Proposing a Multi-touch Interface for Intrusion Detection Environments

Jeffrey Guenther*
Simon Fraser University
School of Interactive Arts and Technology
Surrey, BC, Canada
jguenthe@sfu.ca

Fred Volk
Liberty University
Department of Psychology
Lynchburg, VA
fvolk@liberty.edu

Mark Shaneck
Liberty University
Department of Computer Science
Lynchburg, VA
mshaneck@liberty.edu

ABSTRACT

Network Intrusion Detection is a critical task in today's environment, where network attacks and intrusions are everyday occurrences and state-level cyber warfare is a major concern. At the same time, it is a very difficult task, in part due to the large scale of the data logs where the attack information is hidden, and also in part because of the lack of effective data exploration tools for the intrusion detection tasks. In this work, we examine the current state of visualization techniques and identify some key limitations. Based on this analysis, we propose a novel design for an interface for network security analysts, capitalizing on cutting edge technology, and discuss implications for future research.

Categories and Subject Descriptors

C.2.0 [**General**]: Security and Protection; C.2.3 [**Network Operations**]: Network Monitoring; H.1.2 [**User/Machine Systems**]: Human Factors; H.5.2 [**User Interfaces**]: GUI, Interaction Strategies; K.6.5 [**Security and Protection**]: Unauthorized Access

General Terms

Security, Human Factors

Keywords

Network Intrusion Detection Visualization, Data Exploration, Cognitive Task Analysis, Activity Theory, Natural User Interfaces

1. INTRODUCTION

Computer networks are an integral part of organizations of all sizes, and maintaining the security of these networks is a crucial part of operating them. As network-connected computing-devices become ubiquitous in almost every industry, the burden of maintaining network security compounds. Currently, network security personnel use a combination of command line tools and web interfaces to monitor and respond to network traffic. While these tools are excellent for accessing the details of traffic information, they do not present the data in the context of the whole network, something that is essential for assessing traffic patterns and alerts. Experienced analysts use their tacit understanding of the network environment, the sum of technical and organizational details which result in an operational network, as they work, making them invaluable to the organization. Consequently, it takes a long time for a new analyst to gain the environmental knowledge and wisdom to be able to successfully identify threats and respond to them. As networks have become more complex, the informational load on the analyst has become immense. There are hundreds of details to track and manage in an environment that is constantly in flux. To add more strain on the analyst, every decision to respond, or detail to be ignored, affects an organization's ability to keep its network secure. For the government, military, and large corporations, the ability to keep one's network secure is essential to maintain its operations. Network analysts need tools which help manage environmental complexity and afford intuitive analysis of traffic.

Creativity support tools [50], specifically visual analytics tools, are designed to help users manage this sort of complexity and make better decisions. Data analysis is a creative process, where creativity can be described as the action of connecting and expressing ideas. Tools for visual analytics allow users to begin exploring data. The context in which data are explored directly impacts the decisions made as result of the information. As a result, understanding the larger context of an event is essential. In network data analysis and intrusion detection, understanding the context of traffic data is paramount in distinguishing a false-positive alert from an actual alert. An analyst's understanding of their network enables them to determine if a seemingly suspicious traffic pattern is actually an exploit of a larger vulnerability, or is just something strange occurring on the network. Having tools which enable an analyst to rapidly explore data in context will be essential for the future defense of networks as traffic volumes grow and organizations increasingly depend on them for their critical operations. Analysts need tools which allow them to explore traffic data and help manage the cognitive load that network management tasks place upon them.

*Work done while at Liberty University.

VizSec '10, September 14, 2010, Ottawa, Ontario, Canada.

The primary challenge of network intrusion detection is textual analysis. In its most basic form, network traffic information is a semi-structured text file. Command-line tools like grep are excellent for manipulating textual data. A problem arises when an analyst needs to explore data for traffic patterns which might be suspicious, as command-line tools are not well suited for this task. Traditional text processing tools work best when the user knows what they are looking for. Often, analysts won't know what they are looking for, just something "strange" [22]. An analyst needs a tool which allows her to see traffic data in the context of the larger network environment and its relation to other traffic and alerts. With log sizes quickly heading into the range of terabytes, the task of making sense of traffic is becoming increasingly difficult. New analytics tools must be developed to handle this challenge. Patterns detected and alerts generated by sensors, like Snort [51] and MINDS [16], must be fused with contextual information, like device configurations, bandwidth usage, and other pertinent metadata. Combining alerts, detected anomalies, and a contextually-informed user interface, analysts will be able to more efficiently and effectually engage in the higher order problem solving needed to detect attacks.

We are not the first to propose work in this area. Many projects have dealt with the issue of providing better user interfaces to the security analyst. However, these are largely aimed at speeding up the individual steps that the analyst takes in the process of an intrusion investigation, as they are largely based on cognitive task analysis. This approach has its place and is incredibly useful to the analysts. Our claim is that this approach fundamentally limits the innovation that can take place in the field, and does not allow the analyst to change his approach to the task of securing the network and tracking down intrusions. Our contributions are as follows. We examine current visualization methods and explore the literature detailing the task analysis of the security analysts' workflows. We then propose a design for a new intrusion detection visualization/interaction framework which places information gained by sensors in the context of the broader network environment. This design makes use of the cutting edge technology of multi-touch interfaces to provide a comprehensive data visualization/exploration tool for the security analysts. To the best of our knowledge, no one has yet studied multi-touch interfaces in the context of intrusion detection. We also identify directions for future research and the role that we believe Activity Theory should play in user interface design for Intrusion Detection Systems.

The rest of this paper is organized as follows. In Section 2 we examine the limitations of the current approaches and outline the major roadblocks to innovation in this area of research. We then propose our design in Section 3. We outline the role of Activity Theory in IDS user interface design and conclude in Section 4.

2. CURRENT LIMITATIONS

Much work has been done in the area of visualization for intrusion detection. Many of these are informed by and built upon the results of cognitive task analysis of security analysts, where their workflow has been examined and broken down into steps. We believe that this approach has certain limitations, which we discuss in this section.

2.1 Vizualization Techniques

Published visualization methods for intrusion detection data range from the use of glyphs [14, 13] to specialized histograms [46]. We examine the strengths and weaknesses of these techniques and use this knowledge to inform our design.

Glyph-based network visualization [14] provides a unique approach to network visualization. It uses nodes and lines to indicate the nature of a connection, with line style indicating connection type. Directional arrows are used to indicate the direction of the connection. Cross-hatches are used to denote the number of users on a host, and hatch thickness indicates the number of connections per user. Color is used to describe the character of the connection, such as a connection time-out, a system that is not responding, or when a sensor has sensed an attack. Nodes are laid out around the perimeter of a circle, with the center of the circle being the monitored system. Nodes slowly fade from view as they become inactive, giving the viewer a sense of time, movement through which can be controlled by playing the visualization forward and backward. The technique is successful in visually representing different types of attacks as traffic patterns are clearly detectable. The authors recognize the need to be able to access more information about the nodes by allowing the user to click on them.

The primary purpose of this visualization is monitoring of a single device and in this regard is successful. While the visualization provides a local context for activity on a single system, it does not provide this information in the context of the whole network. This method demonstrates the successful use of glyphs in utilizing human pattern recognition and provides a helpful level of abstraction for the viewer. The viewer is aided by the pre-classification of connections and is not overwhelmed by packet and connection information. As conceded earlier, additional information should be accessible to the user on demand.

Continuing with the use of glyphs, Erbacher [13] uses the method to visualize activity between interconnected devices. Building off the methods discussed in the previous paper, the thickness of a circle representing a machine increases with load. Device type (i.e. whether a device is a router, hub, switch, or host) is indicated by its central ring. Hubs, and switches are represented by "the joining of network connections" [13].

These added components allow the visualization of traffic in the context of the larger network environment. This refinement faces the same issues as the did the previous glyph-based method. Analysts are not provided with a means to drill into packet information. System load is added to the visualization, providing analysts with another dimension of information when analyzing trends. It is successful in providing a visualization for the purpose of monitoring network activity. Similar techniques that allow for visualization of network traffic using node-link graphs include [35, 52, 12, 17, 34].

Intrusion detections systems, like signature based Snort [51], anomaly based systems such as MINDS [16], and port scan detectors [26, 15], are commonly used to generate alerts when suspicious traffic is encountered. Alerts are generated based on traffic signatures and can often result in false-positives. Koike and Ohno [29] detail a tool for visualizing Snort alerts. Rather than refining signatures in an attempt to have fewer false-positives, visualization is used to determine which alerts are worth investigating. SnortView uses

a 2-dimensional grid to show source IP address on the vertical axis and time on the horizontal. Alerts, represented by characterizing symbols, are placed along the horizontal axis. When alerts are clicked, a line is drawn between the source IP and the destination IP, which is in the matrix frame where each destination IP is represented by a column. The details of an alert are shown in a lower panel when it is clicked. Color is used to indicate the priority of an alert. Analysts use their knowledge of the types and frequency of alerts common on their network to help determine if an alert is a false-positive.

This tool is effective for handling alerts and is able to show alerts in the context of other source IP addresses. However, the logical context of the network is not shown. One advantage of the tool is its real-time operation. The major drawbacks are the lack of connectivity information and packet information.

Livnat et. al. [32] detail the most promising visualization tool encountered in our review of the literature. VisAlert uses a circular layout to correlate the three main factors of an alert: what, when, and where. Alerts are spread around the circumference of the circle. The circumference of the circle represents the span of time being investigated. Alerts are distinguished by color and are stacked in time order by type around the outer circle. Nodes representing devices are within the circle and are connected by arcs to their corresponding alerts. Level of detail calculations are used to control visual complexity. Graphical indicators like arc thickness and node size are used to draw the attention of the analyst. Larger nodes have more alerts connected to them. Arc thickness is used to show that multiple alerts have been triggered in a short period of time. Situational awareness is promoted by an embedded network map overlaid with nodes receiving alerts.

This visualization does an excellent job of controlling visual complexity by using the animation of transitions between levels of detail. It is able to aggregate sensor logs and system logs. The use of a circular timeline allows alerts to be shown in their temporal context. Other elements of context such as device name and IP address are included with the node. Missing from the visualization is the ability drill into individual packet information. In the results of the paper, several shortcomings were identified by the testers: more detailed information about the alerts was wanted, along with a history of explored events, as well as the ability to capture the visualization for future reference and communication. These requests encompass a set of requirements, which will be discussed in Section 3.1.

Another approach taken for visualization is the use of histograms [46] and scatter plots [18, 25, 36]. These approaches visualize a large amount of data in a relatively small space, and thus are particularly effective at quickly bringing attention to attacks such as port scans or otherwise "noisy" attacks. They are useful for helping an analyst begin an investigation, providing starting points, but do not allow for more fine grained analysis of the data. Interface design for intrusion detection has also been considered using haptic integration [44] as well as incorporating all senses (tactile, auditory, gustatory, and olfactory) [19].

2.2 Cognitive Task Analysis

Cognitive task analysis (CTA) is a common user research approach when assessing knowledge workers in context. More specifically, researchers attempt to capture both the observable behavior and the thought processes of the user(s) as they complete their tasks [47]. This and other contextual-research techniques are rooted in current manifestations of behavior and are limited by user and researcher knowledge of current technologies, organizations, and environments. These techniques are grounded in the practice of human factors and their relation to task efficiency, job-person fit, and role activity analysis [55]. These approaches fail to take into account the potential for innovations in technology, process, and organizations [57].

Like other user research techniques, CTA is almost exclusively applied within the framework of the product development process. This process is dominated by tactical cost-benefit analysis approaches requiring specified deliverables that can be handed-off to engineers to produce a solution ready for market. Given its mechanistic historical roots, CTA is well suited to the production environment for which it is mostly used. Its strength is that it produces requirements and feedback from real users or their proxies in a relatively short time. This voice of the user is often used throughout the development process to guide development decisions and tradeoffs. Unfortunately, the context of its use is ill-suited for generating innovations given development timelines and temporal requirements for generating revenue.

From a methodological perspective, CTA is decidedly weak in terms of reliability and validity. It is rare that its context of use affords the type of investment that will permit the verification of CTA findings that qualified research scientists accept with probabilistic confidence. Even with these weaknesses, the current understanding of IDS environment requirements appears to have face validity. That is, domain experts, novices, and current solution providers agree on the current understanding of the most pressing needs of IDS users [21, 23, 10]. These requirements, however, are limited by the context of methodological use. In the current renditions of IDS user interaction models, these limitations are apparent.

Findings generated from IDS analyst CTAs should be bolstered by other methodologies for a richer understanding of the context relative to both the historical development of the task-user-environment interaction and the potential for future innovations in technology, process, and organization. This will require the types of partnerships that permit extensive access to user environments for researchers with unique theoretical perspectives and skills.

3. NEW DESIGN APPROACH

Having identified these limitations in Section 2, we believe that we can incorporate new technologies and create a novel design for a framework for an intrusion detection interface. In this section, we first describe the requirements for the visual analytics tool, and then we describe the various components of the design.

3.1 Requirements of Visual Analytics Tool

Previous work [21, 30] describing the intrusion detection task divide it into three components: monitoring, analysis, and response. When working in terms of information visualization, this is true. In order to develop a tool to support analytics as a creative process, dividing the task into four portions is more appropriate. As it relates to intrusion de-

tection, those parts would be monitoring, analysis, response, and knowledge management. As an analytic process, knowledge management is key. All exploration, alert correlation, and data viewing center on the idea of creating knowledge. More than going through a series of steps to defend their network, analysts are building a specialized base of knowledge on the subject of their network's security. As such, managing information and turning this information into understanding is key. As analysts work to understand the nature of their networking environment through exploration, they are developing artifacts, implicit and explicit, of the analytics processes. While these thought patterns have not been captured in the past, they are crucial to a successful design.

1. **Monitoring**

   The monitoring task involves capturing traffic data and being alert about the various events taking place in the environment. These events may be bandwidth usage exceeding normal thresholds, alert creation, device failure, etc. In order for an analyst to successfully monitor a network, analysts must be be able to quickly glance at a display and get a general sense of what is going on. At this level, exact numbers are not necessary, but simply that knowledge that some measure has exceeded a threshold. Conversely, if an analyst knows a specific portion of the network is vulnerable, she should be able to jump to the appropriate level of abstraction. The visual properties of the display should be fashioned in such away that the human perception system is fully utilized. Whether or not the action of observation requires an analyst's conscious attention is something the designer will have to consider. The use of pre-attentive visual properties [30] will determine how much of the analyst's attention the monitoring task will require.

2. **Analysis**

   Analysis of network events involves building an understanding of the nature of an event. The ability to explore multiple views and levels of data and aggregate multiple data sources are key requirements [30].

3. **Response**

   An effective network intrusion analytics tool requires a means to respond to the attack. There are two aspects to the response: information about the event must be recorded and appropriate network changes made. Due to the possible unintended consequences of small changes, all actions at this stage should require the analysts' full attention.

4. **Knowledge Management**

   Knowledge management is an activity which occurs during each stage of the network detection process. Knowledge management includes storing observations and experiences for future reference, collaboration, and reporting. Analysts need a means to communicate their findings to other analysts and their managers. A successful analytics tool will have the ability to store explorations so they can become the basis for training future analysts.

3.2 Multi-touch Interface

With the recent coming of age of multi-touch interface technology [5], a means of adding interactivity to data visualizations has appeared. Multi-touch interfaces are user interfaces which detect a user's point(s) of contact with the viewing surface. Touches and gestures (combinations of movements and touches) can be detected. Touch sensing technology is undergoing rapid refinement and is in constant development. A wide variety of sensing techniques have been developed making it possible to use a touch interface with almost any display technology. We envision our interface being used on a large display like those sold by Perceptive Pixel [3], Microsoft [1] and MultiTouch [2]. While the actual technology used to detect the touch is unimportant to the user, the advantages it provides will introduce a whole new level of interaction to intrusion detection analysis. The possibility that data can be explored non-linearly, and as quick as the mind can think, poses great potential to aid in network intrusion discovery. Our proposed design seeks to add a new level of interactivity to the data analysis portion of intrusion detection.

Multi-touch user interfaces offer a unique set of capabilities which may enhance the data exploration experience. Objects can be pushed, pulled, sorted, and visually arranged. The ability to detect gestures which are natural and intuitive provide the user with a mode of interaction similar to what is used in real world data exploration. For example, consider sorting through a week's worth of classified ads in search of a dilapidated piano to restore. Papers will be sorted into groups and spread into arrays. Interesting ads will be highlighted for comparison. Our interface seeks to offer the same kind of experience.

Recent work by Wobbrock, Morris, and Wilson [56] and North et. al. [43] examine the use of multi-touch for surface computing. Wobbrock et. al. investigate the building of a library of gestures for common tasks. Choosing what gestures to support when building an interface is crucial. Gestures are what make multi-touch interfaces more efficient than the traditional keyboard and mouse paradigm for certain tasks. An empirical study completed by Kin, Argawal, and DeRose [28] showed that for selection tasks "one finger direct-touch is faster than using a mouse and ... bimanual interactions are faster than using one finger." With selection and manipulation being the key actions of exploration, it is justified to investigate the use of multi-touch interfaces to interact with intrusion detection data.

By using touch interaction, our interface will enable exploration through the use of gestures to zoom, pan, and manipulate data. Because of their similarity to interaction in the physical world, we expect this form of interaction to allow analysts to easily explore network data. We recognize the limitations [48] such as sizing UI elements to multi-touch interaction to account for finger size and are currently working to build a set of widgets designed specifically for touch interaction. We also recognize multi-touch's limitations with text entry and intend our interface to be used in conjunction with a keyboard mounted near the display. By doing this, we can enhance the analysis process by providing the interaction paradigm best fit for the task.

3.3 Zoomable, Spatial Exploration

To facilitate exploration and maintain context, we have designed an interface which capitalizes on the affordances of

multi-touch. Our interface is based around the logical connection of network devices and displaying alerts in the context of the device for which they were created. We display nodes using a node-link cloud, shown in Figure 1. Devices within an organization's subnet are organized in a logical cloud. All connections originating from devices outside of the subnet are visualized as nodes around an outer ring. In order to inform the analyst these devices are external to the network, the nodes are constrained to the ring. Using visual characteristics, elements of the visualization can be made to stand out from one another. Line color and thickness are two of these qualities. Line width reflects the relative bandwidth usage and color indicates the severity of the alert on the connection. In our case we propose using red and yellow as they are commonly accepted analogs to danger and warning, respectively.

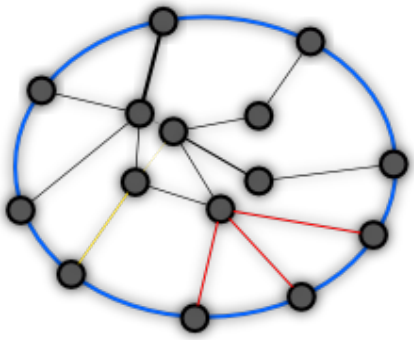

Figure 1: Node-link Cloud

We use zoom to control the metadata displayed for a particular device or alert. At the 30,000 foot level, the analyst sees nodes presenting the major segments and features of a network such as firewalls and backbones. As the analyst zooms in, network segments are split up into smaller and smaller logical units until the analyst is at the device level. Alerts are aggregated at each level of abstraction, until an alert is matched with its originating device. Bandwidth is monitored in a similar way. A bandwidth meter aggregates the sum of bandwidth across the logical units, and as the analyst moves down through the levels of abstraction, bandwidth measures are divided across their respective devices. Alerts are handled in a similar manner. A group of alerts will be placed over the appropriate logical unit and as the analyst moves to lower levels of abstraction, the alerts will be placed over the device or group of devices they correspond to. Being able to rapidly move between levels of abstraction enables analysts to check hunches, and move between an overall view of the network environment and a specific device.

When the analyst reaches the device level, packet information is made available. We provide analysts with a means of accessing the packet level information by examining a stream of packets on an interface. We conceptualize the packets flowing through an interface as a vertical stream, as shown in Figure 2. Using the idea of a lens [4], we offer a view of a user-defined number of packets. The analyst

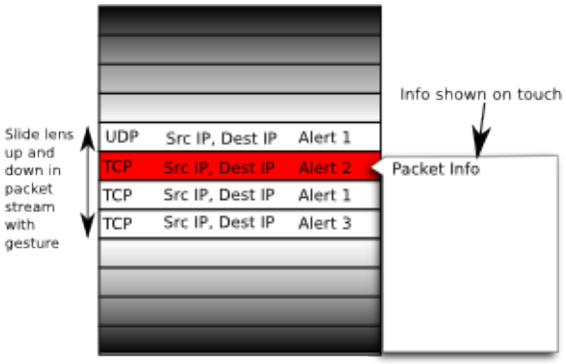

Figure 2: Packet Stream

can expand the lens and move the packet stream by using a gesture. Protocol and a brief summary are shown for all packets in the lens. Packets connected with alerts are highlighted. To access all of the packet data, an analyst touches the packet and a panel appears with the information.

The ability to mark locations for rapid return is essential. In the same way that landmarks are placed on a map, we include a feature which allows the analyst to jump to locations within the network mapping. Selecting and making a gesture marks the level of abstraction and location for future return.

At any level of zoom, analysts can pan the view. Panning and zooming are two intuitive and familiar gestures. As mentioned in Section 3.2, multi-touch interfaces afford exploration by allowing the user to move within the viewing space and organize objects. By allowing the analyst to manipulate displayed nodes and move within the logical organization of a network environment, we facilitate the exploration of network intrusion data.

3.4 Time-based Analysis

The temporal aspect of exploration must also be considered. We have designed a dial to allow an analyst to move forward and backward in time. When the dial is depressed, the display is in real-time mode. Alerts and bandwidth information are updated live. When the dial is raised, the movement through historical records is enabled. Turning the dial right or left moves the analyst forward and backward in time respectively. Alerts and bandwidth measures are updated as movements through time are made. The further the dial is turned the greater the increment of time that is moved. In this way, small steps or large leaps can be made through time. The dial snaps back to the center position once it is released. With this spring-like action, the dial can be feathered for high levels of control. Through this widget, we enable analysts to explore historical events.

3.5 Metadata Informed Architecture

After closer examination of the concept of situational awareness or context, we are led to the following observation: situational awareness is the product of metadata. An example of metadata is the lens and exposure information embedded within a digital image. While this data may not seem important, it is the data which describes the context in which the data was created. It is useful when doing comparisons or sorting information in some way. Similarly, in the domain

of intrusion detection, metadata is essential. The metadata present in the network intrusion domain are everything that describes the context of an alert. An alert signifies the presence of a data pattern, or packet, with certain characteristics. The metadata would be data such as the configuration settings of the device the alert was generated for or what other devices are connected to the given device. All of this situational information comes together to form the context of an alert. Depending on one's point of view, this data may be more important than the actual alert since it describes the context of what is happening within a network.

If metadata defines the environment of an intrusion event, then the successful exploration and management of this data is essential for understanding and responding to potential threats. Currently, the way that analysts develop a collection of this sort of data is through experience; however, this poses a problem. If all of the important information about a network is contained in an analyst's mind, it is not accessible to the less experienced. Also, if the analyst is to have a bad day or a detail does not come to mind, his response to an alert may not be appropriate. Instead, this descriptive data must be made part of the analytics tool. Creativity support tools seek to help the human mind manage an immense amount of detail in a reasonable way. In our case, we want to put the information at the tip of analysts' fingers.

An intrusion detection visualization tool needs to be more than an interactive graph. It needs to be an explorable, interrelated knowledge store. Once tools are designed from this perspective, analysts will be equipped to deal with the high volumes of network traffic currently pulsing through today's organizations. Our design seeks to provide analysts with a way to interact with alerts in light of their metadata.

There is a second use for metadata: the description of interaction. The system we are proposing uses metadata to describe a view and a set of actions on the data. Rather than moving data between analysts, a metadata description of an analyst's exploration is shared. With network traffic datasets becoming increasingly large, this approach allows an organization to invest in a central storage system for their network traffic and keep the data off the analysts' workstations. As long as the receiver has the viewing tool and access to the central traffic store, he will be able to see and interact with the thoughts of his co-workers. In this way, metadata encapsulates an analyst's process of reasoning. Visual analytics tools become more than data interaction tools, they become knowledge storehouses.

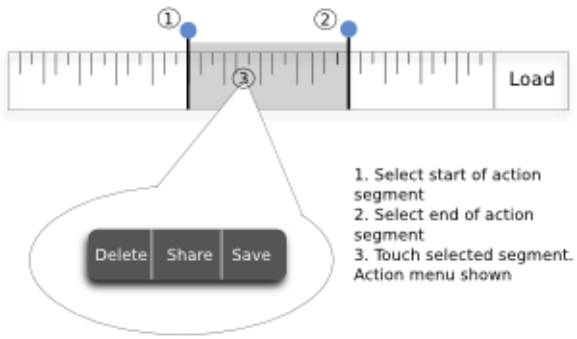

Figure 3: Action Line

Our design incorporates these two ideas in the following ways. Inspired by image graphs [33] and CzSaw's [27] history dependency graphs, we have developed what we are calling the Actionline, shown in Figure 3. It is a line of explorations described by metadata. Each change the analyst makes to the interface, or metric that is invoked, is stored on the line.

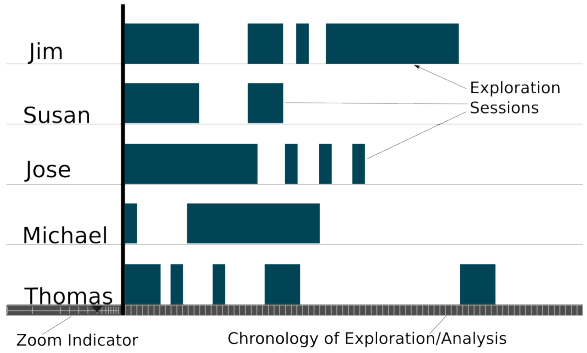

Figure 4: Repository Track View

This metadata description is stored in a repository sorted by users, shown in Figure 4. The repository is analogous to a multi-track audio recording interface, where each user is a track. Explorations are organized vertically by user and mapped along a horizontal time axis. The repository then maintains its own set of metadata (creator, creation data, etc.) about the exploration strips. By using this approach, we are able to capture explorations and provide the organization with a knowledge-base for collaboration, training, and reference purposes.

3.6 User-refined Alert Correlation

Finally, in order to assist the analyst in navigating and exploring the data, our design also incorporates alert correlation. Much work has been done in this field [60, 59, 58, 54, 53, 45, 39, 38, 37, 11, 7], however, what is largely missing is interaction with the correlation by the analyst. Our design is to take the existing work on alert correlation and modify it in such a way that it provides for this interaction. More specifically, correlation techniques have been used to construct attack scenarios [42, 41, 40, 9, 8, 6]. Much of what the security analysts do is to attempt to reconstruct all steps of the attack in order to gain the situational awareness of what is happening on the network and what potential consequences there are from these malicious actions, ultimately culminating in decisions about how to react to these intrusions. Our design incorporates these scenarios by showing the user the results of the correlation (i.e. grouping alerts and connections that are related into one attack scenario) and allowing the user to add connections and alerts to the correlated group. This can also be matched up with known attack graphs [49] to allow the user to explore what potential paths the attacker has taken.

This could also be taken one step further, in that current correlation techniques focus on relating alerts together. However, it is inevitable that certain steps in the attack are not detected by any IDS sensor and thus will go undetected, leaving the analyst to either manually find these connections or let the connections remain undetected. One way to add these connections in would be to develop a metric to

measure how suspicious each connection is, by incorporating data such as whether or not it was flagged by an IDS sensor, whether it was connected in some way to a known bad connection (e.g. shares an IP address known to have been compromised), known vulnerabilities, etc. This metric would also include input by the analyst through our interface. Fully designing this metric is left as future work.

4. ROLE OF ACTIVITY THEORY IN IDS INTERFACE DESIGN

We believe that the current approaches, while important and useful, place fundamental limits on innovation in the intrusion detection space. As an alternative, an approach based on Activity Theory can help to surpass these limits. Our design allows for interactive data exploration coupled with input from automatic analysis and correlation methods to allow the analyst to view the data in ways never before available.

The theoretical framework of Activity Theory is flexible in that it can easily adapt to a wide range of research perspectives: a basic unit of behavior such as learning to respond to simple stimuli can be described in the action to operation transition, while it can just as effectively account for the complex socio-cultural forces that influence the development of fundamental technologies (i.e. printing press).

Activity Theory proposes that human activity is goal oriented in that a person acts on an object (either mediated by a tool or not) in an effort to transform that object to some desired outcome [31]. For Intrusion Detection environments, the object is the network and a clean network is the desired outcome. Tools include IDSs, logs, email, etc. Activity theory proposes a three level structure for human behavior that includes activities, actions, and operations. In the ID domain preventing intrusions is the activity, reviewing logs an action, and in responding to a threat by a simple pointing device, interaction with an IDS an operation. At the risk of adding complexity to the classical theoretical levels of human activity, Gonzalez, Nardi, and Mark [20] propose an ensemble that is situated between actions and the activity that are "...work efforts that encompass actions, but, at the same time, lack the definitive object-related nature of an activity" [20]. These activities are often defined as collaborative and appear to be related to creating tools or artifacts that are then supportive of object transformation. For example, after a network security analyst responds to an intrusion, that analyst may document and report the attack to build the knowledge base for others to use [30] in support of the clean network.

Dekker and Woods [57] argue for the type of user research techniques that abstract the dynamics of change with new technologies to better enable the effect that new technologies have on users and their environments. Researchers have had success in understanding the historical development of technologies and tools and their inherent limitations using Activity Theory as a framework for understanding human activity [24]. We propose a multi-method approach that includes an extensive ethnography to provide a richer understanding of socio-organizational development of ID environments, tools, and technologies; additional CTAs across contexts to add validity and reliability to the current body of knowledge regarding ID analysts and their tasks; and an iterative prototyping model with leading edge user interaction and collaboration tools to generate and test hypotheses of the impact of technological change. Activity Theory and its associated research methods can provide an interpretive framework for integrating these findings into a coherent description of the ID future.

5. REFERENCES

[1] Microsoft surface. `http://www.microsoft.com/surface/`.

[2] Multitouch. `http://multitouch.fi/`.

[3] Perceptive pixel. `http://www.perceptivepixel.com/`.

[4] B. Buxton. *Sketching user experiences: getting the design right and the right design*. Morgan Kaufmann, 2007.

[5] B. Buxton. Multi-touch systems that i have known and loved. `http://www.billbuxton.com/multitouchOverview.html`, 2009.

[6] S. Cheung, U. Lindqvist, and M. Fong. Modeling Multistep Cyber Attacks for Scenario Recognition. In *Proceedings of the Third DARPA Information Survivability Conference and Exposition (DISCEX-III 2003)*, 2003.

[7] F. Cuppens and A. Miege. Alert Correlation in a Cooperative Intrusion Detection Framework. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2002.

[8] O. Dain and R. Cunningham. Fusing a Heterogenous Alert Stream Into Scenarios. In *Proceedings of the 2001 IEEE Workshop on Data Mining for Security Application*, 2001.

[9] O. Dain and R. Cunningham. Building Scenarios from a Heterogeneous Alert Stream. In *IEEE Transactions on Systems, Man and Cybernetics*, 2002.

[10] A. D'Amico and K. Whitley. The real work of computer network defense analysts: The analysis roles and processes that transform network data into security situation awareness. In *VizSEC 2007 Proceedings of the Workshop on Visualization for Computer Security*. Springer Berlin Heidelberg, 2008.

[11] H. Debar and A. Wespi. Aggregation and Correlation of Intrusion Detection Alerts. In *Proceedings of the International Symposium on Recent Advances in Intrusion Detection*, 2001.

[12] P. Dobrev, S. Stancu-Mara, and J. Schönwälder. Visualization of node interaction dynamics in network traces. In *AIMS '09: Proceedings of the 3rd International Conference on Autonomous Infrastructure, Management and Security*, pages 147–160, Berlin, Heidelberg, 2009. Springer-Verlag.

[13] R. F. Erbacher. Glyph-based generic network visualization. volume 4665, pages 228–237. SPIE, 2002.

[14] R. F. Erbacher, K. L. Walker, and D. A. Frincke. Intrusion and misuse detection in large-scale systems. *IEEE Computer Graphics and Applications*, 22(1):35–48, Jan./Feb. 2002.

[15] L. Ertoz, E. Eilertson, P. Dokas, V. Kumar, and K. Long. Scan Detection - Revisited. Technical Report 127, Army High Performance Computing Research Center, 2004.

[16] L. Ertoz, A. Lazarevic, E. Eilertson, P.-N. Tan, P. Dokas, V. Kumar, and J. Srivastava. Protecting

against cyber threats in networked information systems. In *Proceedings of the SPIE Annual Symposium on AeroSense, Battlespace Digitization and Network Centric Systems III*, 2003.

[17] F. Fischer, F. Mansmann, D. A. Keim, S. Pietzko, and M. Waldvogel. Large-scale network monitoring for visual analysis of attacks. In *VizSec '08: Proceedings of the 5th international workshop on Visualization for Computer Security*, pages 111–118, Berlin, Heidelberg, 2008. Springer-Verlag.

[18] R. Fontugne, T. Hirotsu, and K. Fukuda. A visualization tool for exploring multi-scale network traffic anomalies. In *SPECTS'09: Proceedings of the 12th international conference on Symposium on Performance Evaluation of Computer & Telecommunication Systems*, pages 274–281, Piscataway, NJ, USA, 2009. IEEE Press.

[19] M. A. García-Ruiz, M. A. Martin, and B. Kapralos. Towards multimodal interfaces for intrusion detection. In *Proceedings of the 122nd Convention of the Audio Engineering Society*, 2007.

[20] V. Gonzalez, B. Nardi, and G. Mark. Ensembles: Understanding the instantiation of activities. *Information Technology and People*, 22(2):109–131, 2009.

[21] J. R. Goodall, W. G. Lutters, and A. Komlodi. The real work of intrusion detection: Rethinking the role of security analysts. In *Tenth Americas Conference on Information Systems, New York, New York, August 2004*, 2004.

[22] J. R. Goodall, W. G. Lutters, and A. Komlodi. Developing expertise for network intrusion detection. *Information Technology and People*, 22(2):92–108, 2009.

[23] J. R. Goodall, A. A. Ozok, W. G. Lutters, P. Rheingans, and A. Komlodi. A user-centered approach to visualizing network traffic for intrusion detection. In *CHI '05: CHI '05 extended abstracts on Human factors in computing systems*. ACM, 2005.

[24] M. Hasu. In search of sensitive ethnography of change: Tracing the invisible handoffs from technology developers to users. *Mind, Culture, and Activity*, 12(2):90–112, 2005.

[25] T. Itoh, H. Takakura, A. Sawada, and K. Koyamada. Hierarchical visualization of network intrusion detection data. *IEEE Comput. Graph. Appl.*, 26(2):40–47, 2006.

[26] J. Jung, V. Paxson, A. Berger, and H. Balakrishnan. Fast Portscan Detection Using Sequential Hypothesis Testing. In *Proceedings IEEE Symposium on Security and Privacy*, 2004.

[27] N. Kadivar, V. Chen, D. Dunsmuir, E. Lee, C. Qian, J. Dill, C. Shaw, and R. Woodbury. Capturing and supporting the analysis process. In *Proceedings of the IEEE Symposium on Visual Analytics Science and Technology, VAST 2009*, pages 131 –138, oct. 2009.

[28] K. Kin, M. Agrawala, and T. DeRose. Determining the benefits of direct-touch, bimanual, and multifinger input on a multitouch workstation. In *GI '09: Proceedings of Graphics Interface 2009*, Toronto, Ont., Canada, Canada, 2009. Canadian Information Processing Society.

[29] H. Koike and K. Ohno. Snortview: visualization system of snort logs. In *VizSEC/DMSEC '04: Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security*, pages 143–147, New York, NY, USA, 2004. ACM.

[30] A. Komlodi, J. R. Goodall, and W. G. Lutters. An information visualization framework for intrusion detection. In *CHI '04: CHI '04 extended abstracts on Human factors in computing systems*, page 1743, New York, NY, USA, 2004. ACM.

[31] K. Kuutti. Activity theory as a potential framework for human-computer interaction research. In B. A. Nardi, editor, *Context and Consciousness*, pages 17–44. MIT Press, 1995.

[32] Y. Livnat, J. Agutter, S. Moon, R. F. Erbacher, and S. Foresti. A visualization paradigm for network intrusion detection. In *Proceedings of the 2005 IEEE Workshop on Information Assurance and Security*, 2005.

[33] K.-L. Ma. Image graphs—a novel approach to visual data exploration. In *VIS '99: Proceedings of the conference on Visualization '99*, pages 81–88, Los Alamitos, CA, USA, 1999. IEEE Computer Society Press.

[34] F. Mansmann, F. Fischer, D. A. Keim, and S. C. North. Visual support for analyzing network traffic and intrusion detection events using treemap and graph representations. In *CHiMiT '09: Proceedings of the Symposium on Computer Human Interaction for the Management of Information Technology*, pages 19–28, New York, NY, USA, 2009. ACM.

[35] P. Minarik and T. Dymacek. Netflow data visualization based on graphs. In *VizSec '08: Proceedings of the 5th international workshop on Visualization for Computer Security*, pages 144–151, Berlin, Heidelberg, 2008. Springer-Verlag.

[36] C. Muelder, K. liu Ma, and T. Bartoletti. Interactive visualization for network and port scan detection. In *In Proceedings of 2005 Recent Advances in Intrusion Detection*, page 05, 2005.

[37] P. Ning, Y. Cui, and D. Reeves. Analyzing Intensive Intrusion Alerts via Correlation. In *Proceedings of the International Symposium on Recent Advances in Intrusion Detection*, 2002.

[38] P. Ning, Y. Cui, and D. Reeves. Constructing Attack Scenarios Through Correlation of Intrusion Alerts. In *Proceedings of the ACM Conference on Computer and Communications Security*, 2002.

[39] P. Ning, D. Reeves, and Y. Cui. Correlating Alerts Using Prerequisites of Intrusions. Technical report, North Carolina State University, Department of Computer Science, 2001.

[40] P. Ning and D. Xu. Learning Attack Strategies from Intrusion Alerts. In *Proceedings of the ACM Conference on Computer and Communications Security*, 2003.

[41] P. Ning and D. Xu. Hypothesizing and Reasoning about Attacks Missed by Intrusion Detection Systems. In *ACM Transactions on Information and System Security*, 2004.

[42] P. Ning, D. Xu, C. Healey, and R. S. Amant. Building Attack Scenarios through Integration of

Complementary Alert Correlation Methods. In *Network and Distributed System Security Symposium*, 2004.

[43] C. North, T. Dwyer, B. Lee, D. Fisher, P. Isenberg, G. Robertson, and K. Inkpen. Understanding multi-touch manipulation for surface computing. In *INTERACT '09: Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction*, pages 236–249, Berlin, Heidelberg, 2009. Springer-Verlag.

[44] K. Nyarko, T. Capers, C. Scott, and K. Ladeji-Osias. Network intrusion visualization with niva, an intrusion detection visual analyzer with haptic integration. In *Haptic Interfaces for Virtual Environment and Teleoperator Systems, 2002. HAPTICS 2002. Proceedings. 10th Symposium on*, 2002.

[45] P. Porras, M. Fong, and A. Valdes. A Mission-Impact-Based Approach to INFOSEC Alarm Correlation. In *Proceedings of the International Symposium on Recent Advances in Intrusion Detection*, 2002.

[46] P. Ren, Y. Gao, Z. Li, Y. Chen, and B. Watson. Idgraphs: Intrusion detection and analysis using stream compositing. *IEEE Comput. Graph. Appl.*, 26(2), 2006.

[47] E. M. Roth. Uncovering the requirements of cognitive work. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 50(3):475–480, 2008.

[48] K. Ryall, C. Forlines, C. Shen, M. R. Morris, and K. Everitt. Experiences with and observations of direct-touch tabletops. In *TABLETOP '06: Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems*, pages 89–96, Washington, DC, USA, 2006. IEEE Computer Society.

[49] O. Sheyner, J. Haines, S. Jha, R. Lippmann, and J. Wing. Automated Generation and Analysis of Attack Graphs. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2002.

[50] B. Shneiderman. Creativity support tools: accelerating discovery and innovation. *Commun. ACM*, 50(12):20–32, December 2007.

[51] Snort - The Open Source Network Intrusion Detection System. `http://www.snort.org`.

[52] T. Taylor, D. Paterson, J. Glanfield, C. Gates, S. Brooks, and J. McHugh. Flovis: Flow visualization system. In *CATCH '09: Proceedings of the 2009 Cybersecurity Applications & Technology Conference for Homeland Security*, pages 186–198, Washington, DC, USA, 2009. IEEE Computer Society.

[53] A. Valdes and K. Skinner. Probabilistic Alert Correlation. In *Proceedings of the International Symposium on Recent Advances in Intrusion Detection*, 2001.

[54] F. Valeur, G. Vigna, C. Kruegel, and R. Kemmerer. A Comprehensive Approach to Intrusion Detection Alert Correlation. In *IEEE Transactions on Dependable and Secure Computing*, 2004.

[55] J. Wei and G. Salvendy. The cognitive task analysis methods for job and task design: review and reappraisal. *Behaviour & Information Technology*, 23(4):273 – 299, 2004.

[56] J. O. Wobbrock, M. R. Morris, and A. D. Wilson. User-defined gestures for surface computing. *CHI '09: Proceedings of the 27th international conference on Human factors in computing systems*, pages 1083–1092, 2009.

[57] D. Woods and S. Dekker. Anticipating the effects of technological change: a new era of dynamics for human factors. *Theoretical Issues in Ergonomics Science*, 1(3):272 – 282, 2000.

[58] Y. Wu, B. Foo, Y. Mei, and S. Bagchi. Collaborative Intrusion Detection System (CIDS): A Framework for Accurate and Efficient IDS. In *Proceedings of the 19th Annual Computer Security Applications Conference*, 2003.

[59] D. Xu and P. Ning. Alert Correlation Through Triggering Events and Common Resources. In *Proceedings of the 20th Annual Computer Security Applications Conference*, 2004.

[60] Y. Zhai, P. Ning, P. Iyer, and D. Reeves. Reasoning About Complementary Intrusion Evidence. In *Proceedings of the 20th Annual Computer Security Applications Conference*, 2004.