are placed into bins of 100's, registered ports ($<50{,}000$) are placed into bins of 10,000's, and the remaining private/dynamic ports (50,000-65,535) are placed into a single bin. Color and scaling methods are also used effectively to distinguish between the aggregated port groups. In their developed system, the user has the ability to drill down in order to view finer details of the visualization. Displaying data over time also helps to highlight any patterns or trends appearing in irregular activities. The visualization is intuitive, easy to work with, and meets its intended design goals.

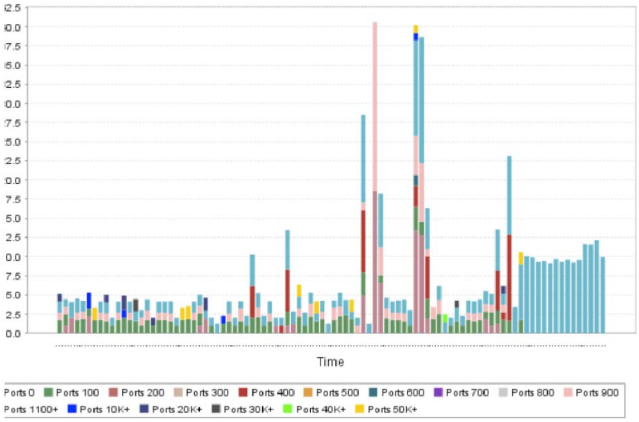
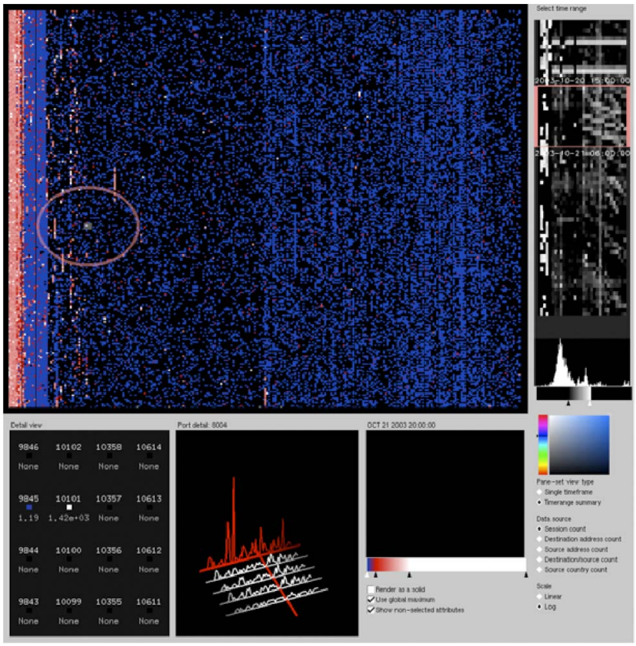


Fig. 6. Botnet traffic capture displayed using a cube root scale histogram in [18].

The Spinning Cube of Potential Doom [19] is an interesting example of security visualization. A system that visualizes real-time port and IP data in a three dimensional cube, displayed as a rotating scatter plot. Each axis of the 3D display represents a component of a TCP connection. Destination IP addresses are mapped to the X-axis, port numbers to the Y-axis, and source IP addresses to the Z-axis. TCP connections are displayed as individual dots with color used to distinguish a successful connection from an unsuccessful one. Time is displayed through the use of animation. While quite useful to see coarse trends in large-scale networks, it lacks drill down mechanisms, multiple views, and interactive capabilities. The system is good for solo attacks and can only be used for port scan detection.

PortVis [20] employs a colored-based grid visualization to map network activity to cells of a grid. As depicted in Fig. 7, the main display contains a 256×256 grid where each point represents one of the possible 65,536 port numbers. The location of a port on the gird is determined by breaking the port number into a 2-byte (X,Y) location. X being the high byte of the port number and Y being the low byte. Changes and variations of each point, with respect to time, is depicted using color. Black portrays no variation or change, blue depicts a small level of variance, red refers to a larger level of variance, while white denotes the most variant. The grid can be magnified to provide further detailed information about specific ports. A drawback of the system is seen when a port with suspicious activity is located among a collection of ports with a high, legitimate, level of activity. In this case, the ability to identify and focus on that region is not an easy task.

NetBytes Viewer [21] allows a detailed inspection of the behavior of an individual host over time. It facilitates in identifying behavioral changes that manifest themselves as unusual port usage or traffic volume regarding a single host. NetBytes offers multiple views in both two and three dimensions, making it possible for an analyst to view the



□ Render as a solid
☑ Use global maximum
☑ Show non-selected attributes

Pane-set view type
○ Single timeframe
● Timerange summary

Data source
● Session count
○ Destination address count
○ Source address count
○ Destination/source count
○ Source country count

Scale
○ Linear
● Log

Fig. 7. A screen shot of the Portvis [20] application with the main visualization displaying a magnification square.

data from different perspectives. It achieves this by displaying the traffic per port on an hourly basis over a user specified period. The initial screen displays the port activity of a single host by mapping the port list along the Z-axis, time on the X-axis, and the magnitude of traffic on the Y-axis.

Since the number of communicating ports of a single host is far too many to be visualized using conventional methods, Janies proposes existence plots [22], a low-resolution time series plot for port behavior analysis. Existence plots provide a high-level summary of traffic for a particular host. The vertical axis of the plot represents the entire port range in log scale while the horizontal axis represents time. Drastic changes in magnitude are measured over earlier time periods and are denoted as different colors. The system is evaluated in identifying hidden servers that provide services to hosts outside the network.

Overall assessment of the port activity class. In modern day networks, applications have not only become increasingly evasive, they have also become the predominate target of todays threat developers. In this regard, applications tend to adjust how they communicate over the Internet to circumvent traditional firewalls. Port hopping, use of nonstandard ports, tunneling, and encryption constitute the main tactics incorporated in applications. Due to this evolving nature, almost two-thirds of all enterprise traffic is currently routed through ports 80 (HTTP) and 443 (HTTPS) [23]. This change in behavior greatly influences and alters the objectives of security visualizations of this class. Rather than allocating a limited visual space to a larger range of port numbers, visualizations of this class should consider depicting the in-depth activity of a limited but predominant number of ports in a network. In this context, systems of this class should step away from the traditional port scan attack detection, a trivial task that is automated using a handful of signatures, and should focus greatly on detecting abnormal behaviors of different applications running simultaneously on a single port. In this manner, visualizations would not suffer from scalability issues and can be used more effectively in tasks that are not already automated. Histograms and scatter plots are the two prominent visual techniques of this class. Similar to the previous class, packet traces and network flows account for the main data sources.

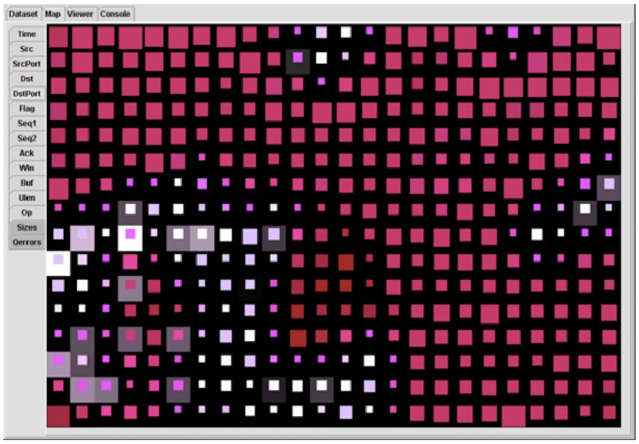


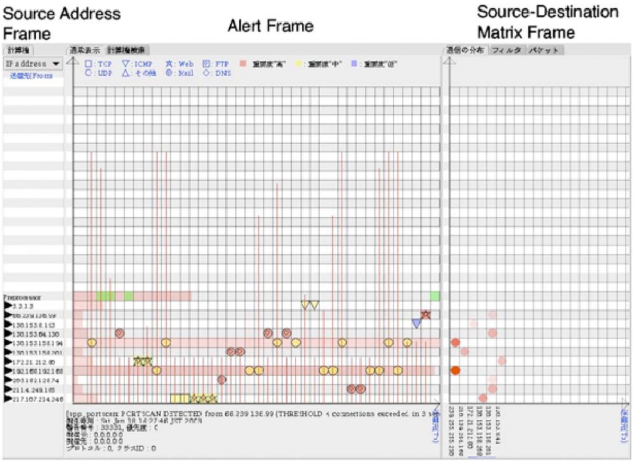
Fig. 8. A SOM colored by different attributes (panel on the left) depicting network activity as seen in [29].

3.4 Attack Patterns

Visualizations of this class aid an administrator in not only the detection of attacks but also the display of multistep attacks. Different types of attacks show different behaviors and accordingly different visual patterns appear. Many types of attacks are carried out in multiple phases, generally starting with reconnaissance, followed by scanning, acquiring access, maintaining access, and finally clearing tracks and installing back doors for future access. Visualizations of this class should aid in displaying these phases.

Intrusion detection alerts constitute one of the common data sources of this class. A major drawback of IDSs, regardless of their detection mechanism, is the overwhelming number of alerts they generate on a daily basis that can easily exhaust security administrators [24], [25], [26]. Additionally, since IDS signatures are not always written precisely enough or are written too specific and because deviation from normal behavior does not always correspond to malicious behavior, the phenomenon of false positives and false negatives also arises [26], [27], [28]. Traversing through multiple pages of log entries, in search for malicious and anomalous behavior is a nontrivial task for a security administrator. Visual systems should assist in the visual correlation and reduction of false positives.

Girardin [29] is one of the first in academia to develop a visualization system aimed at detecting anomalous and intrusive behaviors inside a network. By incorporating a self-organizing map to reduce the dimensionality of the network space, Girardin maps the output of the machine learning algorithm to a two dimensional color map representative of the state of the network. As illustrated in Fig. 8, the system uses foreground and background colors, size, and relative positioning to display both the network state and the deviance from normal behavior. The map is



[spp_portscan PORTSCAN DETECTED from 66.239.136.99 (THRESHOLD 4 connections exceeded in 3 se

Fig. 9. A Snapshot of the main display of SnortView [31].

organized by similarity of events resulting in similar events to be grouped together. The proposed method is unsupervised, which means no prior knowledge of network data is required, but the final design is abstract in nature and far from intuitive.

NIVA [30] is an early example of an intrusion detection data visualizer. NIVA uses data from various intrusion detectors and incorporates links and colors to signify attacks. The application consists of a rendering window and a graphical user interface window. The 3D rendering window consists of glyphs connected by links. The location of each glyph is based on its IP address such that closer glyphs resemble closer IP addresses, possibly on the same subnet. The link color represents the severity of attacks. Yellow is moderate, while red is the most crucial. Node positions are calculated using gravitational theory, electromagnetics, and fluid dynamics. The authors conclude that the system is able to display millions of nodes, but fails when representing a month of alert data.

SnortView's [31] primary purpose is to use visualization to effectively recognize false positives. As depicted in Fig. 9, the visualization consists of three main panels: the source address, alert, and source destination. The source address panel, located on the left side of the display, sorts and lists IP addresses vertically. The vertical axis of the alert frame represents a list of source IP addresses while the horizontal axis displays time. Each IDS alert is displayed using a colored glyph inside the alert panel. Each color displays a different priority: red being the highest and blue being the lowest. Different shapes (squares, circles, stars, and, etc.) are used to represent different types of alerts regarding different services. A detailed window, located at the bottom of the display, provides further information regarding an alert. The system is limited in representing a large number of IP addresses and is therefore suitable for smaller networks.

Unlike previous visualization systems of this class in which identifying trends and exceptions in IDS related data were the target, IDGraphs [32] takes on a different approach. It tends to identify intrusive behavior by incorporating flow level traces plotted with time on the horizontal axis and aggregated number of unsuccessful connections on the vertical axis. As illustrated in Fig. 10, the visualization technique utilizes Histographs: a visual technique to map brightness of a pixel to data frequency. By mapping multiple feature combinations of the input data, attacks with different characteristics are able to be identified. IDGraphs not only displays an overview of the underlying data but also allows for an in-depth analysis of identifying potential anomalies through dynamic querying.

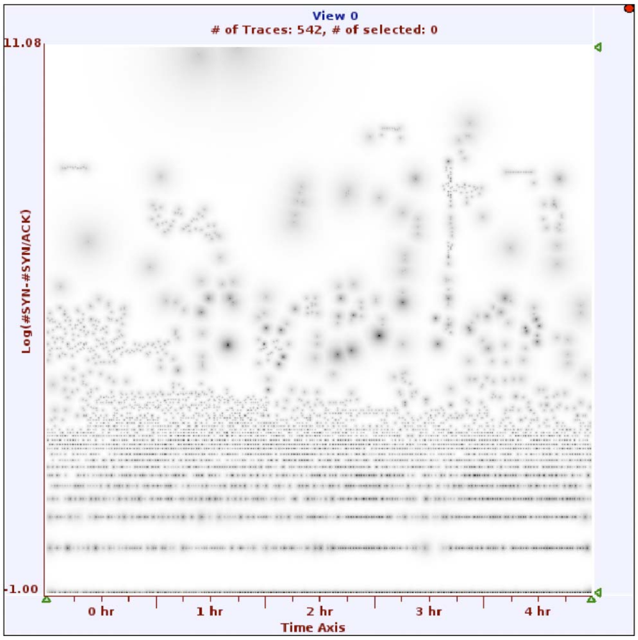
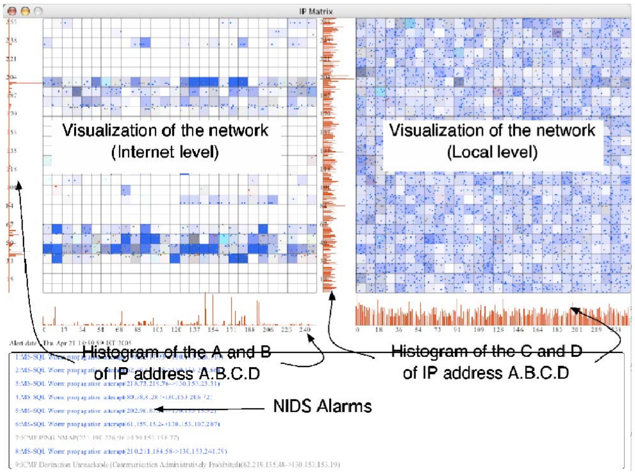


Fig. 10. A screen shot of IDGraphs [32] displaying Netflow streams plotted on a Histograph.

The authors of IP Matrix [33] believe that an attacker's IP address, although spoofed, is a significant factor in an attack and administrators can take appropriate countermeasures based on it. Subject to this belief, IP Matrix is geared toward representing the entire IP range. As illustrated in Fig. 11, IP Matrix incorporates two 256×256 matrices. The first, the Internet level matrix, only maps the first two octets $(a.b.*.*)$ while the local level matrix maps the last two octets $(*.*.c.d)$, allowing local and Internet level IP addresses to be seen at the same time. Each alert generated by an IDS is mapped using a pixel inside its appropriate cell. Pixels are color coded to



Alert date: Thu Apr 21 13:59:59 IST 2005
1:MS-SQL Worm propagation attempt([illegible])
2:MS-SQL Worm propagation attempt([illegible])
3:MS-SQL Worm propagation attempt(218.73.219.76->130.153.23.31)
4:MS-SQL Worm propagation attempt(80.58.3.28->130.153.206.72)
5:MS-SQL Worm propagation attempt(202.96.87.[illegible]->130.153.15.92)
6:MS-SQL Worm propagation attempt(61.155.15.2->130.153.107.207)
7:ICMP PING NMAP(221.190.226.96->130.153.158.77)
8:MS-SQL Worm propagation attempt(210.211.184.58->130.153.241.75)
9:ICMP Destination Unreachable (Communication Administratively Prohibited)(62.219.135.48->130.153.153.19)

Fig. 11. A snapshot of IP Matrix [33]. The left pane is the Internet-level IP Matrix, and the right pane is the local-level IP Matrix.

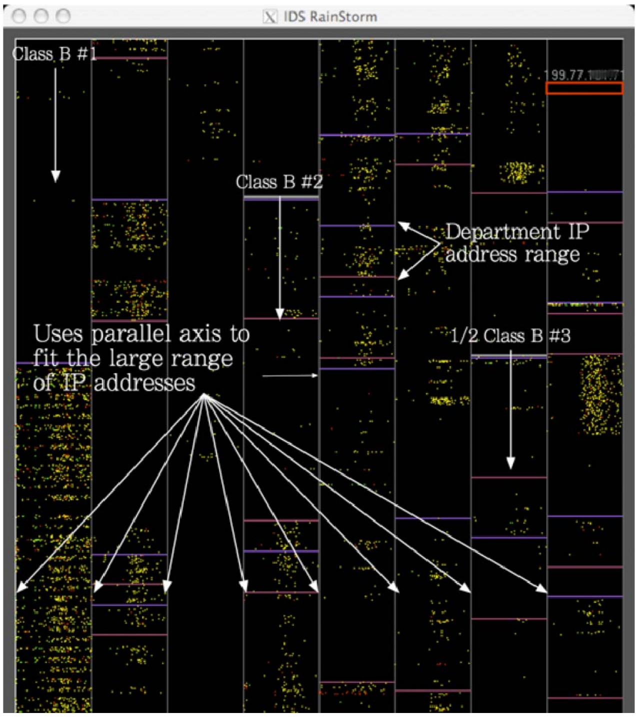

IDS RainStorm
Class B #1
Class B #2
Department IP address range
Uses parallel axis to fit the large range of IP addresses
1/2 Class B #3
99.77.

Fig. 12. IDS RainStorm [35] main view composed of eight vertical axes representing a 2.5 Class B IP range.

represent attacks of a different nature, but since a pixel is too small to be perceived, a cell's background is colored with the most frequent attack type. A disadvantage of this system is that there are no connections between the internal and Internet level hosts, making the system less intuitive. The system demonstrates its ability to detect virus propagations by visualizing the Welhia and Sasser.D viruses.

Visual Firewall [34] visualizes firewall operations, IDS alerts, and overall network statistics through multiple views on a single screen. The Real-Time Traffic view is composed of two parallel axes: one representing the firewall and the other the external hosts. The view shows packets, illustrated as glyphs, flowing between the firewall and external hosts in an interactive manner. Motion is used to depict direction and acceptance of traffic by the firewall. The Statistics View illustrates the networks overall throughput using colored histograms: purple for summary, red for incoming, and green for outgoing throughput. The IDS Alert View is displayed using a quad axis diagram. The left axis lists different alert types. The right axis represents external IPs generating the alerts. The bottom axis represents a 24 hour time period while the top axis represents internal hosts of the network. Alerts are displayed using color coded lines, and line transparency is used to represent time. The authors believe that Visual Firewall can aid in the fine tuning process of firewall rules and consequently lead to the detection of anomalous activities.

Abdullah et al. [35] have developed a novel visualization system, IDS RainStorm, to address the problem of flourishing number of alerts generated from intrusion detection systems in large networks. They argue that manually traversing textual logs is not only a frustrating and time consuming task, but also may result in important details being overlooked. They insist that the use of information visualization in network security would assist the analyst in gaining new insights and would aid in identifying patterns of anomalous network behavior. As illustrated in Fig. 12, the developed system consists of a main view which displays an overall representation of the network and a zoom view that provides a detailed display of a user selected range of IP addresses. The main view is depicted in eight columns each showing, in a top to bottom manner, a contiguous set of IP addresses such that 20 addresses are allocated to a single row of pixels. Using this method allows for a 2.5 class B IP address block to be represented onto a single display screen for a full 24 hour period. Alerts are also represented as color coded pixels such that red displays high, yellow medium, and green low-severity levels. An analyst can gain an overall perception of network activity by analyzing the main view, and if suspicious of any particular alert pattern, she can then use the zoom view to gain greater insight. The designers of IDS RainStorm evaluate their system by showing various usage scenarios, each carefully chosen to display the diverse set of activities that can occur, internally or externally, against a large network. These include abnormal network usage, worm propagation, and botnet activity.

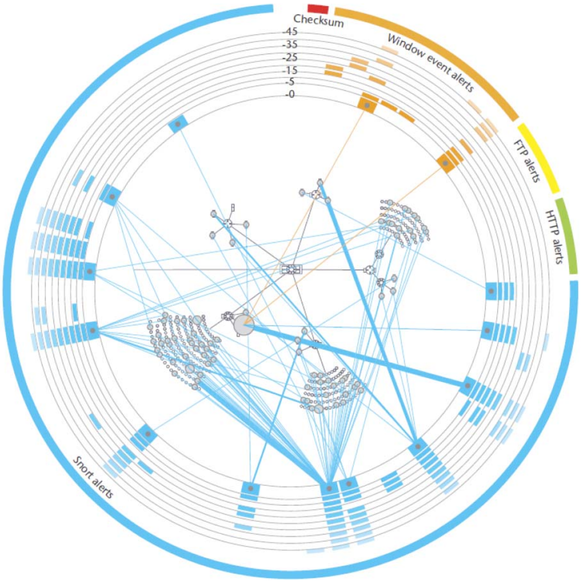


Fig. 13. Vizalert [36]: a novel visualization paradigm for network intrusion detection.

Livnat et al. [36], [37], [38] have developed a novel paradigm for visual correlation of network alerts. Their approach is based on the notion that an alert must possess three attributes namely: what, when, and where and that these attributes can be used as a basis for comparing heterogeneous events. As illustrated in Fig. 13, the developed paradigm displays the local network topology map in the center with the various alert logs on a surrounding ring. The ring's width represents time and is divided into several history periods. A line is drawn from an alert type on the outer ring to a particular host on the topology map to represent a triggered alarm. Thicker lines show a higher number of alerts of a single type, and larger nodes in the topology map represent hosts experiencing unique alerts.

Conti et al. [39], [40] have developed Rumint, a system composed of seven different visualizations, with the goal of

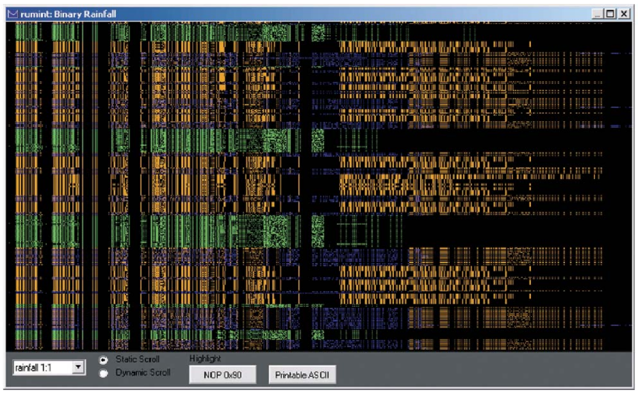


Fig. 14. Binary rainfall [39] illuminating pixels in a one-to-one correspondence to packet bits.

viewing a large number of network packets in a way that supports rapid comparison and efficient analysis. The designers of Rumint believe that unlike many previous systems that focus only on one perspective of the problem, multiple semantic windows on network traffic will enhance the knowledge and insight of the analyst. In spite of the fact that Rumint has the intention of displaying network traffic in multiple displays, only the parallel coordinate plot and the binary rainfall display seem more practical than others. The parallel coordinated-plot visualization displays scaled values from packet header fields. This tool combined with various filtering and querying methods can be used for a great in-depth analysis of an event. As illustrated in Fig. 14, the binary rainfall visualization displays packet contents, one per line. It has three primary views which map packet contents to display pixels. It has the potential of displaying up to 1,000 packets for rapid comparison of data. Unlike many visualization systems, Rumint is very different in the techniques it applies. It aims at using novel methods in displaying network data, but in many cases, the visualizations are only suitable for a particular kind of attack and would not be applicable for an analyst's everyday use. However, it seems more appropriate for forensic analysis of past events, where one has enough time to utilize the dynamic query and filtering techniques of various visualizations to come up with a firm final resolution.

One of the few visualization systems geared toward the identification and detection of DNS related attacks is the work of Ren et al. [41]. The authors believe that since the DNS protocol has not been developed with security concerns in mind it is prone and vulnerable to multiple attacks such as reflection, amplification, and cache poisoning. The system proposes a methodology to identify, detect, classify, and analyze abnormal DNS querying behaviors. The visual metaphor, Flying Term, is introduced in this system and is used to visualize target objects (query strings, IP addresses, port numbers) inside a rectangular area, where the spatial location of objects is computed based on the frequency of occurrence within a moving time frame.

Xiao et al. [42] propose a visual analysis system that incorporates a declarative knowledge representation language based on first order logic. The authors believe that a specific pattern identified by an analyst through a visualization system can be transformed into a logical representation and can later be used to identify and label similar sequences. In the first step of this process, an analyst, via a visualization system, identifies a pattern of interest. In the second step, using information stored in the knowledge base, she will then try to model the underlying pattern by iteratively adding, removing, and modifying predicates to form a logical clause. In the final step, the system incorporates these logical clauses into its knowledge base so that they can be used in the future to label similar types of events. To capture the versatile nature of network traffic, systems of this kind require an extensive and expressive set of predicates in their knowledge base.

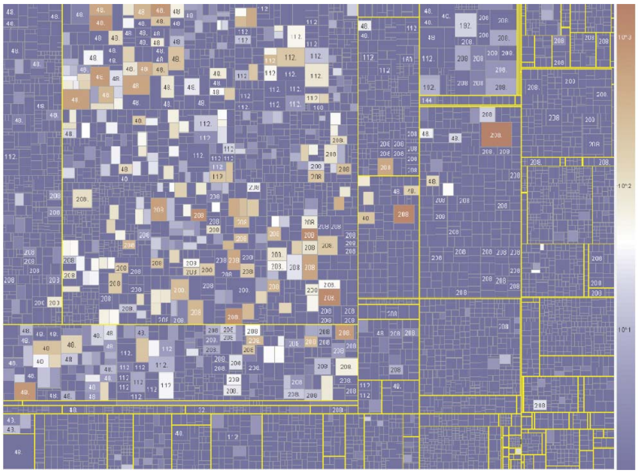


Fig. 15. Rapid spread of botnet computers in China in August 2006 as seen from the perspective of a large service provider as illustrated in [44].

Svision [43] aims at clustering internal and external hosts of a network by incorporating a 3D space defined by a set of network services. A host is attracted to a service point based on the amount of service usage in a predefined time window. Since a host may use multiple services on a network, an attraction force formula is defined to measure the distance of a host from a service point. The visual system also utilizes the third dimension to represent a hosts' load in a specified period. The system is validated at identifying a variety of network attacks (portsweep, UDP storm, worm propagation, Mailbomb, and ICMP flood) as they are depicted on the main display.

Mansmann et al. [44] extend the concept of Treemap layouts by proposing an interactive Hierarchical Network Maps visualization. As depicted in Fig. 15, the hierarchical structure of IP prefixes is utilized to allow for visualization of large scale network data aggregated by prefix, autonomous system, country, and continent. The applicability of various layout algorithms are assessed at different hierarchical levels, and several case studies demonstrate the fitness of the visual system. Although the visualization system is very well implemented, due to the high-level nature of the system, its application to network monitoring may seem greater than network security.

The ability to correlate security alerts with specific network applications and users is a necessity for security analysts. SpiralView [45], a visualization system built on top of a detection engine, is geared toward achieving this goal. The main display is composed of a spiral visualization that depicts security events based on their time of appearance in a radial fashion. The spiral display itself is composed of multiple concentric circles, one for each day, with the outer most ring displaying the most recent time period. Alerts, based on their triggered time, are displayed using circle elements where color indicates alert category and size depicts severity. The user/application view of the system correlates individual or group of alerts with specific applications and users. This extremely useful feature provides the analyst with an exploratory correlation system that can be utilized to identify hosts running malicious software inside a network. The use of dynamic queries, interactive visual components, and animated zooming provides further exploratory features to the user.

NFlowVis [46] analyzes NetFlow data using a relational database system. It is built with the intention of providing a quick visual insight of the communication patterns. A Treemap visualization is incorporated to map the monitored network. Source IPs of external machines are arranged at the borders and linked using splines parameterized with prefix information. The tool can be used to assess the relevance of alerts, to reveal large distributed attacks, and to analyze service usage within a network.

Yelizarov and Gamayunov [47] have designed a 3D visualization system aimed at illustrating multistep attacks. In a 3D environment, each event is represented using a cylinder glyph with the height of the cylinder indicating severity level and the color displaying type of attack. The system can be viewed from multiple angles and perspectives with the Cartesian option more perceivable than others. In order to accommodate external hosts inside the design, a subsidiary axis has been added with lines connecting them to the respected cylinder glyphs. The visualization, in its current status, is difficult to perceive and not intuitive for the average user.

ENAVis [48] is a graph-based visualization geared toward visualizing network activities that involves hosts, users, and applications. Although little contribution is seen from the visualization perspective of the work, greater emphasis is put on graph construction and metrics. The system views network activity form three different perspectives. The intergraph cluster visualization is used to measure and depict similarities or changes between two moderately sized graphs, e.g., two days of network activity. The intragraph cluster visualization moves a level down and identifies similarities and group structures within nodes of a single graph. The node similarity visualization looks into the dynamics or similarities of an individual node in a graph, providing insight to the dynamic behavior of hosts, users, and applications.

Shiravi et al. [49] propose Avisa, a security visualization system that embraces a situation assessment component. As illustrated in Fig. 16, the system assigns scores to hosts based on a collection of metrics that reflect change in a variety of aspects related to the alerts received by a particular host in a monitored network. The system utilizes three categories of heuristic functions, each composed of multiple heuristic measures, to collectively identify hosts with peculiar, irregular, and variant behaviors. The top n hosts are arranged along a radial display, while multiple statistics are mapped to various visual attributes. The intrusion alerts regarding each host are depicted using Beta-Splines, starting form the alert type category on the top left to the specific host on the bottom right. To reduce visual clutter and enhance the detection of patterns, multiple control points are incorporated for custom clustering of edges.

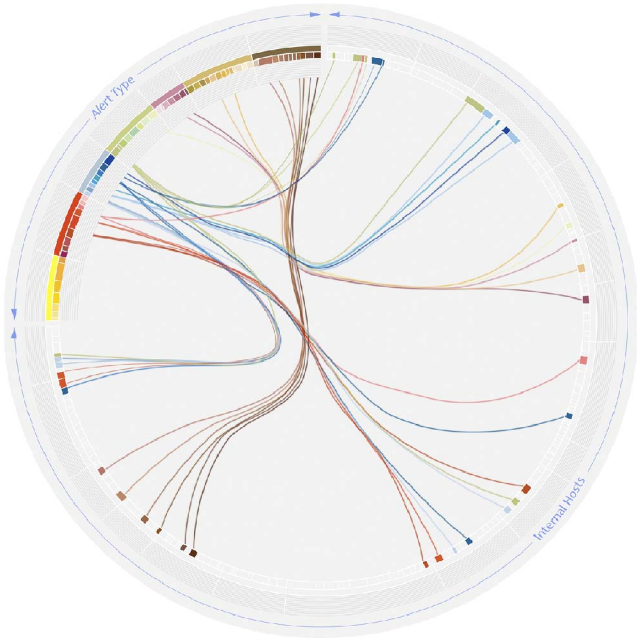


Fig. 16. Radial design of Avisa [49] depicting prioritized hosts and alert categories.

Overall assessment of the attack patterns class. This class of visualization has seen greater attention from the research community due to the fascinating ability of visualization in providing insight into the attack detection process. A visualization system devised for the process of detecting attack patterns is most likely a complementary tool to the already comprehensive line of network security products. Visualizations of this class should be thought as systems that provide insight into areas that other security systems fail to enlighten. As previously mentioned, visualization allows for inherent attributes of a data set, not previously anticipated, to be detected. This feature should be considered as the main contribution of a security visualization system. Any anomalous behavior detected should then be analyzed and automated, if possible, so that an automated application can handle the task in future, conserving human time and attention. Common visual techniques of this class include scatter plots, glyphs, color maps, and parallel coordinates. Systems may also incorporate several techniques in multiple views to allow for a more thorough data analysis. Inner related views provide an analyst with different perspectives of the same data, facilitating in identifying abnormal activities.

3.5 Routing Behavior

Understanding the evolution of Border Gateway Protocol (BGP) routing patterns over time is the main goal of this visualization class. The distributed nature of BGP and the lack of verification of the validity of the announcements causes Internet routing to be susceptible to attacks. The ability to detect and correct disruptions in Internet traffic

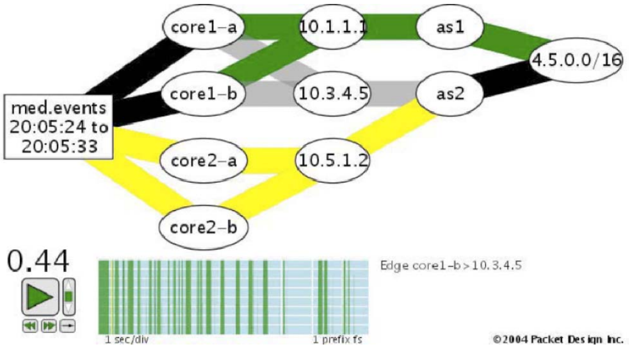


Fig. 17. A snapshot of a TAMP animation as illustrated in [51].

caused by router misconfigurations or malicious attacks is considered in this class.

The fluctuating nature of Internet routing has led recent works of this class to focus on providing a high-level view of interactions between autonomous systems. BGPlay [50] allows Internet Service Providers to monitor or observe the reachability of a specified prefix from the perspective of a given border router while incorporating animation to highlight routing changes.

Wong et al. [51] describe in their proposed system, TAMP, statistical methods to aggregate BGP data in order to visualize and diagnose BGP anomalies. As depicted in Fig. 17, the system incorporates animation to show how the routing behavior changes over time, aiding network administrators in identifying irregular and anomalous patterns. The LinkRank visualization [52] illustrated in Fig. 18, provides a high-level view of Internet routing changes. The authors believe that since minor connectivity changes can result in thousands of BGP messages, the system should be able to highlight important routing changes and facilitate in conducting root cause analysis.

In all three systems, since the overlaying logic is of higher importance than the visualization itself, simpler visualization techniques such as node link graphs have been used. Autonomous systems are displayed as nodes while connectivity is illustrated using links. Preattentive attributes such as size, color, and width are used to encode further information.

The work of Teoh et al. has also been focused on BGP routing but from a somewhat lower level perspective. Teoh et al. provide a collection of visualization tools [53], [54], [55], [56] for examining individual update messages with the aim of problem diagnosis and root-cause analysis.

Table 2 summarizes the materials covered in this survey regarding security visualization systems. For each system, its name, use-case class, incorporated visualization techniques, data sources, and number of citations it has received (as of June 24, 2011) are considered. The table accounts for the best practices of this emerging field.

4 Issues, Concerns, and Future Guidelines Regarding Network Security Visualization

Security visualization is concerned with the use of perception to recognize and amplify cognitive activities. If data are represented in a way that does not comply with the inherent rules of the human visual system, the underlying patterns and trends of the data would not be detectable. As previously mentioned, since the developers of security visualization tools are not necessarily visual designers, their proposed tools may lack a few fundamental features. Below we have outlined a number of these issues:

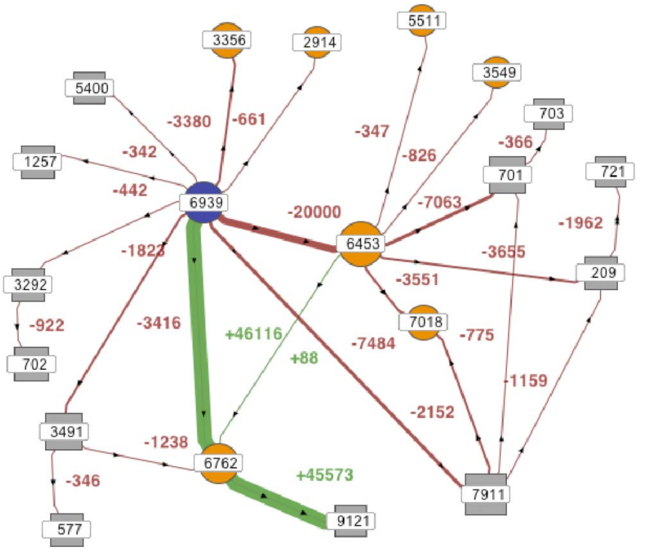
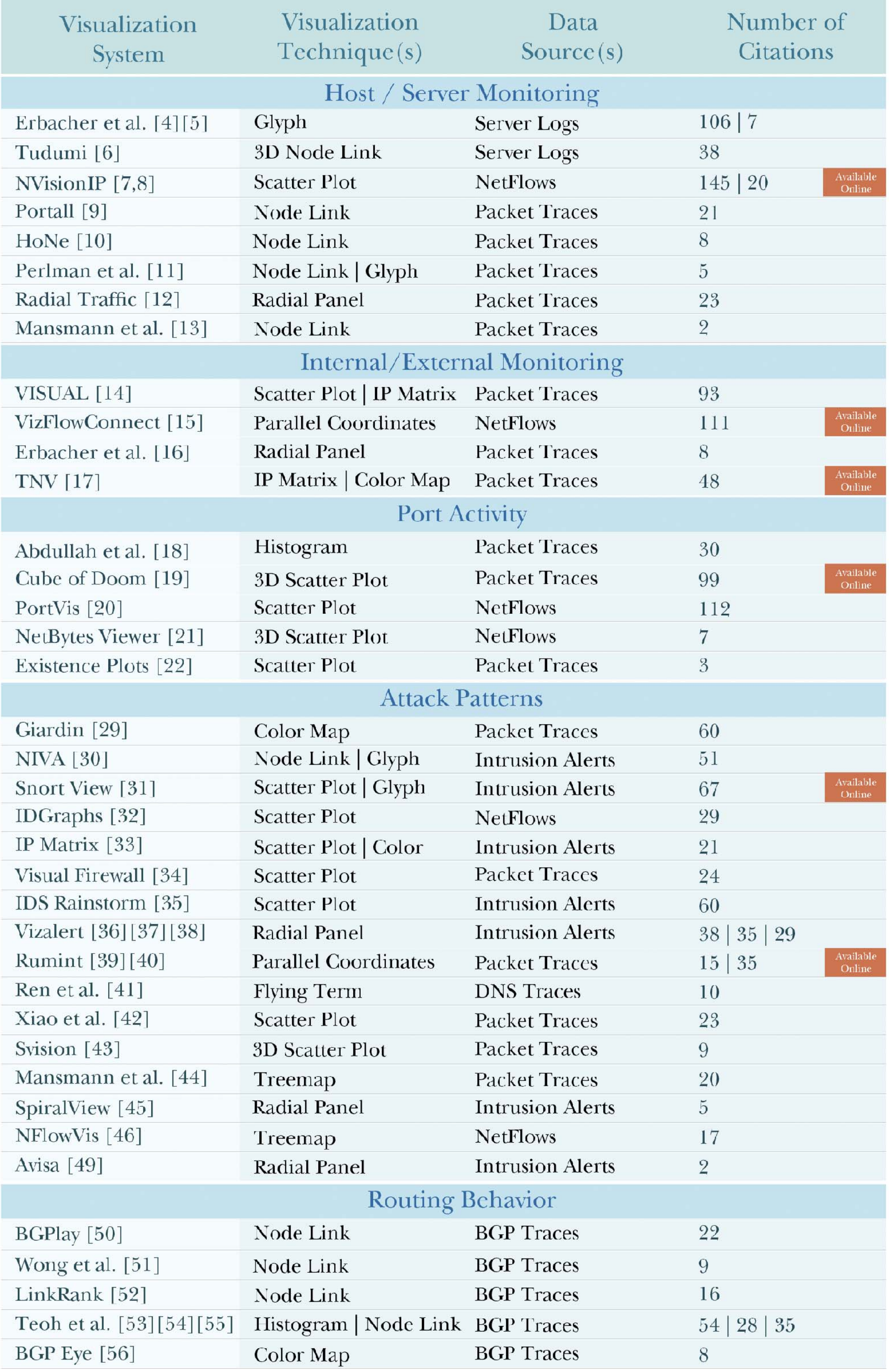


Fig. 18. LinkRank [52] displaying the impact of prefix hijacks on AS 6,939.

Situation awareness. The sheer volume of data generated from modern day networks and the high complexity of relations between data elements has proven conventional technologies as inefficient and inadequate for achieving situational awareness by a human analyst. The ultimate goal of a conventional visualization system is to provide a high-level view of security events to system analysts for more timely and informed decisions. While providing the means to conduct this higher level analysis, these conventional systems often leave the process of identifying critical events and threats to the analyst. Recent research reflects the gradual but inevitable transition of traditional visualization approaches toward developing processes and algorithms that prioritize situations and project critical events. This shift should be considered in the design of modern security visualization systems and should be an intrinsic part of their design philosophy.

Situational awareness is viewed in [57] as a "state of knowledge that results from a process" and must be distinguished from the process used to acquire that state. Subsequently, situation assessment is "the process" used to achieve that knowledge and is considered an aid in the cognitive process of situation awareness. The process of situation assessment from raw data to visual form is influenced by many supporting elements and environment models. It is the ultimate intention of a situation assessment process to project and highlight plausible situations that can thereby be presented by a visualization system that can enhance the situation awareness of an analyst. The situation assessment process must automatically identify and evaluate the impact of underlying events and relate them to assets of the monitored network. The output of these automated processes not only aids in perceiving and comprehending the current status of a network, but may also predict the trend of future situations.

TABLE 2
Best Practices of Security Visualization Systems, Grouped by Use-Case Class and Ordered Chronologically



Visualization System	Visualization Technique(s)	Data Source(s)	Number of Citations	
Host / Server Monitoring				
Erbacher et al. [4][5]	Glyph	Server Logs	106 | 7	
Tudumi [6]	3D Node Link	Server Logs	38	
NVisionIP [7,8]	Scatter Plot	NetFlows	145 | 20	Available Online
Portall [9]	Node Link	Packet Traces	21	
HoNe [10]	Node Link	Packet Traces	8	
Perlman et al. [11]	Node Link | Glyph	Packet Traces	5	
Radial Traffic [12]	Radial Panel	Packet Traces	23	
Mansmann et al. [13]	Node Link	Packet Traces	2	
Internal/External Monitoring				
VISUAL [14]	Scatter Plot | IP Matrix	Packet Traces	93	
VizFlowConnect [15]	Parallel Coordinates	NetFlows	111	Available Online
Erbacher et al. [16]	Radial Panel	Packet Traces	8	
TNV [17]	IP Matrix | Color Map	Packet Traces	48	Available Online
Port Activity				
Abdullah et al. [18]	Histogram	Packet Traces	30	
Cube of Doom [19]	3D Scatter Plot	Packet Traces	99	Available Online
PortVis [20]	Scatter Plot	NetFlows	112	
NetBytes Viewer [21]	3D Scatter Plot	NetFlows	7	
Existence Plots [22]	Scatter Plot	Packet Traces	3	
Attack Patterns				
Giardin [29]	Color Map	Packet Traces	60	
NIVA [30]	Node Link | Glyph	Intrusion Alerts	51	
Snort View [31]	Scatter Plot | Glyph	Intrusion Alerts	67	Available Online
IDGraphs [32]	Scatter Plot	NetFlows	29	
IP Matrix [33]	Scatter Plot | Color	Intrusion Alerts	21	
Visual Firewall [34]	Scatter Plot	Packet Traces	24	
IDS Rainstorm [35]	Scatter Plot	Intrusion Alerts	60	
Vizalert [36][37][38]	Radial Panel	Intrusion Alerts	38 | 35 | 29	
Rumint [39][40]	Parallel Coordinates	Packet Traces	15 | 35	Available Online
Ren et al. [41]	Flying Term	DNS Traces	10	
Xiao et al. [42]	Scatter Plot	Packet Traces	23	
Svision [43]	3D Scatter Plot	Packet Traces	9	
Mansmann et al. [44]	Treemap	Packet Traces	20	
SpiralView [45]	Radial Panel	Intrusion Alerts	5	
NFlowVis [46]	Treemap	NetFlows	17	
Avisa [49]	Radial Panel	Intrusion Alerts	2	
Routing Behavior				
BGPlay [50]	Node Link	BGP Traces	22	
Wong et al. [51]	Node Link	BGP Traces	9	
LinkRank [52]	Node Link	BGP Traces	16	
Teoh et al. [53][54][55]	Histogram | Node Link	BGP Traces	54 | 28 | 35	
BGP Eye [56]	Color Map	BGP Traces	8	

The number of citations is included to emphasize systems that have been referred more.

The process of achieving situational awareness is closely related to the capability of a system in conducting real time analysis. Security visualization systems, in their current state, are mostly suitable for offline forensics analysis. Real-time processing of network events requires extensive resources, both in terms of the computation power required to process an event, as well as the amount of memory needed to store the aggregated statistics. The exponential increase in link speeds and consequently, the higher number of security events triggered, poses challenges to this emerging field to cope with higher traffic throughput. A viable option to address this performance requirement is to leverage the parallel processing capabilities of current generation Graphics Processing Units (GPUs) to offload processing entirely from the CPU to thousands of hardware threads. Security researchers should look further into utilizing highly data parallel algorithms to make use of the massive parallelism of GPU architectures.

User experience. When thinking of security visualization systems many tend to envision it in terms of aesthetics: is the general look and feel of the system well designed? Others may think of it in functional terms: does the system perform what it promises? User experience looks beyond function and aesthetic [58]. Knowing the target audience of a system [59], defining who they are, identifying their requirements, and designing with the goal of user experience can ensure that all aspects and ramifications of the working context is known with explicit intent. Focal user experience elements such as familiarity, learnability, responsiveness, performance, intuitiveness, efficiency, helpfulness, and satisfactoriness ensures, to a larger extent, that a user only focuses on accomplishing her goals while being able to access appropriate, relevant content at the right time and place. One of the reasons that security visualization systems, despite their great potential, are not often incorporated in security appliance dashboards is the result of failing to address the focal points of user experience. Security visualization systems should be designed around the explicit needs of security analysts and their requirements must be taken into account every step of the way. The cognitive task analysis process performed by Erbacher et al. [60] is a valuable study in identifying the specific needs of network managers and analysts, in assessing the critical concepts required by security visualizations, and understanding the variety of situations analysts encounter.

Scalability. This phenomenon typically manifests itself as a growth in the amount of data being created and the size of a typical data set or model. The number of security events and traces produced in modern type networks has risen greatly in recent years, leading to scalability issues in conventional visualization techniques. Although larger, higher resolution displays can be used to increase the scalability of visualizations, unfortunately, in almost every industry and application segment, the growth in data far exceeds the growth in computing power. Additionally, different techniques have been proposed for scaling up visualizations on single [61] or multimonitor [62] environments. This brings up the need for security related events to be intelligently preprocessed, filtered, transformed, or statistically summarized as discussed above and further elaborated below.

Occlusion. In all security visualization systems described in this survey, only a few perform processing prior to visual display. As mentioned previously, the sheer amount of data in network security not only causes scalability issues in visualization systems but also leads to the phenomenon of occlusion and overcrowding of displays. Tending to display every aspect of a data set prior to any processing or filtering mechanism is far from practical. This unfavorable phenomenon decreases the power of visualization and consequently reduces the power of the human visual system in perceiving patterns and trends within the underlying data. IP address, for example, is a well-known feature used in security visualization. Many systems incorporate this feature inside their designs by displaying the complete IP range on a single axis. Mapping 2^{32} elements to a single axis leads to a single pixel to be mapped to thousands of IPs, generating multiple overlapping issues. With the growth in demand for address space becoming more and more urgent, the use of Internet Protocol Version 6 (IPv6), with support of 2^{128} addresses, is becoming eminent. As of February 3, 2011, the Internet Assigned Numbers Authoritys (IANA) store of unallocated IPv4 address space has been fully exhausted. This is a sign that designers of security visualizations need to consider and thereby focus on developing novel techniques and paradigms that address occlusion in an acceptable manner. In relation to IPv6, although still in its infancy, limited research [63] has started to accommodate this massive increase, but greater research and analysis is needed.

Interactive techniques concentrated on both the data and visual spaces are a necessity in understanding complex visualizations. As described in [64], although interactions on inherent attributes of a data set such as IP addresses, port numbers, or protocol types can simplify complex security visualizations, but with the gradual removal of elements from the display the underlying insights of the data set is also gradually lost. In visualizing network graphs, for example, Van Ham and van Wijk [65] argue that due to the high connectivity of small world graphs one cannot gain insight into the structure of the network through filtering mechanisms of inherent attributes, since these procedures often tend to remove articulation nodes resulting in a disconnected graph [66]. Thereby, as previously stated in the situation awareness section and as seen in visual systems like [64], [67] computed attributes should be calculated using statistical, clustering, or machine learning approaches. These higher level attributes can better discern the inherent structures of data sets. Interactions such as zooming, distorting, or filtering through dynamic queries on these calculated structural properties can lead to much greater insight into the underlying patterns of the data set.

Visualization technique. Multiple visualization techniques are utilized in the design of security visualizations. From scatter plots and heat maps to glyphs, parallel coordinates, and node link graphs. In most cases, designed in both two and three dimensional space. Based on this survey, we identified that most three dimensional systems [6], [47], [68], [69], [70], [71], compared to conventional 2D systems, were harder for an analyst to perceive and interact with. Mainly, because efficient 3D interaction is much harder to achieve than interaction with 2D displays. The 3D systems require users to rotate and zoom the display to fully understand the data, requiring constant interaction from an already overworked analyst. Users often lose sight of data by overzooming or overpanning the display. The disadvantages of a 3D view is in comparing patterns, widths, and heights that are at various distances from the user. Avoiding occlusion is also difficult and often requires user navigation. Enhancing and fine tuning existing two dimensional techniques for the purpose of network security is a more rational approach than dealing with inherent issues of 3D displays. Various research studies [72], [73], [74] have empirically examined user tasks in 2D and 3D interfaces. These studies suggest that although moving to a higher dimension may convey additional semantic information, but due to the increase in cognitive demands and lack of additional function and control over the 3D objects, the task performance gain is minimal [75]. Shneiderman [76] provides helpful guidelines for designers to follow in developing 3D interfaces that facilitate user tasks.

Adding another dimension of data to an interface can be achieved using several methods. These may include the use of glyphs, small multiples, or color. Glyphs, unlike color are not perceived preattentively and must also be semantically decoded, requiring extra attention. Small multiples can display different slices of a multidimensional data set without the visual clutter of superimposing all dimensions on a single graph. Color is a key element in encoding additional dimensions within a visualization. It can be used to show differences, similarities, to emphasize, to help recall, and convey meanings. So many security visualization systems that are meant to give insight use color just to decorate. Security visualizations should devise a color plan to project information about visual elements. Color should also be used consistently throughout the system since change in color pattern may signal a change in meaning. Designing a system in a way that is both informative and aesthetically pleasing remains a challenge in this field.

Preserving privacy. Security systems tend to analyze user behavior through monitoring network traffic or analyzing audit traces. The conflicting desire between user expectations regarding the protection of their data against collection and analysis, and those of an enterprise requiring user data to provide secure, efficient, and stable network services is subject to argument. This argument requires a solution that can provide an acceptable balance between the privacy of the users and the requirements of detection systems. *Pseudonymity* [77], [78] is a form of reversible anonymization approach that balances these interests. It protects the privacy of users by using *pseudonyms* instead of true identities and allows for reidentification of users in cases of suspicious behavior. Various network trace [79], [80], [81], [82], web server [83], anomaly detection [84], and misuse detection [85] audit data pseudonymizers have been developed and security visualization systems should incorporate these concepts in situations where user privacy is of priority.

Evaluation. Security visualization has attracted the attention of many researchers in identifying the ever-increasing issue of intrusive activities. However, its adoption to real-world applications has been hampered due to system complexity, as these systems require a substantial amount of testing, evaluation, and tuning prior to deployment. To prove the fact that a visualization system is capable of representing correlations of events and that it can be used in the identification of irregular behaviors and possible intrusions, it must undergo a thorough evaluation process. Of the many visualization systems surveyed in this paper, only a couple have performed usability studies [86], [87], while the rest have undergone scarce, ad hoc, and unsystematic evaluations. Running these systems over real labeled network traces with a comprehensive and extensive set of intrusions and abnormal behavior is the most idealistic methodology for testing and evaluation. One of the reasons may be that a full fledged labeled benchmark data set is lacking in the field of security visualization so that systems could use as a reference to evaluate and compare their systems. Data Sets provided through the PREDICT program [88], [89] and the USC/ISI ANT project [90] can act as good starting points for evaluation purposes. Availability and cost of usability experts and security analysts also affects the evaluation process. Analytic and empirical evaluations of systems are also of concern in this field.

5 Conclusion

As the number of security related events generated in modern networks is on the rise, the need for network security visualization systems is felt more than ever. In this paper, we have examined recent works in network security visualization from a use-case perspective. Five use-case classes, each representing a different application area, were defined and several recent works in each category were thoroughly described. We detailed the underlying data sources of network security visualization and gave a few examples of each category. Analysis of these systems motivated us to examine several issues and concerns surrounding this emerging field. We elaborated on the advantages and shortcomings of all use-case classes and shed light on paths that researchers should focus toward. We aggregated the findings of our work into an informative table for future references. While the field of visualization is as wide as imagination allows, we hope that the analysis and taxonomy presented here will motivate better future work in this area.

References

[1] C. Ware, *Information Visualization: Perception for Design.* Morgan Kaufmann Publishers, Inc., 2004.

[2] G. Conti, *Security Data Visualization.* No Starch Press, 2007.

[3] R. Marty, *Applied Security Visualization.* Addison-Wesley Professional, 2008.

[4] R. Erbacher, K. Walker, and D. Frincke, "Intrusion and Misuse Detection in Large-Scale Systems," *IEEE Computer Graphics and Applications,* vol. 22, no. 1, pp. 38-48, Jan./Feb. 2002.

[5] R. Erbacher, "Intrusion Behavior Detection through Visualization," *Proc. IEEE Int'l Conf. Systems, Man and Cybernetics,* pp. 2507-2513, 2003.

[6] T. Takada and H. Koike, "Tudumi: Information Visualization System for Monitoring and Auditing Computer Logs," *Proc. Sixth Int'l Conf. Information Visualisation,* pp. 570-576, 2002.

[7] K. Lakkaraju, W. Yurcik, and A. Lee, "NVisionIP: Netflow Visualizations of System State for Security Situational Awareness," *Proc. ACM Workshop Visualization and Data Mining for Computer Security,* vol. 29, pp. 65-72, 2004.

[8] K. Lakkaraju, R. Bearavolu, A. Slagell, W. Yurcik, and S. North, "Closing-the-Loop in Nvisionip: Integrating Discovery and Search in Security Visualizations," *Proc. IEEE Workshop Visualization for Computer Security (VizSEC '05),* pp. 75-82, 2005.

[9] G. Fink, P. Muessig, and C. North, "Visual Correlation of Host Processes and Network Traffic," *Proc. IEEE Workshop Visualization for Computer Security (VizSEC 05),* pp. 11-19, 2005.

[10] G. Fink, V. Duggirala, R. Correa, and C. North, "Bridging the Host-Network Divide: Survey, Taxonomy, and Solution," *Proc. 20th USENIX Conf. Large Installation System Administration,* pp. 247-262, 2006.

[11] J. Pearlman and P. Rheingans, "Visualizing Network Security Events Using Compound Glyphs from a Service-oriented Perspective," *Proc. Workshop Visualization for Computer Security (VizSEC '07),* pp. 131-146, 2008.

[12] D. Keim, F. Mansmann, J. Schneidewind, and T. Schreck, "Monitoring Network Traffic with Radial Traffic Analyzer," *Proc. IEEE Symp. Visual Analytics Science and Technology,* pp. 123-128, 2006.

[13] F. Mansman, L. Meier, and D.A. Keim, "Visualization of Host Behavior for Network Security," *Proc. Workshop Visualization for Computer Security (VizSEC '07),* pp. 187-202, 2008.

[14] R. Ball, G.A. Fink, and C. North, "Home-Centric Visualization of Network Traffic for Security Administration," *Proc. ACM Workshop Visualization and Data Mining for Computer Security,* pp. 55-64, 2004.

[15] X. Yin, W. Yurcik, M. Treaster, Y. Li, and K. Lakkaraju, "Visflowconnect: Netflow Visualizations of Link Relationships for Security Situational Awareness," *Proc. ACM Workshop Visualization and Data Mining for Computer Security,* pp. 26-34, 2004.

[16] R. Erbacher, K. Christensen, and A. Sundberg, "Designing Visualization Capabilities for ids Challenges," *Proc. IEEE Workshop Visualization for Computer Security (VizSEC '05),* pp. 121-127, 2005.

[17] J. Goodall, W. Lutters, P. Rheingans, and A. Komlodi, "Preserving the Big Picture: Visual Network Traffic Analysis with tnv," *Proc. IEEE Workshop Visualization for Computer Security (VizSEC '05),* pp. 47-54, 2005.

[18] K. Abdullah, C. Lee, G. Conti, and J. Copeland, "Visualizing Network Data for Intrusion Detection," *Proc. Sixth Ann. IEEE SMC Information Assurance Workshop (IAW '05),* pp. 100-108, 2005.

[19] S. Lau, "The Spinning Cube of Potential Doom," *Comm. the ACM,* vol. 47, no. 6, pp. 25-26, 2004.

[20] J. McPherson, K. Ma, P. Krystosk, T. Bartoletti, and M. Christensen, "PortVis: A Tool for Port-Based Detection of Security Events," *Proc. the ACM Workshop Visualization and Data Mining for Computer Security,* pp. 73-81, 2004.

[21] T. Taylor, S. Brooks, and J. McHugh, "Netbytes Viewer: An Entity-based Netflow Visualization Utility for Identifying Intrusive Behavior," *Proc. Workshop Visualization for Computer Security (VizSEC '07),* pp. 101-114, 2008.

[22] J. Janies, "Existence Plots: A Low-Resolution Time Series for Port Behavior Analysis," *Proc. Fifth Int'l Workshop Visualization for Computer Security (VizSec '08),* pp. 161-168, 2008.

[23] (2011) Re-Inventing Network Security. Palo Alto Networks. http://www.paloaltonetworks.com/literature/whitepapers/Re-inventing-Net work-Security.pdf, 2011.

[24] H. Debar and A. Wespi, "Aggregation and Correlation of Intrusion-Detection Alerts," *Proc. Fourth Int'l Symp. Recent Advances in Intrusion Detection,* pp. 85-103, 2001.

[25] B. Morin, L. Mé, H. Debar, and M. Ducassé, "M2D2: A Formal Data Model for IDS Alert Correlation," *Proc. Fifth Int'l Symp. Recent Advances in Intrusion Detection (RAID '02),* pp. 115-137, 2002.

[26] M. Shin, E. Kim, and K. Ryu, "False Alarm Classification Model for Network-Based Intrusion Detection System," *Proc. Int'l Conf. Intelligent Data Eng. and Automated Learning (IDEAL),* pp. 259-265, 2004.

[27] F. Cuppens and A. Miege, "Alert Correlation in a Cooperative Intrusion Detection Framework," *Proc. IEEE Symp. Security and Privacy,* pp. 202-215, 2002.

[28] F. Valeur, G. Vigna, C. Kruegel, and R. Kemmerer, "Comprehensive Approach to Intrusion Detection Alert Correlation," *IEEE Trans. Dependable and Secure Computing,* vol. 1, no. 3, pp. 146-169, July-Sept. 2004.

[29] L. Girardin, "An Eye on Network Intruder-Administrator Shootouts," *Proc. First Conf. Workshop Intrusion Detection and Network Monitoring,* vol. 1, pp. 3-13, 1999.

[30] K. Nyarko, T. Capers, C. Scott, and K. Ladeji-Osias, "Network Intrusion Visualization with niva, an Intrusion Detection Visual Analyzer with Haptic Integration," *Proc. 10th Symp. Haptic Interfaces for Virtual Environment and Teleoperator Systems (HAPTICS '02),* pp. 277 -284, 2002.

[31] H. Koike and K. Ohno, "SnortView: Visualization System of Snort Logs," *Proc. ACM Workshop Visualization and Data Mining for Computer Security,* vol. 29, pp. 143-147, 2004.

[32] P. Ren, Y. Gao, Z. Li, Y. Chen, and B. Watson, "Idgraphs: Intrusion Detection and Analysis Using Histographs," *Proc. IEEE Workshop Visualization for Computer Security (VizSEC '05),* pp. 39-46, 2005.

[33] H. Koike, K. Ohno, and K. Koizumi, "Visualizing Cyber Attacks Using ip Matrix," *Proc. IEEE Workshop Visualization for Computer Security (VizSEC '05),* pp. 91-98, 2005.

[34] C. Lee, J. Trost, N. Gibbs, R. Beyah, and J. Copeland, "Visual Firewall: Real-Time Network Security Monitor," *Proc. IEEE Workshop Visualization for Computer Security (VizSEC '05),* pp. 129-136, 2005.

[35] K. Abdullah, C. Lee, G. Conti, J. Copeland, and J. Stasko, "Ids Rainstorm: Visualizing ids Alarms," *Proc. IEEE Workshop Visualization for Computer Security (VizSEC '05),* pp. 1-10, 2005.

[36] Y. Livnat, J. Agutter, S. Moon, R. Erbacher, and S. Foresti, "A Visualization Paradigm for Network Intrusion Detection," *Proc. Sixth Ann. IEEE SMC Information Assurance Workshop (IAW '05),* pp. 92-99, 2005.

[37] Y. Livnat, J. Agutter, S. Moon, and S. Foresti, "Visual Correlation for Situational Awareness," *Proc. IEEE Symp. Information Visualization (INFOVIS '05),* pp. 95-102, 2005.

[38] S. Foresti, J. Agutter, Y. Livnat, S. Moon, and R. Erbacher, "Visual Correlation of Network Alerts," *IEEE Computer Graphics and Applications,* vol. 26, no. 2, pp. 48-59, Mar./Apr. 2006.

[39] G. Conti, K. Abdullah, J. Grizzard, J. Stasko, J. Copeland, M. Ahamad, H. Owen, and C. Lee, "Countering Security Information Overload through Alert and Packet Visualization," *IEEE Computer Graphics and Applications,* vol. 26, no. 2, pp. 60-70, Mar./Apr. 2006.

[40] S. Krasser, G. Conti, J. Grizzard, J. Gribschaw, and H. Owen, "Real-Time and Forensic Network Data Analysis Using Animated and Coordinated Visualization," *Proc. Sixth Ann. IEEE SMC Information Assurance Workshop (IAW '05),* pp. 42-49, 2005.

[41] P. Ren, J. Kristoff, and B. Gooch, "Visualizing dns Traffic," *Proc. Third Int'l Workshop Visualization for Computer Security (VizSEC '06),* pp. 23-30, 2006.

[42] L. Xiao, J. Gerth, and P. Hanrahan, "Enhancing Visual Analysis of Network Traffic Using a Knowledge Representation," *Proc. IEEE Symp. Visual Analytics Science and Technology,* pp. 107-114, 2006.

[43] I.-V. Onut and A.A. Ghorbani, "Svision: A Novel Visual Network-Anomaly Identification Technique," *Computers Security,* vol. 26, no. 3, pp. 201-212, 2007.

[44] F. Mansmann, D. Keim, S. North, B. Rexroad, and D. Sheleheda, "Visual Analysis of Network Traffic for Resource Planning, Interactive Monitoring, and Interpretation of Security Threats," *IEEE Trans. Visualization and Computer Graphics,* vol. 13, no. 6, pp. 1105-1112, Nov./Dec. 2007.

[45] E. Bertini, P. Hertzog, and D. Lalanne, "Spiralview: Towards Security Policies Assessment through Visual Correlation of Network Resources with Evolution of Alarms," *Proc. IEEE Symp. Visual Analytics Science and Technology,* pp. 139-146, 2007.

[46] F. Fischer, F. Mansmann, D.A. Keim, S. Pietzko, and M. Waldvogel, "Large-Scale Network Monitoring for Visual Analysis of Attacks," *Proc. Fifth Int'l Workshop Visualization for Computer Security (VizSec '08),* pp. 111-118, 2008.

[47] A. Yelizarov and D. Gamayunov, "Visualization of Complex Attacks and State of Attacked Network," *Proc. Sixth Int'l Workshop Visualization for Cyber Security (VizSec '09),* pp. 1-9, 2009.

[48] Q. Liao, A. Striegel, and N. Chawla, "Visualizing Graph Dynamics and Similarity for Enterprise Network Security and Management," *Proc. Seventh Int'l Symp. Visualization for Cyber Security (VizSec '10),* pp. 34-45, 2010.

[49] H. Shiravi, A. Shiravi, and A. Ghorbani, "Ids Alert Visualization and Monitoring through Heuristic Host Selection," *Proc. 12th Int'l Conf. Information and Comm. Security,* pp. 445-458, 2010.

[50] L. Colitti, G. Di Battista, F. Mariani, M. Patrignani, and M. Pizzonia, "Visualizing Interdomain Routing with BGPlay," *J. Graph Algorithms and Applications,* vol. 9, pp. 117-148, 2005.

[51] T. Wong, V. Jacobson, and C. Alaettinoglu, "Internet Routing Anomaly Detection and Visualization," *Proc. Int'l Conf. Dependable Systems and Networks (DSN '05),* pp. 172-181, 2005.

[52] M. Lad, D. Massey, and L. Zhang, "Visualizing Internet Routing Changes," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 6, pp. 1450-1460, Nov./Dec. 2006.

[53] S.T. Teoh, K.-L. Ma, S. Wu, and X. Zhao, "Case Study: Interactive Visualization for Internet Security," *Proc. IEEE Visualization (VIS '02),* pp. 505-508, 2002.

[54] S.T. Teoh, K. Zhang, S.-M. Tseng, K.-L. Ma, and S.F. Wu, "Combining Visual and Automated Data Mining for Near-Real-Time Anomaly Detection and Analysis in bgp," *Proc. ACM Workshop Visualization and Data Mining for Computer Security,* pp. 35-44, 2004.

[55] S.T. Teoh, K.-L. Ma, S. Wu, and T. Jankun-Kelly, "Detecting Flaws and Intruders with Visual Data Analysis," *IEEE Computer Graphics and Applications,* vol. 24, no. 5, pp. 27-35, Sept./Oct. 2004.

[56] S.T. Teoh, S. Ranjan, A. Nucci, and C.-N. Chuah, "Bgp Eye: A New Visualization Tool for Real-Time Detection and Analysis of bgp Anomalies," *Proc. Third Int'l Workshop Visualization for Computer Security (VizSEC '06),* pp. 81-90, 2006.

[57] M. Endsley, "Toward a Theory of Situation Awareness in Dynamic Systems: Situation Awareness," *Human Factors,* vol. 37, no. 1, pp. 32-64, 1995.

[58] J.J. Garrett, *The Elements of User Experience: User-Centered Design for the Web.* New Riders Publishing, 2002.

[59] R.F. Erbacher, "User Issues in Visual Monitoring Environments," *Proc. Int'l Conf. Imaging Science, Systems, and Technology,* pp. 644-650, 2001.

[60] R.F. Erbacher, D.A. Frincke, P.C. Wong, S. Moody, and G. Fink, "A Multi-Phase Network Situational Awareness Cognitive Task Analysis," *Information Visualization,* vol. 9, pp. 204-219, 2010.

[61] J.-D. Fekete and C. Plaisant, "Interactive Information Visualization of a Million Items," *Proc. IEEE Symp. Information Visualization (InfoVis '02),* pp. 117-124, 2002.

[62] B. Yost and C. North, "The Perceptual Scalability of Visualization," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 5, pp. 837-844, Sept./Oct. 2006.

[63] D. Barrera and P. van Oorschot, "Security Visualization Tools and ipv6 Addresses," *Proc. Sixth Int'l Workshop Visualization for Cyber Security (VizSec '09),* pp. 21-26, 2009.

[64] A. Perer and B. Shneiderman, "Balancing Systematic and Flexible Exploration of Social Networks," *IEEE Trans. Visualization and Computer Graphics,* vol. 12, no. 5, pp. 693-700, Sept./Oct. 2006.

[65] F. van Ham and J.J. van Wijk, "Interactive Visualization of Small World Graphs," *Proc. IEEE Symp. Information Visualization,* pp. 199-206, 2004.

[66] F. Boutin, J. Thièvre, and M. Hascoët, "Focus-Based Filtering + Clustering Technique for Power-Law Networks with Small World Phenomenon," *Proc. Conf. Visualization and Data Analysis '06,* vol. 6060, no. 1, 2006.

[67] J. Heer and D. Boyd, "Vizster: Visualizing Online Social Networks," *Proc. IEEE Symp. Information Visualization,* pp. 32-39, 2005.

[68] A. Komlodi, P. Rheingans, U. Ayachit, J. Goodall, and A. Joshi, "A User-Centered Look at Glyph-Based Security Visualization," *Proc. IEEE Workshop Visualization for Computer Security (VizSEC '05),* pp. 21-28, 2005.

[69] A. Oline and D. Reiners, "Exploring Three-dimensional Visualization for Intrusion Detection," *Proc. IEEE Workshop Visualization for Computer Security (VizSEC '05),* pp. 113-120, 2005.

[70] J. Oberheide, M. Karir, and D. Blazakis, "Vast: Visualizing Autonomous System Topology," *Proc. the Third Int'l Workshop Visualization for Computer Security (VizSEC '06),* pp. 71-80, 2006.

[71] Z. Jiawan, Y. Peng, L. Liangfu, and C. Lei, "Netviewer: A Visualization Tool for Network Security Events," *Proc. Int'l Conf. Networks Security, Wireless Comm. and Trusted Computing.,* pp. 434-437, 2009.

[72] S.J. Westerman and T. Cribbin, "Mapping Semantic Information in Virtual Space: Dimensions, Variance and Individual Differences," *Int'l J. Human-Computer Studies,* vol. 53, no. 5, pp. 765-787, 2000.

[73] A. Cockburn and B. McKenzie, "3D or Not 3D?: Evaluating the Effect of the Third Dimension in a Document Management System," *Proc. SIGCHI Conf. Human Factors in Computing Systems,* pp. 434-441, 2001.

[74] A. Cockburn and B. McKenzie, "Evaluating the Effectiveness of Spatial Memory in 2D and 3D Physical and Virtual Environments," *Proc. SIGCHI Conf. Human Factors in Computing Systems: Changing Our World, Changing Ourselves,* pp. 203-210, 2002.

[75] J. Steele and N. Iliinsky, *Beautiful Visualization: Looking at Data through the Eyes of Experts,* first ed. O'Reilly Media, 2010.

[76] B. Shneiderman, "Why Not Make Interfaces Better than 3D Reality?," *IEEE Computer Graphics and Applications,* vol. 23, no. 6, pp. 12-15, Nov./Dec. 2003.

[77] U. Flegel, "Introduction," *Privacy-Respecting Intrusion Detection,* ser. Advances in Information Security, pp. 3-8. Springer, 2007.

[78] J. Biskup and U. Flegel, "On Pseudonymization of Audit Data for Intrusion Detection," *Proc. Int'l Workshop Designing Privacy Enhancing Technologies: Design Issues in Anonymity and Unobservability,* pp. 161-180, 2001.

[79] G. Minshall (2011) Tcpdpriv, http://ita.ee.lbl.gov/html/contrib/tcpdpriv.html, 2011.

[80] M. Peuhkuri, "A Method to Compress and Anonymize Packet Traces," *Proc. First ACM SIGCOMM Workshop Internet Measurement,* pp. 257-261, 2001.

[81] J. Xu, J. Fan, M. Ammar, and S.B. Moon, "On the Design and Performance of Prefix-Preserving ip Traffic Trace Anonymization," *Proc. First ACM SIGCOMM Workshop Internet Measurement,* pp. 263-266, 2001.

[82] R. Pang and V. Paxson, "A High-Level Programming Environment for Packet Trace Anonymization and Transformation," *Proc. SIGCOMM '03,* pp. 339-351, 2003.

[83] C. Eckert and A. Pircher, "Internet Anonymity: Problems and Solutions," *Proc. IFIP TC11 16th Ann. Working Conf. Information Security, Trusted Information: The New Decade Challenge,* pp. 35-50, 2001.

[84] E. Lundin and E. Jonsson, "Anomaly-Based Intrusion Detection: Privacy Concerns and Other Problems," *Computer Networks,* vol. 34, pp. 623-640, 2000.

[85] M. Sobirey, S. Fischer-Hübner, and K. Rannenberg, "Pseudonymous Audit for Privacy Enhanced Intrusion Detection," *Proc. IFIP TC11 13 Int'l Conf. Information Security in Research and Business,* pp. 151-163, 1997.

[86] X. Suo, Y. Zhu, and G.S. Owen, "Measuring the Complexity of Computer Security Visualization Designs," *Proc. Workshop Visualization for Computer Security (VizSEC '07),* 2008.

[87] J. Goodall, "Visualization is Better! a Comparative Evaluation," *Proc. Sixth Int'l WorkshopVisualization for Cyber Security (VizSec '09),* pp. 57-68, 2009.

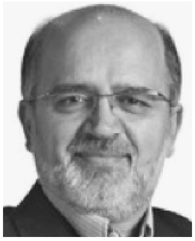
[88] (2011) The PREDICT website. http://www.predict.org, 2011.

[89] (2011) The Skaion website. http://www.skaion.com, 2011.

[90] (2011) USC/ISI Network Traces. http://www.isi.edu/ant/index.html, 2011.

Hadi Shiravi received the BEng degree in software engineering from The University of Science and Culture, Iran and the MCS degree from the University of New Brunswick, Canada. He is currently working toward the PhD degree in the Information Security Centre of Excellence, University of New Brunswick, Canada. His research interests include network security, security data visualization, network traffic generation, botnet detection, botnet propagation, and p2p botnet simulation. He is also a member of the Canadian Honeynet Project.

Ali Shiravi received the BEng degree in software engineering from Tarbiat Moallem University of Tehran, Iran and the MSc degree from Sharif University of Technology, Tehran, Iran. He is currently working toward the PhD degree in the Information Security Centre of Excellence, University of New Brunswick, Canada. His research interests include network security, network traffic analysis, graph theory, graph clustering, network simulation, network traffic profiling, and security data visualization.



Ali A. Ghorbani is currently a professor and a dean with the University of New Brunswick (UNB), where he is the director of Information Security Center of Excellence, and is also the coordinator of the Privacy, Security and Trust network annual conference. He holds a UNB Research Scholar position and is the coeditor-in-chief of the *Computational Intelligence: An International Journal*, and an associate editor of the *International Journal of Information Technology and Web Engineering*. His current research interests include web intelligence, network security, complex adaptive systems, critical infrastructure protection, and trust and security assurance. He is a member of the Association for Computing Machinery, the IEEE Computer Society, the IEEE, and the Canadian Society for Computational Studies of Intelligence.