Focusing on Context in Network Traffic Analysis

John R. Goodall, Wayne G. Lutters, Penny Rheingans, and Anita Komlodi
University of Maryland, Baltimore County

With network size and complexity continuously increasing, securing computing infrastructures from attacks is an escalating challenge. Intrusion detection systems (IDSs) are often used to aid analysts' efforts by automatically identifying successful and unsuccessful system attacks or abuses. Although IDS alerts can be a useful first step in uncovering security compromises, they're often just that: a starting point. While IDS alerts contain some pertinent information, analysts can rarely determine an event's accuracy and severity from an IDS alert alone. Rather, they must collect and construct the event's relevant context within voluminous network traffic data. Building this contextual understanding of an event is fundamental to intrusion detection (ID) analysis.

The Time-Based Network Traffic Visualizer combines low-level, textual detail with multiple visualizations of the larger context to help users construct a security event's big picture.

Whether the starting point of analysis is data rich (as with an IDS alert) or data poor (as with a phone call from a user), analysis of a network security event is a complex task. Generally, contextual data comes from collecting packet-level detail of the event-related network traffic. The textual or tabular tools that analysts currently use—such as Tcpdump (http://www.tcpdump.org) or Ethereal (http://www.ethereal.com)—focus on extracting this vital, detailed information from individual packets. However, such tools lack a mechanism for providing a simultaneous big picture view of the data. As analysts try to understand the details of the packets within the larger context of surrounding network activity, they must continually shift their attention, increasing their already considerable cognitive load. In addition, these tools excel at filtering and searching for details—but only if analysts know exactly what they're looking for in the data. For less structured data exploration tasks aimed at discovering and understanding patterns and anomalies, the tools are less effective.

To overcome these limitations, we designed an information visualization tool that gives network analysts a simultaneous view of both the big picture and individual packet details. By integrating both of these essential views into a single tool, we can help reduce analysts' cognitive burden. The tool also helps preserve the context required to comprehensively support the process of discovering, analyzing, and making decisions about anomalous or potentially malicious activity. We've grounded our visualization design in the actual work practices of security analysts. Here, we describe this design process, present details of our visualization support tool, and demonstrate how it aids ID in three common scenarios.

Intrusion detection: an overview

In previous research,[1] we interviewed a diverse sample of security analysts to identify some of ID's most significant challenges. One such challenge is data overload—a well-documented problem with many examples in the literature.[2] This pressing concern is one reason that information visualization—which can make large amounts of data more compact and understandable—offers such an appealing solution to the challenges of ID.

Intrusion detection tasks

Based on our findings from this fieldwork, we classify ID work into three tasks: monitoring, analysis, and response.[1] The monitoring task is typically focused on surveillance of the output of an IDS and other systems that monitor network state. This time-consuming task focuses on analysts' need to maintain situational awareness of their networks' dynamic activities. The analysis task focuses on determining the accuracy and severity of security events uncovered during monitoring. This is the most complex ID task, requiring substantial knowledge and experience. Often, on further analysis, indications of malicious activity turn out to be benign. Even when analysts find that such activities are truly malicious, they must then determine how to prioritize an event. They do this on the basis of their knowledge of both the event itself and the relative importance of the targeted network device. Finally, response refers to an analyst's reaction to a security event. This task ranges from proactively countering the attack if it's a true malicious event, to updating their systems to ignore the event in the future if it's a false positive.

 Published by the IEEE Computer Society

Visualization and Network Security

Much recent research has applied information visualization to the considerable challenges facing security analysts. Most of this research has been directed at what we broadly call the *monitoring task*: detecting intrusive or anomalous events among the myriad benign events in a network's traffic flow. Two systems supporting the monitoring task are NVisionIP[1] and VisFlowConnect,[2] which are both geared at increasing an analyst's situational awareness by visualizing NetFlows (aggregated traffic records). NVisionIP displays a class-B network as a scatterplot in the broadest view and lets analysts drill down into the data through a small multiple view and a histogram of host details. VisFlowConnect uses parallel coordinate plots and animation for link analysis.

VisAlert[3] is an extensible visualization that can accept multiple data sources, including intrusion detection (ID) alerts and system log files. VisAlert integrates these into a single display depicting alerts as vectors between a radial view's perimeter (representing alert time and type) and interior (representing network topology).

D'Amico and Larkin[4] describe prototypes that focus on the temporal importance of security events. One view uses 3D space to visualize time and classification on a vertical wall; sources and destinations are on the horizontal floor on either side of the wall, with the lines between each floor going through the wall representing events.

PortVis[5] takes summary network data and visualizes port activity as a scatterplot linked to several other data views. Erbacher and colleagues[6] have developed animated glyph-based visualizations that use system log files to show connections from external hosts to a monitored server or small network environment. Teoh and colleagues[7] visualize routing data to detect anomalies, intrusions, and router instability using three visualization methods.

The systems previously described can alert analysts to anomalous activity on their networks or systems, augment the monitoring tasks, or increase situational awareness. However, they're unlikely to support the more detailed packet-level inspection required to analyze network security events.

Krasser and colleagues[8] developed a system that shows connections between source IP address and destination port in parallel coordinates. The system uses 3D animation to provide temporal meaning in combination with packet size and protocol. This system, like ours, allows analysts to view the raw network traffic's details. This combination of higher-level visualizations with low-level textual details is crucial for analysts to gain meaning from security events.

References

1. K. Lakkaraju, W. Yurcik, and A.J. Lee, "NVisionIP: Netflow Visualizations of System State for Security Situational Awareness," *Proc. ACM Workshop Visualization and Data Mining for Computer Security* (VizSEC/DMSEC), ACM Press, 2004, pp. 65-72.
2. X. Yin et al., "VisFlowConnect: NetFlow Visualizations of Link Relationships for Security Situational Awareness," *Proc. ACM Workshop Visualization and Data Mining for Computer Security* (VizSEC/DMSEC), ACM Press, 2004, pp. 26-34.
3. Y. Livnat et al., "A Visualization Paradigm for Network Intrusion Detection," *Proc. IEEE Workshop Information Assurance and Security* (IAW), IEEE Press, 2005, pp. 92-99.
4. A. D'Amico and M. Larkin, "Methods of Visualizing Temporal Patterns in and Mission Impact of Computer Security Breaches," *Proc. DARPA Information Survivability Conf. and Exposition* (DISCEX II), IEEE CS Press, 2001, pp. 343-354.
5. J. McPherson et al., "PortVis: A Tool for Port-Based Detection of Security Events," *Proc. ACM Workshop Visualization and Data Mining for Computer Security* (VizSEC/DMSEC), ACM Press, 2004, pp. 73-81.
6. R.F. Erbacher et al., "Intrusion and Misuse Detection in Large-Scale Systems," *IEEE Computer Graphics and Applications*, vol. 22, no. 1, 2002, pp. 38-48.
7. S.-T. Teoh et al., "Detecting Flaws and Intruders with Visual Data Analysis," *IEEE Computer Graphics and Applications*, vol. 24, no. 5, 2004, pp. 27-35.
8. S. Krasser et al., "Real-Time and Forensic Network Data Analysis Using Animated and Coordinated Visualization," *Proc. IEEE Workshop Information Assurance and Security* (IAW), IEEE Press, 2005, pp. 42-49.

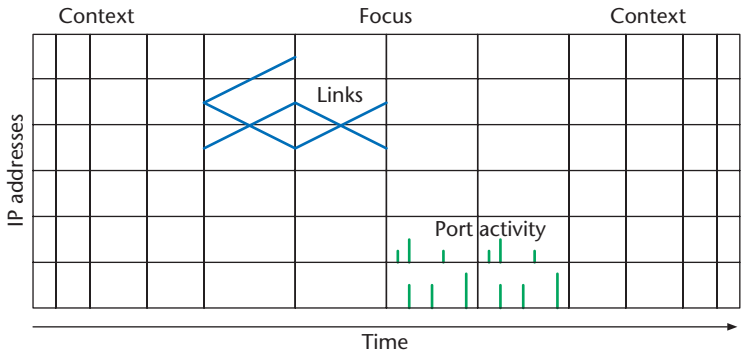
For both the monitoring and analysis tasks, information visualization shows great potential. As the "Visualization and Network Security" sidebar describes, much recent research into this area focuses on supporting the monitoring task. Because there's currently little visual support for the complexities of analysis, we targeted our information visualization tool to support this task. Our findings suggest that one of the key analysis challenges is to both attain and preserve a high-level contextual awareness while investigating an event's low-level details.

The importance of context

The need to identify and retain an event's context during analysis is a recurring theme among analysts. Indeed, analysts can rarely make decisions about security events (such as IDS alerts) based solely on the available data (the alert's text, for example). Instead, they supplement this with data collected from other sources, often captured on an ad hoc basis. Analysts also rely heavily on their own knowledge and experience.

To properly diagnose an event, analysts must assemble this contextual information, whether it's based on supplemental data sources, their own knowledge, or both. Diagnosis includes such things as determining the event's root cause, reconstructing its timeline, and identifying any related outcomes. To make these determinations, analysts must fuse together the event's details with the larger, surrounding context of activity.

Nonetheless, many popular tools for collecting and investigating contextual data take a microlevel approach; they fail to support the development of a coherent understanding of the surrounding context. Other tools offer higher-level aggregations that can provide event context. To obtain a comprehensive view,



1 The main visualization's conceptual design. In the display, columns represent time intervals, while rows represent individual host IP addresses. Relative port activity within each host cell is shown as vertical bars that represent aggregated bins of ports active during that time period.

analysts must continually refocus attention between these higher-level tools and tools that provide details. In our interviews, analysts repeatedly discussed how they would lose context once they left high-level context displays to examine packet details. Lacking external representational supports, analysts often rely on their short-term memory to integrate low-level and high-level data.

The role of time

In addition to the importance of context, our study found that time was also a crucial factor in the analysis process. There were several reasons why the data's temporal attributes were fundamentally important. One reason was data synchronization. All the data sources and tools that our participants used generated time stamps that corresponded nearly exactly, despite being generated on different hosts. All participants used Network Time Protocol (NTP) on their systems, which let them synchronize different data elements from different sources.

In addition, while an IDS or other automated system might initiate a security event, it could also come from a more ambiguous source, such as user feedback. With such vague sources, beginning the analysis task from anything other than time was problematic. End users, for example, were more likely to tell administrators that something strange happened at a specific or even approximate time (such as "after lunch") than they were to recite their network addresses or the protocols that their instant-message clients used.

Finally, there's the issue of temporal context. When events occur before or after trigger events, analysts can get vital clues about the security event's nature. As a straightforward example, if just prior to an event's investigation, every network host was port scanned from a single destination, this could indicate that an attacker doing reconnaissance identified and exploited a vulnerability.

Network data's temporal aspects:

- let analysts correlate events across multiple systems,
- are easily identifiable by all data sources, and
- are essential to attack deconstruction.

Time-Based Network Traffic Visualizer

TNV is a visualization tool grounded in an understanding of the work practices of security analysts. We designed it to support ID analysis by giving analysts a visual display that facilitates pattern and anomaly recognition, particularly over time. It also offers more focused views on packet-level detail in the context of the surrounding network traffic.

Conceptual design

To support the monitoring task, visualizations must facilitate rapid pattern recognition. To support the more complex analysis task, however, our goals were to

- facilitate data exploration and correlation between events,
- support the discovery of relationships between hosts, and
- help analysts understand suspected attacks and anomalies.

So, TNV's emphasis is not on speed, but rather on providing multiple views of networking data to support decision-making and understanding. Thus, TNV can be more complex than visualizations designed to support monitoring.

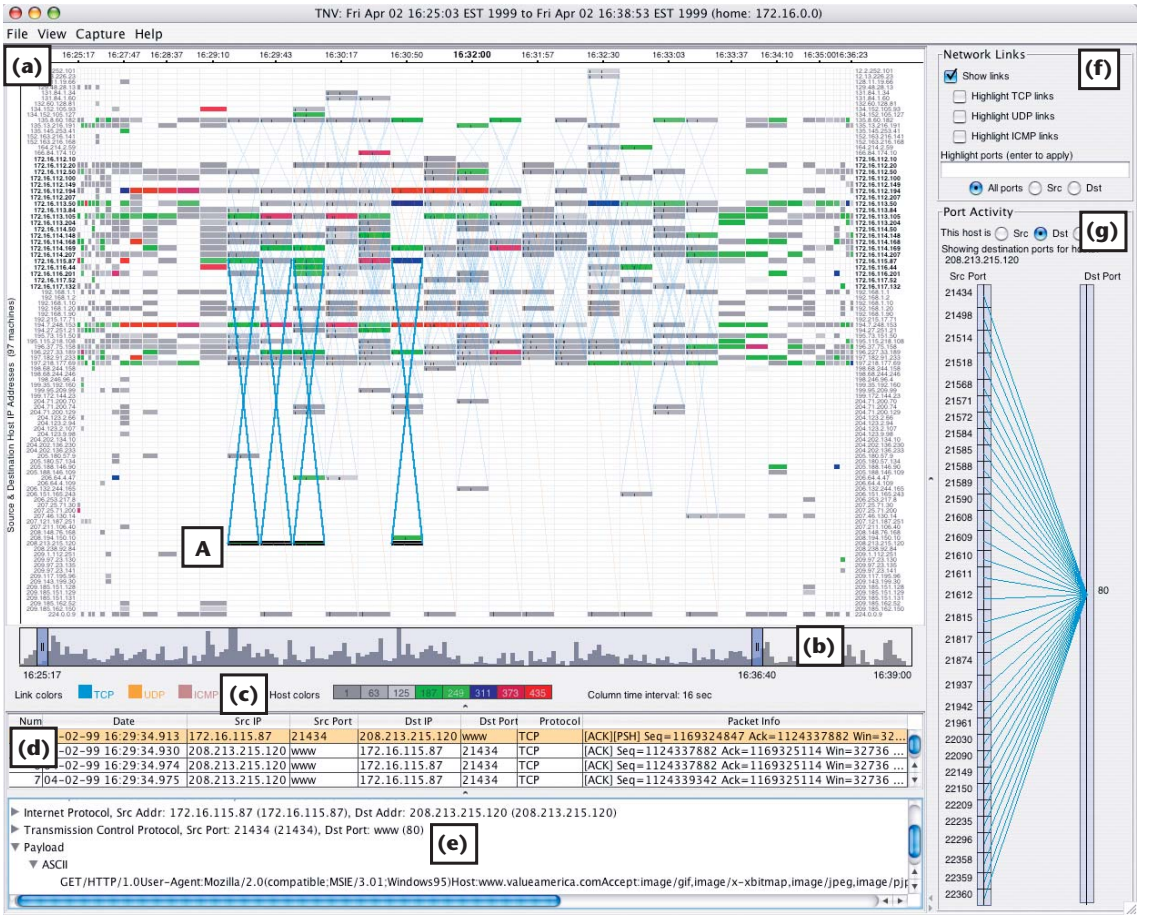
Given the importance of context in ID analysis, TNV's main visualization takes a focus + context approach, comparable to the Perspective Wall.[3] Figure 1 shows a conceptual drawing. Columns represent time intervals and rows represent hosts. The focal area's wider columns show link communications between hosts, as well as individual host's port activity. The context areas' columns gradually narrow, providing continuity between the focal and contextual areas without a sudden jump to a smaller width. The context areas show the number of packets within a time period, but not link or port activity. This offers additional context and conserves display space. Using this approach, analysts can explore interest areas in greater detail, zooming in on traffic patterns, while still viewing contextual information about the time periods before and after the target area.

Visualizing network traffic over time

Figure 2 shows the TNV display. The main visual display (Figure 2a) combines a matrix-style display of host IP address and network packet time stamps with a display that shows links between hosts. Other TNV features include

- the navigation and data overview mechanism (Figure 2b),
- the legend panel (Figure 2c),
- a table of packets (Figure 2d) for the selected host (host A),
- packet details for a selected row in the table (Figure 2e),
- the emphasis filtering panel (Figure 2f), and
- host A's port activity (Figure 2g).

The number of packets for each time interval is encoded in the resulting cell's color. The user-defined color-to-



2 TNV showing 50,000 network packets. (a) The main visualization matrix, along with details of a selected host A, including network links with Web (TCP port 80) activity. Other TNV features include (b) the navigation and data overview mechanism; (c) the legend panel; (d) a table of packets for the selected host; (e) packet details for a selected row in the table; (f) the emphasis filtering panel; and (g) the selected host's port activity.

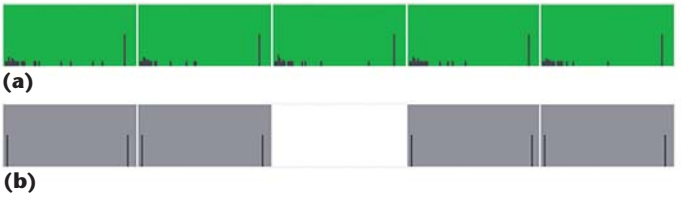
number-of-packets mapping is in the center of the legend panel (see Figure 2c). In this example, gray represents a relatively low number of packets and red a high number of packets; hues within each color scale represent gradations. Analysts can therefore quickly identify hotspots within the data set—a key requirement, according to the analysts we interviewed. Because we designed the visualization around a timeline, analysts can also identify trends and anomalies in network activity for individual hosts. So, for example, if time intervals containing few packets are interrupted by a time interval with numerous packets, analysts might investigate further.

Hosts on the analyst's local network are of paramount importance, and TNV emphasizes them accordingly. Remote hosts, at least initially, are often important only in terms of their potential to attack the analyst's network. Analysts can set an IP address space that constitutes their home, or local, networks. TNV gives the hosts that meet this criterion a slightly greater height and larger, bolded labels (see the identical labels at the right and left of the main visualization display). This lets analysts quickly differentiate between internal and external network hosts. This is important because certain kinds of network communication—such as intranet Web traffic—is expected from internal network hosts, but suspicious if it originates from an external host. In TNV, these unusual network communications between hosts are readily identifiable because it explicitly displays such links.

Network links

TNV can display network link communications between hosts within each time period, as shown in the center of Figure 2a. Links are drawn from the link's source to the link's destination, starting from either side of the column. In most cases, links are displayed as an X pattern, because hosts often both send and receive data as they communicate. A single line between hosts might indicate a host sending packets to determine which ports are active. The link colors, which the user defines, show the link's traffic protocol. By default, TNV draws links with a very low opacity and a fixed, relatively narrow width. This gives analysts an overall sense of link communications between hosts without cluttering the display. To encourage data exploration, various filtering mechanisms (described later) emphasize links that match the analyst's criteria.

In network traffic analysis, analysts must understand communication patterns between hosts. However, current network tools require analysts to mentally store and correlate these patterns. By making such links visually



3 Relative port activity within host cells. (a) Many connections to multiple ports indicates a possible port scan, while (b) two equally sized ports indicate typical client-server activity, such as Web traffic.

explicit, TNV can both assist ID analysis and make it easier for novices to learn what normal network activity looks like. For example, public Web servers typically have links from both local and remote hosts, so we can expect the server to be connected to many lines from multiple hosts. In contrast, an intranet Web server should have connections only from local hosts. Also, textual network analysis tools demand that analysts store and recall IP addresses in their memory as they inspect the data. Using TNV, which explicitly shows the links between machines, significantly reduces analysts' cognitive burden and lets them focus on solving the intrusion problem rather than deciphering textual data.

The link display is somewhat like parallel coordinates, which consist of parallel axes with line segments between them.[4] In parallel coordinates, the axes represent different data attributes; intersections of line segments and axes represent the value of that observation for the variable represented by that axis. In TNV, however, the parallel axes represent different values for the same attribute: time. The value of each intersection is also constant, representing the host IP address. The display therefore reveals relationships between hosts and how those relationships change and evolve over time. Unlike parallel coordinates, however, the link display can't show relations between multiple attributes. However, we can see some recognizable patterns in parallel coordinates in the display (and in the port activity display, discussed later). Among these patterns are fan-in and fan-out links, where one host communicates with many other hosts.

Port activity

While the link display reveals the hosts involved in a communication, analysts must examine the ports to understand the nature of that communication. To support the gleaning of port information, TNV offers two different views of port activity:

- a highly aggregated representation integrated into the main visualization, and
- a separate representation that explicitly shows the details of port relationships.

The main visualization can display relative port activity for each time period within each host cell. Because there are 65,536 possible ports and limited display space, we bin ports into groups and represent each group as a vertical bar. Each bar's height corresponds to the relative percentage of port activity within that time period. Because servers generally run on the well-known (privileged) ports—those numbered below 1,024—we bin these ports in smaller groups, and bin the remaining ports according to the remaining display space. This is a high-level overview of port activity. It's not intended to reveal exactly which ports are active, but rather to show general patterns that can reveal certain kinds of patterns or anomalies.

Figure 3 shows two port bar examples. Figure 3a shows what a slow, randomized scan might look like. This scan is a portion of one taken over a five-hour period; each cell represents about 30 minutes. The type of slow scan it shows can often be difficult to automatically detect. Visually, however, it's easy to see the many short bars within each cell. In contrast, Figure 3b shows more common client-server activity. Here, a low-numbered server port and a high-numbered client port—represented as bars toward the edges of cells of equal height—are the only active ports.

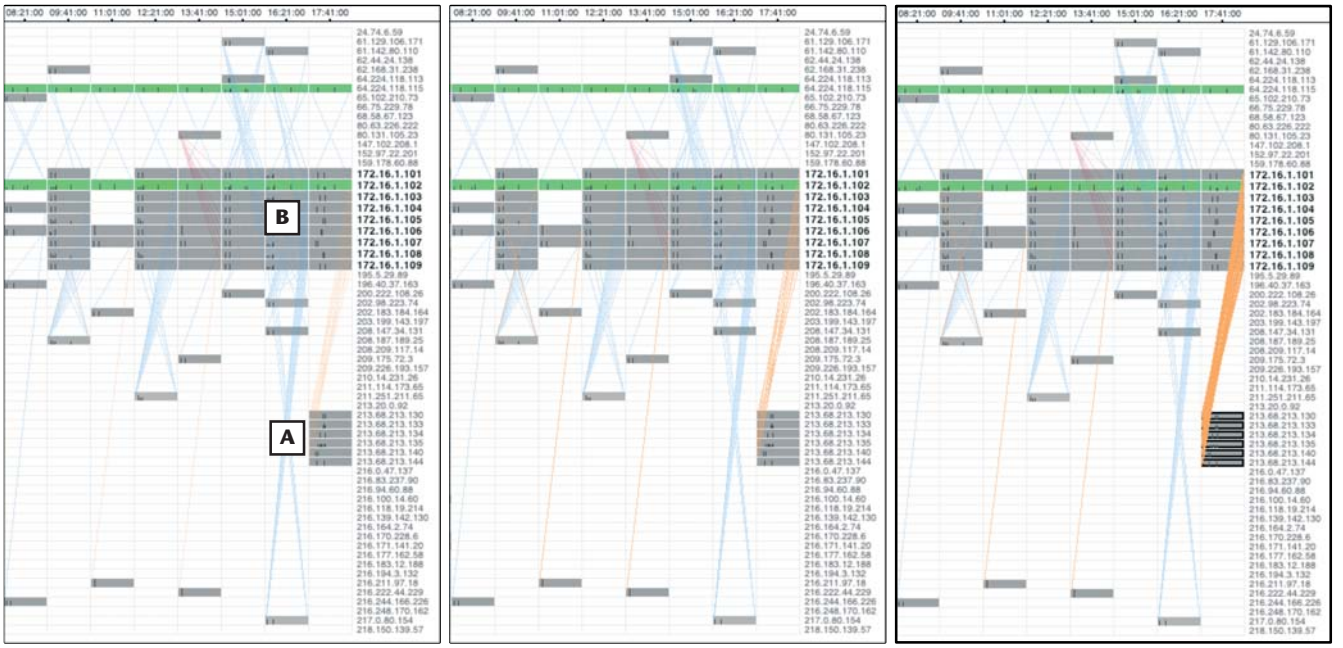
This aggregated summary is complemented by a more detailed view of the target host's port activity when analysts select a host or hosts, as Figure 2g shows. TNV shows port activity by the connections between two parallel axes representing the selected host's source and destination ports. The color of the connections matches the link color selected by the analyst for the main visualization. Analysts can view either the source or destination port activity, or both, for a selected host or hosts. The height of the boxes overlaid on the parallel axes shows the relative number of connections for each port. In Figure 2g, there are several relatively equally active source ports, all going to one destination port.

The analysts we interviewed reported their need for a display that explicitly showed a single-to-multiple port relationship to support their ID analysis. TNV's display easily accommodates this need. By showing the relative amount of port activity over time, TNV's display also shows strange, one-time port activity that often indicates anomalous activity, and is often overlooked with text-based tools.

Navigation, interaction, and details

Users navigate through and zoom into the main visualization's data by moving the handles on either side of the scroll navigation interface (Figure 2b). The scroll handles set the main display's start and end times and determine how much time each column represents. To zoom into the data, users move the handles closer together, decreasing each column's time interval; to increase the time each column represents, users move the handles farther apart. The scroll navigation's background is a histogram of the entire data set's relative network traffic activity. The data displayed in the main visualization is the shaded area between the handles. This gives analysts a very high-level overview of the data and keeps them continually aware of the current focus area within the context of the entire data set.

We designed TNV to encourage network traffic exploration. As Figure 2f shows, analysts can emphasize links based on port numbers or protocol type using a simple interface to highlight areas of interest. Rather than completely remove links that don't meet the analyst's criteria, the display emphasizes the links that do—by



4 TNV visualization of anomalous activity. (a) The initial display shows that a group of hosts A is sweeping the network B with UDP packets. The display highlights (b) increasing UDP activity and (c) the group of attacking hosts.

increasing their opacity—while keeping nonmatching links semitransparent. This alerts the analyst to interest areas without removing the context that other links provide. Analysts can also select a host cell or multiple cells to highlight the links associated with them in the selected time period. TNV makes these links opaque and increases the line thickness, making it clear which hosts the selection is communicating to. It's also possible to select an arbitrary screen area to highlight all of the hosts and links within it. This is especially useful for identifying all activity associated with a particular host.

In addition to displaying port activity and emphasizing host-associated links, picking a host or multiple hosts reveals the packet details associated with the selected time period. Analysts can anchor these details to the bottom of the display (as in Figure 2) or view them in a new window to improve readability. When an analyst selects a host (labeled A in Figure 2), TNV populates the table (see Figure 2d) with: a summary of all packets showing that arrival time, the source and destination address and port, and a summary of the other packet headers. When the analyst selects a table row, TNV displays details of an individual packet (See Figure 2e). This shows all of the packet headers, as well as the packet contents (or payload) in a tree format, similar to Ethereal. These textual details are essential to ID analysis. By providing multiple, linked views, analysts can simultaneously explore the data set from multiple perspectives—from a highly aggregated overview to the raw, packet-level details.

Applying TNV

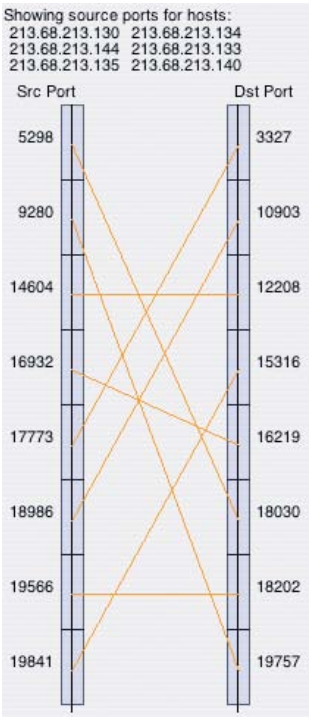
The following three scenarios demonstrate how TNV supports ID tasks and helps users understand network traffic. The first example's data comes from the HoneyNet Project's scan 21 (http://www.honeynet.org/scans/); the data for the other examples comes from MIT's ID evaluation data (http://www.ll.mit.edu/IST/ideval/data/1999/1999_data_index.html). Because these data sources have known limitations,[5] we plan to evaluate TNV in the field with analysts' own data. However, as our interviews confirmed, the following ID situations are like those that analysts typically encounter.

Scenario 1: discovering anomalies

This example shows how TNV supports discovery by quickly revealing anomalous activity that would be difficult to detect automatically, such as by an IDS. This data set includes a burst of User Datagram Protocol (UDP) packets to multiple destinations from four different source addresses, with different source and destination ports for each packet. The packets' source ports' randomization and multiple source addresses—likely crafted and sent from a single host—are unlikely to trigger an IDS alert.

Figure 4a shows a portion of TNV's initial visualization. The home network hosts are the bolded `172.16.1.*` addresses (labeled B in Figure 4a). UDP packets are displayed in orange; each column shows 80 minutes of activity. The rightmost column shows the network sweep. As this screen shows, a cluster of hosts (labeled A in Figure 4a) has sent many UDP packets to all of the local network hosts, which is anomalous in this data set. As the multiple vertical bars in the source host cells show, these connections clearly come from multiple ports. While this type of traffic is probably unusual in most networks, it is unquestionably unusual within this data set: the data contains several other UDP connections, all of which are sporadic and use a single port. Figure 4b's display shows the increasing UDP traffic.

Figure 4c shows the highlighted hosts and links, further emphasizing this UDP activity's anomalous nature. The other UDP links in the display are both from a



5 Display for selected hosts. Here, six hosts show an equal number of source and destination connections to and from multiple ports.

single port range, as the single vertical bar within the other two host cells shows (at the bottom of the screen). By emphasizing the UDP links, TNV also highlights the many-to-many relationship between the source and destination hosts in this traffic.

As Figure 5 shows, when the user selects the group of source hosts, TNV displays detailed port activity. The source and destination overlaid port boxes are of equal height, showing an equal number of packets with different source and destination ports. The visual pattern of multiple links crossing each other from one single, high-numbered port to another is atypical in this data set. It would be difficult for analysts to identify and correlate these seemingly random packets textually, because the attacker is using multiple source addresses and has randomized the source and destination ports. With TNV, this randomness is made visually explicit. The main visualization also reveals that the attacking hosts hadn't attempted any other prior attacks.

Analysts can examine the packet details to determine the attacker's motives. In this case, examining packet details shows that all packets contain a payload of `DOM`—the string that activates the `RST.b` Trojan. As this example shows, TNV can quickly alert analysts to anomalous activity and also help them to determine the meaning behind that activity—in this case, an attacker hunting for a possible back door.

Scenario 2: from discovery to analysis

ID analysis often begins with an IDS alert or other monitoring system notification. The alert itself, however, rarely contains sufficient details to let analysts determine the attack's accuracy and severity. In this scenario, we assume the analyst has received an IDS alert indicating the hosts involved and the attack type—a simple network management protocol (SNMP) attack. The analyst must therefore determine if:

- the attack succeeded,
- the attacker targeted any other hosts, and
- any other hosts were compromised.

Figure 6 shows three screen cutouts of the attack's surrounding activity. Figure 6a shows the hosts involved in this attack (labeled A), the local host being targeted, and an external host (labeled B) attempting the exploit. This activity is immediately suspicious; external hosts should rarely, if ever, be legitimately issuing queries over SNMP (which is typically used for performance monitoring). Additionally, SNMP traffic is often sporadic, with one host querying another periodically about its state. Here, the SNMP traffic is sustained over a long period of time (about 30 minutes). To determine if the attack was successful, the analyst must examine the packet contents, which is easy to do using TNV's detail tables.

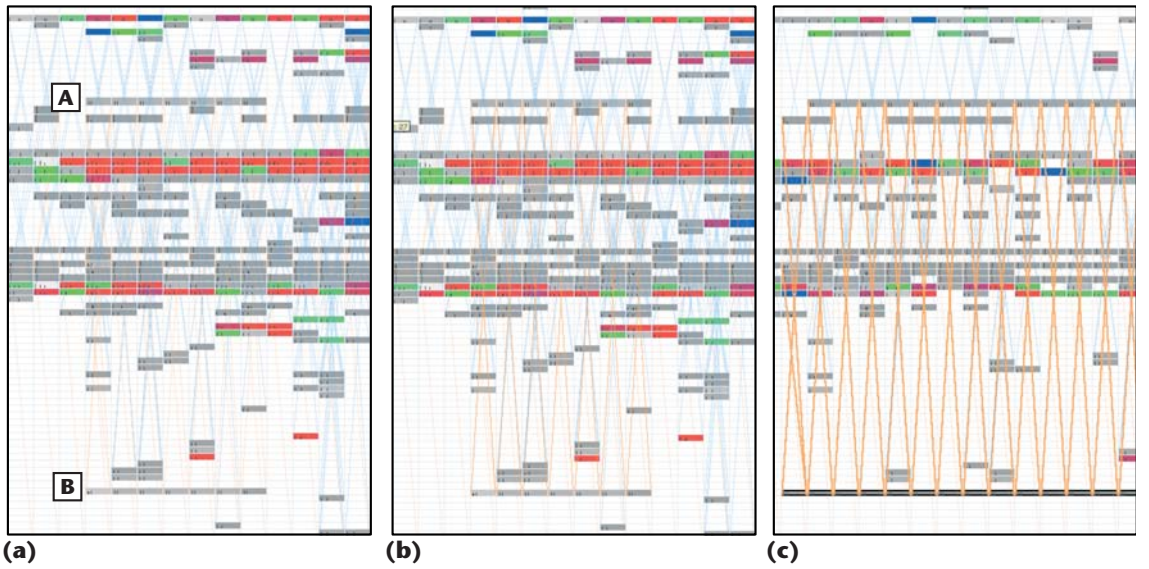
Next, the analyst must determine whether the attack affected any other hosts. Here, combining the visualization and textual packet details is useful, allowing analysts to quickly see which other hosts are communicating with the attack's source and whether this type of attack has been attempted anywhere else. In this example, the analyst can use the port highlighting mechanism to emphasize hosts that are sending SNMP requests (see Figure 6b). A quick glance at the display shows that there is no other SNMP traffic in the data set. The analysts can therefore be fairly certain that this was the only host targeted by this particular attack.

To view more information about communications between the attacker and the targeted host, the analyst can zoom in on the targeted host, reducing the time period that each column represents (see Figure 6c). As the leftmost column shows, the attacker is communicating with another host, which might warrant further investigation. The visualization also shows that the attacker is issuing domain name service (DNS) queries to two other hosts. Looking at the details of these queries shows the analysts that the requests were part of the attacker's reconnaissance activity immediately prior to the attack.

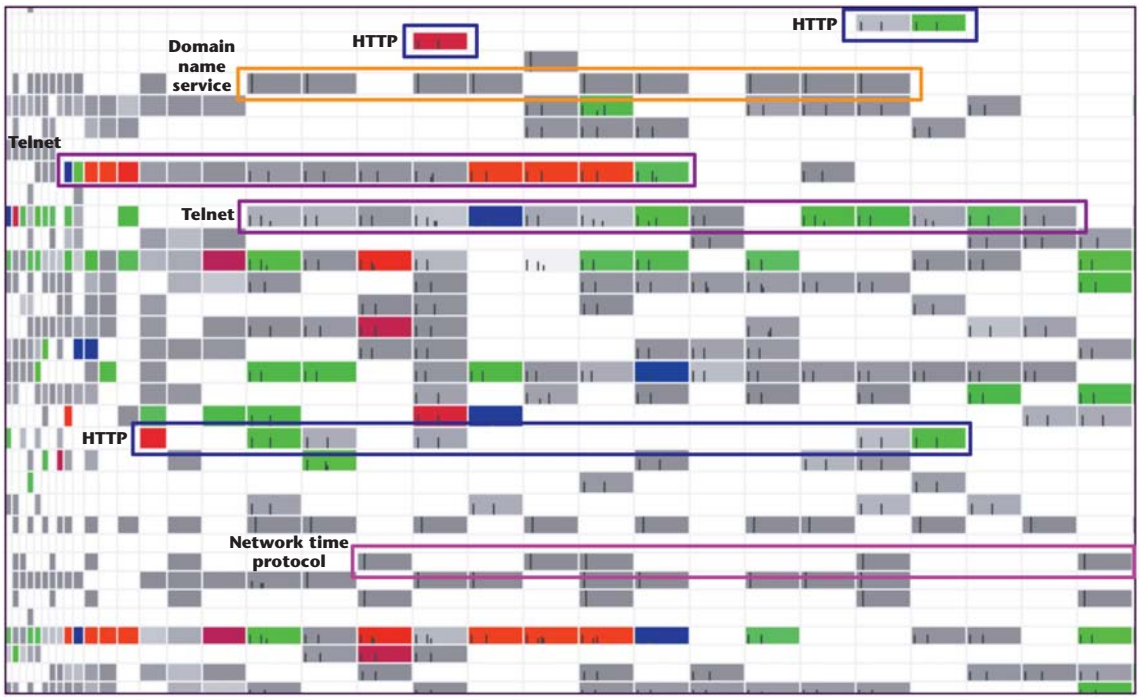
In these examples, the analyst relies on the textual details to examine packet contents and make the ultimate decision about the suspicious activity. The visualization helps the analysts keep such details in context, highlights the activity's anomalous nature, and supports the analyst's event deconstruction to determine both if other hosts have been attacked and if similar network attacks have occurred.

Scenario 3: learning the network

In ID, the importance of knowing the network cannot be overstated. In our previous work, ID analysts repeatedly said that their most critical tool was an intimate knowledge of the target environment.[6] However, gaining an understanding of what is normal—and therefore abnormal—on a network is a nontrivial task. It requires understanding not only how different networking protocols behave, but also how they are manifested in the context of a network. The textual tools analysts typically use help them learn about individual network packets, but they don't give them an understanding of normal network activity at a higher level. This is one of



6 Activity surrounding an attack. (a) Host B is attacking host A. (b) The display highlights port activity related to this attack and (c) zooms in to highlight the attacking host.



7 TNV display of different traffic types. Learning these patterns is essential in performing ID analysis.

the greatest strengths of a tool like TNV—it encourages both the high-level understanding of network traffic (through the main visualization), while also allowing exploration down to the level of packet content details.

Network traffic manifests differently on different networks. TNV can help analysts learn to understand their networks' normal traffic patterns. As Figure 7 shows, TNV's visualization can help analysts quickly and easily grasp the distinct visual patterns of different types of traffic. Login traffic—such as secure shell or telnet—is often sustained over long time periods with high activity levels. Figure 7 clearly shows this: there are few gaps during the duration of the telnet sessions (although these gaps also depend on what the client is doing and the time period's size for each column). In contrast, Web traffic (HTTP) generally manifests as sporadic bursts of high activity as clients request pages from a Web server. Visually, this displays as periods of high activity followed by no activity. The port summary bars—the vertical lines within each time period representing relative port activity—are typically displayed as one line toward the left of the cell (representing the server) and another of relatively equal size to the right. This is typical of client-server traffic, in which the client uses a high-numbered port or group of ports, and the server uses a well-known, low port number.

In these examples, two hosts are communicating: one client and one server. If there were multiple clients, there might be more, smaller port bars relative to the server lines. Figure 7 shows this multiple-client relationship in the lower of the two telnet examples. The server, which is highlighted, begins communicating with multiple clients on different high-number ports. DNS and NTP traffic usually appear as sporadic bursts of low activity. These types of traffic typically use a constant low-number source and destination port, which is usually represented by a single vertical bar toward the left of the host cell.

These examples are specific to the data sets. Other networks have their own distinct patterns that the analyst will come to recognize using a visual tool like TNV. Learning how TNV displays different kinds of traffic can help analysts learn their networks' distinctive character, while also helping them understand the more subtle differences in lower-level packet details.

Future work

Functionally, we plan to increase TNV's filtering and emphasis capabilities to permit more flexible data exploration (such as filtering on TCP flags). We also plan to let analysts reorder rows in the matrix, either manually or using a clustering algorithm. Currently, TNV's *y*-axis is ordered by IP address—to ensure that hosts on the same subnetwork are adjacent—but this doesn't allow much flexibility. In addition, we plan to incorporate additional, intermediate views of the network traffic between the overview and textual detail levels.

In our analyst interviews, we found that they typically prefilter network data, limiting its scope, before attempting detailed analysis. However, even with prefiltered data, the number of hosts can get very large on busy networks. Currently, TNV can easily fit approximately 100 hosts on a $1{,}280 \times 1{,}024$ display and still show the detail table. To increase the number of hosts that TNV can simultaneously display, our first strategy is to use a separate window for the detail table. To further increase TNV's scalability, we might extend the bifocal display to the *y*-axis as well, so that hosts toward the edges have smaller heights. Although aggregating hosts into subnets would also increase scalability, it would result in a loss of detail. To contend with this, we plan to run user evaluations to determine the acceptable tradeoffs between scalability and readability.

TNV has great potential for facilitating the ID analysis task and the process of understanding normal network traffic. Preliminary usability testing on an earlier version demonstrated that TNV is easy to learn and helped novices understand network patterns.[7] To further improve TNV's usability and assess its utility, we are planning further evaluations. A lab-based comparative evaluation of TNV and Ethereal, for example, will help us determine the relative strengths of each tool for particular tasks. We also plan to test TNV in the field, evaluating its utility in the context both of experts, who have intimate knowledge of their networks and data, and novices, who must gain such knowledge. ■

References

1. J.R. Goodall, "User Requirements and Design of a Visualization for Intrusion Detection Analysis," *Proc. IEEE Workshop Information Assurance and Security* (IAW), IEEE Press, 2005, pp. 394-401.
2. K. Julisch, "Clustering Intrusion Detection Alarms to Support Root Cause Analysis," *ACM Trans. Information and System Security*, vol. 6, no. 4, 2003, pp. 443-471.
3 J.D. Mackinlay, G.G. Robertson, and S.K. Card, "The Perspective Wall: Detail and Context Smoothly Integrated," *Proc. ACM Conf. Human Factors in Computing Systems* (CHI), ACM Press, 1991, pp. 173-179.
4. A. Inselberg, "The Plane with Parallel Coordinates," *The Visual Computer*, vol. 1, 1985, pp. 69-91.
5. J. McHugh, "Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Off-Line Intrusion Detection System Evaluation as Performed by Lincoln Laboratory," *ACM Trans. Information and System Security*, vol. 3, no. 4, 2000, pp. 262-294.
6. J.R. Goodall, W.G. Lutters, and A. Komlodi, "I Know My Network: Collaboration and Expertise in Intrusion Detection," *Proc. ACM Conf. Computer-Supported Cooperative Work* (CSCW), ACM Press, 2004, pp. 342-345.
7. J.R. Goodall et al., "A User-Centered Approach to Visualizing Network Traffic for Intrusion Detection," *Extended Abstracts ACM Conf. Human Factors in Computing Systems* (CHI), ACM Press, 2005, pp. 1403-1406.

John R. Goodall *is a doctoral candidate in the Information Systems Department at the University of Maryland, Baltimore County (UMBC). His research interests include workplace studies to inform design, information visualization, and usability for computer security. Goodall has a BA in history from Binghamton University and an MS in information systems from UMBC. Contact him at jgood@umbc.edu.*

Wayne G. Lutters *is an assistant professor in the Information Systems Department at UMBC. His research interests include examining the processes of expertise identification and information reuse in collaborative systems. Lutters has a BA in cognitive science and history from Connecticut College, and an MS and PhD in information and computer science from the University of California, Irvine. Contact him at lutters@umbc.edu.*

Penny Rheingans *is an associate professor in the Computer Science Department at UMBC. Her research interests include using perceptual and illustrative principles to show multivariate data with associated uncertainty and time varying structure. Rheingans has an AB in computer science from Harvard University and a PhD in computer science from the University of North Carolina, Chapel Hill. Contact her at rheingan@cs.umbc.edu.*

Anita Komlodi *is an assistant professor in the Information Systems Department at UMBC. Her research interests include learning about users' information-seeking behavior and, based on this knowledge, designing user interfaces for information systems. Komlodi has a PhD in library and information science from the University of Maryland, College Park. Contact her at komlodi@umbc.edu.*