Nimble Cybersecurity Incident Management through Visualization and Defensible Recommendations

Jamie Rasmussen, Kate Ehrlich, Steven Ross, Susanna Kirk, Daniel Gruen, John Patterson
IBM Research
1 Rogers St, Cambridge, MA 02142
{jrasmus, katee, steven_ross, sekirk, daniel_gruen, john_patterson}@us.ibm.com

ABSTRACT

Analysts engaged in real-time monitoring of cybersecurity incidents must quickly and accurately respond to alerts generated by intrusion detection systems. We investigated two complementary approaches to improving analyst performance on this vigilance task: a graph-based visualization of correlated IDS output and defensible recommendations based on machine learning from historical analyst behavior. We tested our approach with 18 professional cybersecurity analysts using a prototype environment in which we compared the visualization with a conventional tabular display, and the defensible recommendations with limited or no recommendations. Quantitative results showed improved analyst accuracy with the visual display and the defensible recommendations. Additional qualitative data from a “talk aloud” protocol illustrated the role of displays and recommendations in analysts’ decision-making process. Implications for the design of future online analysis environments are discussed.

Categories and Subject Descriptors

H.5.2. [**Information Interfaces and Presentation**]: User Interfaces – *user-centered design*;
K.6.5. [**Management of Computing and Information Systems**]: Security and Protection – *invasive software, unauthorized access.*

General Terms

Design, Security, Human Factors.

Keywords

Managed security services, information visualization, user studies.

1. INTRODUCTION

Vigilance tasks are those that require sustained attention, in which participants typically monitor frequent, repetitive signals for uncommon or unpredictable events, and react appropriately when such events occur. [22] In the cybersecurity domain, vigilance tasks are epitomized by the work of “online” analysts in a Security Operations Center (SOC), who engage in real-time monitoring and investigation of computer and network health, with a particular emphasis on the detection and triage of problems caused by malicious people and code.

Online analysts have a difficult job, characterized by the need to integrate specialized technical knowledge with contextual knowledge under severe time constraints, often with too little or extraneous information, an abundance of false alarms, and adversaries actively seeking to prevent or mislead analysis. Worse still, failure in this task can have direct and severe consequences for the financial health and reputation of the injured party. For example, it is estimated that computer viruses alone cost businesses billions of dollars each year. [14] Furthermore, the problem has been getting worse, straining analyst and organizational resources. [8]

Our goal is to help online analysts complete their tasks more quickly and accurately. We have investigated two complementary approaches to improving analyst performance: an interactive graph-based visualization of correlated output from Intrusion Detection Systems (IDS) and defensible incident categorization recommendations based on machine learning from historical analyst behavior. Defensible recommendations are those that the system can justify or explain, supporting analysts’ capacity to understand and evaluate the relevance of the recommendations. Both incident visualization and categorization recommendations are driven from a graph-structured model created for each incident, which is initialized from raw event data and augmented with contextual and asset information.

In a managed security environment, each online analyst may be monitoring dozens or hundreds of protected networks with only a few minutes to devote to the analysis of a typical alarm. This time constraint suggests different design criteria for visualizations targeting online use. Visualizations for online analysis should emphasize simple, intuitive representations rather than information density. The salience of visual features should be determined by their utility in diagnosis and response. Manipulations of the visualization should be facile and natural.

We believe that our incident visualization, which we refer to as an "Interactive Incident Diagram" (IID), could have a variety of advantages over the tabular display of IDS events offered by many security tools. In this paper, we study the impact of a visual or tabular display on analyst performance. We also test the interplay of display method with the presence of incident classification recommendations and the ability to visualize the system’s justifications for its recommendations.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 describes the operations of one commercial SOC and data we obtained from their production systems. Section 4 describes a prototype environment we constructed to test our Interactive Incident Diagrams and the

VizSec ‘10, September 14, 2010, Ottawa, Ontario, Canada.
Copyright 2010 ACM 978-1-4503-0013-1/10/09…$10.00.

effects of combining them with defensible recommendations. Section 5 describes the design of a study with professional cybersecurity analysts; Section 6 presents that study's results. Section 7 discusses analyst feedback on the design of our environment. We conclude with suggestions for future work in Section 8.

2. RELATED WORK

Komlodi et al. [13] described three phases of work in intrusion detection: Monitoring, Analysis, and Response, identifying particular visualization needs for each. In the Analysis phase, which we are primarily concerned with, analysts require powerful interactive visualizations that fuse information from disparate data sources and allow analysts to view data at multiple levels of detail. The authors also emphasized the importance of grounding cybersecurity visualizations in analyst needs and validating proposed visualizations through user studies.

Goodall [7] applied this user-centered design approach in a study comparing user performance with two tools for analyzing network packet capture data, one based on visualization, another based on a textual/tabular display. The study involved both quantitative and qualitative methods, finding that users of TNV, the visualization tool, showed increased accuracy on well-defined tasks as well as a clear preference for the visual interface. Our goal is also to compare a visualization interface with one more commonly used, however we focus on visualizing correlated IDS output rather than network traffic. The Goodall study was conducted with novice users; our participants were domain experts, most with over five years of professional experience.

A similar study by Thompson et al. [19] compared user performance on an intrusion detection task when using command line tools versus a visualization tool. Task performance was generally better with the command line tools; the tasks were completed more quickly and users were more confident in their analysis. Although participants preferred the command line tools overall, they found several aspects of the visual interface useful, such as the ability to see overall network activity and quickly identify anomalous behavior. As in [7], the primary visualization was of network traffic information rather than correlated IDS output, and the participants were mostly students.

There have been a number of visualizations designed for IDS output. Koike and Ohno's SnortView used simple geometric shapes to indicate protocol and severity in a two-dimensional grid relating source IP address to destination IP address and time. [12] IDS RainStorm by Abdullah et al. presented a zoomable interface allowing the display of a full day's worth of IDS events across a large network. [1] VisAlert by Livnat et al. used a radial visualization with smoothly animated transitions to help analysts visually identify IDS event patterns in time, type, and network location. [15] Although both IDS RainStorm and VisAlert were tested with professional users this evaluation did not include quantitative measures of performance.

Artificial Intelligence (AI) techniques have long been applied to almost every phase of real-time cybersecurity incident processing, from initial traffic classification to automated recommendations on incident disposition. Particularly relevant to this work are systems where analysts' judgments feed future defensible recommendations. The work by Pietraszek on the Adaptive Learner for Alert Classification (ALAC) system is notable. [17] ALAC learned from analyst classifications which IDS events are true positives. This was accomplished through non-intrusive methods, by seeing which events were included in the security incident tickets the analysts created. The automated classifications could be used as suggestions for the analyst, or used to automatically ignore events classified as false positives with high confidence. ALAC used a modified rule induction algorithm, allowing for the potential interpretation and review of generalized patterns by subject matter experts.

Recommendations from AI systems are sometimes accompanied by system-generated explanations for the recommendations, which may serve multiple purposes. Among other things, explanations can increase understanding and acceptance of recommendations [11], support faster and more accurate decision making [20][4], and enhance trust in the system recommendations [18]. Vig et al. [21] distinguish explanations from justifications. Justifications may express reasoning in the form of a conceptual model significantly different from the underlying recommendation mechanism, but should serve to support user comprehension and reasoning about system behavior.

3. BACKGROUND

3.1 Managed Security Services

We worked with a managed security provider that offers a variety of cybersecurity services, including round-the-clock monitoring and management of security devices on customer networks. Online analysts in globally distributed SOCs evaluate real-time data from intrusion detection systems, systematically categorizing and prioritizing threats. Raw events from multiple IDSes are funneled through an AI engine that correlates the data streams and identifies the situations of greatest concern, providing a significant reduction of what would otherwise be overwhelming data streams. Higher level alerts from the AI system are distributed to online analysts for handling, which can include reviewing raw event and log details or examining the character of involved devices by viewing customer-provided asset data, historical activity, presence on blacklists, or the results of queries such as geolocation or WHOIS lookups.

Analysts use a ticketing system to track security incidents. Before creating a new ticket for an incident, analysts will check for existing tickets that the alert should be associated with, or for special instructions from customers that would affect the disposition of the alert. The analyst assigns a category and priority to newly created tickets. Customers are notified based on threat severity, customer preference, and the level of service contracted. Attacks with severe repercussions may result in phoning or paging a customer representative. Lower priority problems may result in email notification only. Some alerts are the result of authorized activity or of malicious activity that the targeted assets are immune to; these types of alerts are logged for accountability and can be reviewed by the customer, but do not result in notification.

In rare cases, the online analysts from the managed security provider may be authorized to take direct remedial actions on the protected network, such as updating firewall policies. In most cases, however, the online analyst is limited to a notification and advisory role.

3.2 Data Collection

We worked with threat engineers from the managed security provider to extract data from production log files and databases. We received four types of information: alerts, events, asset details, and analyst ratings. The threat engineers selected three undisclosed customers they felt were reasonably representative of their customer base. These particular customers were also chosen because they all used the same type of IDS hardware, ensuring that attack signatures would be consistent within and between customers. Information was retrieved for each of the customers for an eight day period for which the threat engineers believed there had been typical levels of activity. The AI system created a total of 164 alerts for these customers during the time period. Data from the AI system was merged with data from the ticketing system, allowing us to see how each alert was handled by the assigned analyst. SOC managers provided a table with the job role and management-assessed skill level for each of the 29 analysts who handled an alert in our dataset.

There were a total of seven hardware sensors across the three customers. From the collected alerts, we extracted a list of all IP addresses that had been either a source or a destination for an alert. This list was used to filter the raw stored data from each of the seven sensors to a manageable level. Filtering was required to avoid disk quota and processing time limits. A standard set of fields was collected from any event whose source or destination was one of the identified IP addresses. This resulted in a collection of 2,869,108 raw events.

For each of the three customers, we extracted some of the available hardware asset information. This information may have been entered by the customer through a web-based portal or added by an analyst based on customer feedback. Information for 106 critical assets was available. The asset data fields are usually semi-structured text. For example, some information regarding critical open ports was given as a number, some given as a well-known protocol abbreviation such as "HTTP". We addressed some of these issues through normalization during post-processing for use in our knowledgebase.

One of our primary concerns was ensuring that the collected data would not compromise the identity, security, or business interests of the customers it was taken from.

Automatic anonymization of plain text fields such as analysts' notes or remediation recommendations was infeasible, so these fields were not collected. Analyst names, alert identifiers, and ticket identifiers were replaced with freshly minted numeric identifiers in a consistent manner.

IP addresses across the dataset were anonymized using the prefix-preserving Crypto-PAn method. [6] The anonymization was applied using a consistent encryption key, ensuring that a given IP address would consistently map to the same output value regardless of which data source it was extracted from. The anonymization code was executed by threat engineers who disposed of the encryption key after use; researchers never received customer IP address values.

IP address anonymization had several side effects. During our study we could not provide analysts with query actions such as geolocation or WHOIS lookups, as these would have returned random values. We did not attempt to preserve special addresses such as those used for loopback, multicast, or private-use, which may have misled analysts. We also saw instances where the initial octet of the anonymized address was either a well-known or unlikely value, which may have caused analysts to be more or less concerned about an alert than they otherwise would be.

4. NIMBLE

The name of our prototype cybersecurity environment is NIMBLE (Network Intrusion Management Benefiting from Learned Expertise). The NIMBLE software reads correlated alert data from an input file, creates a semantic model for each alert, matches each alert model against historical models in order to create recommendations, and displays alerts to the analyst either visually or as tables of data.

4.1 Semantic Summarization

The AI alerts are triggered from aggregated events, but they do not contain events. Analysts use the information in the alert to dig for more information in correlated events from network and host IDSes, firewall and proxy logs, etc. Our gathered data was limited to network IDS events. Normally, an analyst would be able to request e.g. all of the IDS events that involved two machines in the previous four hours. However, in the context of our study, in which we were asking analysts to make decisions under severe time constraints, we did not want to introduce the variability in timing that a querying mechanism would have caused. Instead, we needed to choose a correlation strategy that would select a fixed set of events from our dataset to connect to each alert. The correlation strategy had to balance competing needs. If the strategy selected too many events the problem would be too hard and the analysts would not be able to complete the assigned task in the time provided. If the strategy selected too few events there might be too little information for the analyst to make an assessment, or the problem would be too easy, and the differences between analyst performances on the various conditions could be obscured by differences in reaction time.

We investigated a variety of correlation strategies before choosing one that yielded problems of appropriate complexity. The correlation strategy we ultimately chose for use in our study only included events from a relatively short time window, from the time of the first event triggering an alert condition to the time the alert was created. Events within that time window that did not involve any machine in the alert information were excluded. This strategy did not partner any event with more than one alert. The mean number of events per alert was 106.71 ($\sigma = 179.88$), with a minimum and mode of 1 event and a maximum of 863.

Semantic summarization of the set of events correlated with an alert represents the foundation for both our visualization and the NIMBLE learning and suggestion mechanism. Our goal in constructing a semantic summary is to present the analyst with a condensed version of the information that highlights the salient aspects of the event set. These condensed representations are also the basis by which the NIMBLE learning mechanism constructs models that are matched against future semantic summaries to derive suggestions.

The semantic summarization algorithm clusters a set of events into a set of partitions. Each partition can be characterized by a set of source machines, a set of destination machines, and a set of signatures with associated counts. The partition signifies that each source has communicated with each destination by each signature at least once. The counts associated with each signature represent the total number of events in the partition with that signature. The summarization algorithm abstracts away the distribution of events between the sources and destinations in a partition and the timing of the events. The summarization process partitions the events in two phases. In the first phase, events are grouped into partitions with the same source and destination. As the communication between two machines may often result in hundreds of events, often with just one or a few event signatures involved, this can immediately result in a vast reduction in the bulk of information to be dealt with. The second phase seeks to merge partitions that can be merged without violating partition semantics. In order to make that determination, we must be aware of the predecessors and successors of each machine, and the event signatures by which these predecessors and successors are connected. Partitions can be merged if they meet the following criteria:

1. Both partitions have the same set of signatures. (Counts are irrelevant for merging purposes.)

2. The sources of both partitions have the same predecessors by the same signatures.

3. The sources of both partitions have the same successors by the same signatures.

4. The destinations of both partitions have the same successors by the same signatures.

5. The destinations of both partitions have the same predecessors by the same signatures.

When partitions are merged, their source, destination, and signature sets are combined, and the signature counts for individual signatures are totaled.

The set of partitions forms the basis for a graph-structured semantic model of the alert, which is represented in terms of a modifiable OWL ontology [2] and persisted to a shared knowledgebase. Nodes within the graph represent ontological entities or literal values; edges represent claims of binary relationships between entities or between an entity and a literal value. The construction process supplements the semantic alert model with any information from the shared knowledgebase concerning the individual machines described. This information could include indications of manufacturer, operating system, network location, geographic location, owner, importance, installed applications and known services, etc.

4.2 Generating Defensible Recommendations

To make incident classification recommendations, NIMBLE calculates the similarity between the model for a given alert and historical alert models. The scoring algorithm is based on a general purpose semantic matching algorithm, which attempts to find the least-cost correspondence between two semantic models. This is a classic inexact graph matching problem. While there are many sophisticated approaches to doing this kind of matching (see for example [16]), for the NIMBLE prototype we used a simple best-first search of the space of possible correspondences between the claims. Our matching procedure is asymmetric. We wish to treat one model as a template graph, for which we seek correspondences in the other model's matching graph. Thus a smaller template model may find a good match embedded in the context of a larger matching model, and our system will have detected a target attack embedded within the context of a larger alert. As our cybersecurity ontology does not use relationship hierarchies, correspondences only need to be considered between claims involving identical properties. The cost function for matching corresponding claims depends on the sum of the ontological distance between unequal corresponding source and destination entities, which itself is determined by the percentage of classes in the ancestry of the template entity that are not found in the ancestry of the matching entity. The search finds the set of correspondences which result in the lowest cost, thus achieving the highest degree of match. The reported match score ranges from 0 to 1.0, representing the degree of match found between the two semantic graphs.

One of the benefits of this approach to recommendations is that it is possible for the system to provide a justification for the suggestions that it makes. Many machine learning approaches do not share this property. As part of the similarity calculation, we identify how the features of the current alert model correspond to the features of a similar historical model, and we can visualize this alignment to analysts curious about the reasoning behind the suggestion.

Although we used this simple case-based reasoning approach to generate incident classification recommendations for use in our study, our ontology-based models can also be aggregated and generalized to form a more succinct set of abstract rules. Justifications remain possible with generalized rules.

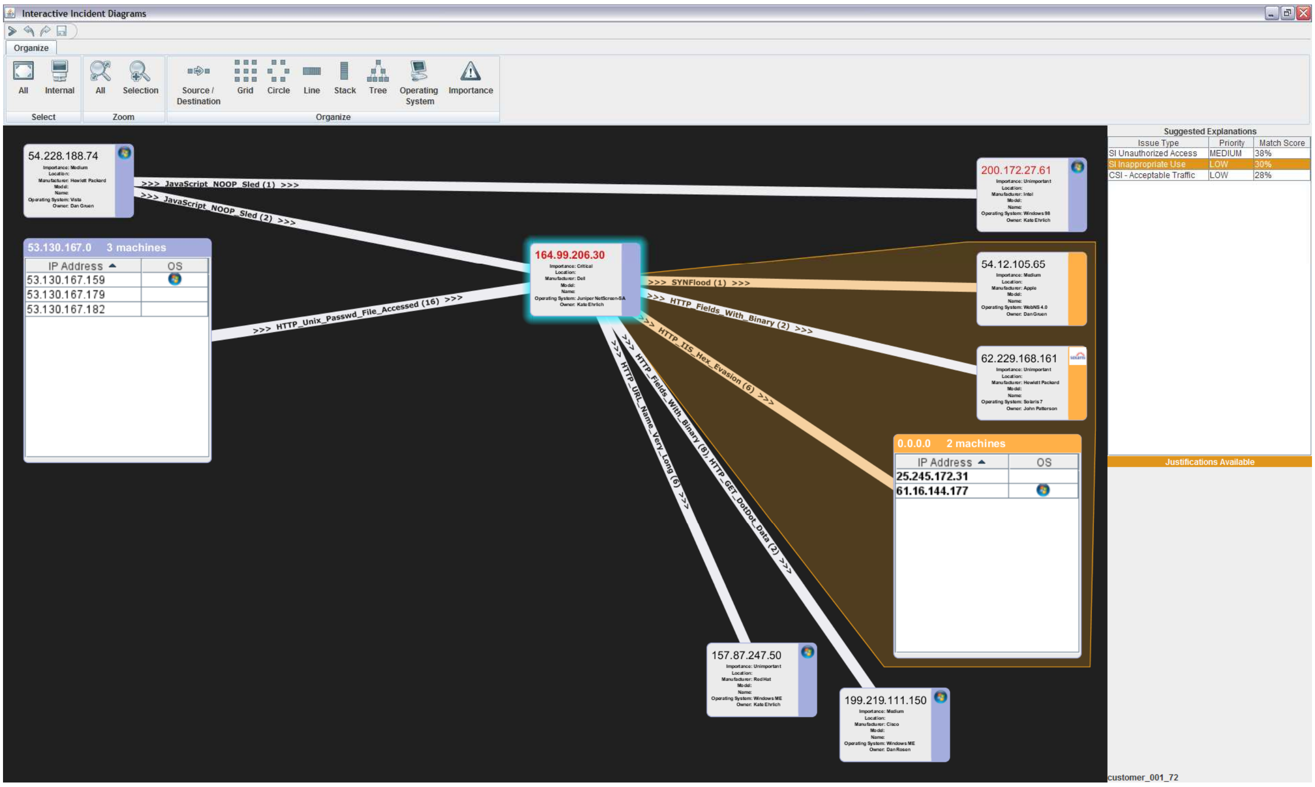
4.3 Interactive Incident Diagrams

We created two custom versions of the NIMBLE user interface for use in our study with cybersecurity analysts. One version was used in timed trials with varying experimental conditions, the other allowed exploratory interaction with our Interactive Incident Diagrams. The exploratory interface is shown in Figure 1. The visual styling and interaction capabilities of the IID were influenced by feedback from analysts on early mockups.

The diagrams are built using the zoomable scene graph capabilities of the Piccolo2D library, which allows for multiple representations of scene elements at different zoom levels, smoothly animated zooms and transitions, and rich interactivity with embedded widgets. [3]

The IID visualization is a graph in which each node or "card" represents one or more machines, and the edges connecting nodes represent sets of IDS signatures involving the connected machines. IID graphs are directly constructed from the semantic summary model of an alert. Fundamental user interactions include manual or automatic selection and arrangement of cards and smoothly zooming and moving the canvas to show cards at varying levels of detail. The visual representation of a card can change depending on the scale at which it is being viewed. For example, card text and icon decorators only appear when the zoom level is sufficient for legibility.

Single Machine cards are labeled with an IP address. If the card represents an internal machine – one that is part of the protected network – the IP address will be colored red. If the card


Interactive Incident Diagrams
Organize
All
Internal
Select
All
Selection
Zoom
Source / Destination
Grid
Circle
Line
Stack
Tree
Operating System
Importance
Organize
>>> JavaScript_NOOP_Sled (1) >>>
>>> JavaScript_NOOP_Sled (2) >>>
>>> HTTP_Unix_Passwd_File_Accessed (16) >>>
>>> SYNFlood (1) >>>
>>> HTTP_Fields_With_Binary (2) >>>
>>> HTTP_IIS_Hex_Evasion (6) >>>
>>> HTTP_URL_Name_Very_Long (6) >>>
>>> HTTP_Fields_With_Binary (8), HTTP_GET_DotDot_Data (2) >>>
Justifications Available
customer_001_72

54.228.188.74
Importance: Medium
Location:
Manufacturer: Hewlett Packard
Model:
Name:
Operating System: Vista
Owner: Dan Gruen

53.130.167.0 3 machines

IP Address ▲	OS
53.130.167.159	●
53.130.167.179	
53.130.167.182	

164.99.206.30
Importance: Critical
Location:
Manufacturer: Dell
Model:
Name:
Operating System: Juniper NetScreen-SA
Owner: Kate Ehrlich

200.172.27.61
Importance: Unimportant
Location:
Manufacturer: Intel
Model:
Name:
Operating System: Windows 98
Owner: Kate Ehrlich

54.12.105.65
Importance: Medium
Location:
Manufacturer: Apple
Model:
Name:
Operating System: WebNS 4.0
Owner: Dan Gruen

62.229.168.161
Importance: Unimportant
Location:
Manufacturer: Hewlett Packard
Model:
Name:
Operating System: Solaris 7
Owner: John Patterson

0.0.0.0 2 machines

IP Address ▲	OS
25.245.172.31	
61.16.144.177	●

157.87.247.50
Importance: Unimportant
Location:
Manufacturer: Red Hat
Model:
Name:
Operating System: Windows ME
Owner: Kate Ehrlich

199.219.111.150
Importance: Medium
Location:
Manufacturer: Cisco
Model:
Name:
Operating System: Windows ME
Owner: Dan Rosen

Suggested Explanations

Issue Type	Priority	Match Score
SI Unauthorized Access	MEDIUM	38%
SI Inappropriate Use	LOW	30%
CSI - Acceptable Traffic	LOW	28%

Figure 1: Screenshot of IID Exploration Console

represents a machine that is a critical asset, the IP address will be shown in bold. These conventions were chosen to be familiar to study participants. We considered using realistic images of the relevant hardware for Single Machine cards but this asset information was typically unavailable in our collected data.

Multiple Machine cards contain an interactive table widget where each row corresponds to a machine. The table allows row selection, sorting by column, and vertical scrolling as necessary. Multiple Machine cards are labeled to indicate the common IP address prefix and count of the machines represented.

Cards have a colored handle that allows for easy dragging for manual repositioning. Single Machine cards also use the handle area to show icons for operating system, primary machine function, or other important information that may be available. If the display is sufficiently zoomed in, Single Machine cards display other attributes such as geographic location, administrator, manufacturer, etc. as structured text. Cards that have been selected by an analyst are indicated by surrounding each card with a translucent cyan glow.

Cards are connected with labeled edges that indicate IDS event signatures and counts. The width of the edge is an indication of the total number of events it represents. The width is scaled by the natural logarithm of the event count, however a minimum size is enforced to ensure legibility of the label, which is drawn inside the edge.

Although the IID is capable of using many different layout algorithms to position cards, our default layout algorithm places machines that were the source of an event on the left side of the diagram, those that were a destination on the right side of the diagram, and those that were both a source and a destination in the center. The position of cards was adjusted to limit overlap of either cards or edges.

In order to keep the diagram simple, some information such as TCP port usage is only shown as a tooltip, displayed when the analyst hovers over an element of the diagram. Other mouse gestures and control keys provide features such as selection of one or more cards to scope subsequent commands, zooming or panning the canvas, and organization of cards either geometrically, e.g. into grids, circles, stacks etc., or by attributes such as operating system or importance.

NIMBLE provides suggested incident categorizations or “explanations” in a panel next to the alert display. When an explanation is selected from the list of suggested explanations, the IID highlights the portions of the currently viewed alert that match that explanation. The degree of match is indicated using three shades of orange. The colors mean slightly different things for cards and edges, but in both cases the darker the orange the closer the match.

For the machine cards:

Dark Orange: Exactly the same machines.

Medium Orange: Not the same machines, but the same clustering, i.e. a single machine mapped to a single machine, or many machines mapped to many machines.

Light Orange: Single machines mapping to multiple machines or vice-versa.

For the edges:

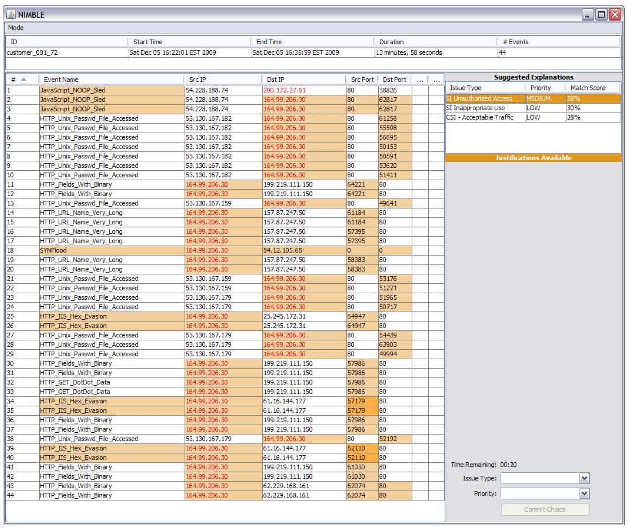
Dark Orange: Exactly the same set of signatures (but counts may vary).

Medium Orange: Some overlap in the set of signatures.

Light Orange: No overlap in the actual signatures, but the system interprets something about the event activity as corresponding to the template model. (E.g. could have been the same TCP port in both cases.)

Hovering over an orange card or edge would show a tooltip detailing the differences between the currently viewed alert and the historical alert that was the basis for the recommendation.

In the NIMBLE IID exploration interface, we tried a variation on the justification highlighting used during the study, in which we drew further attention to the matching portion of the IID by shading the background region corresponding to the convex hull of the matched nodes and edges. This region would update as nodes were repositioned. Some analysts preferred this rendering; others found the possible presence of non-matching nodes on top of the shaded region confusing. One possible improvement would be to use a Bubble Set method to avoid shading behind non-matching nodes. [5] This future mechanism could be used for emphasizing other groups of cards on demand, for example, when the analyst chooses to organize cards by operating system the added background shading could emphasize different families of operating systems.



ID	Start Time	End Time	Duration	# Events
customer_001_72	Sat Dec 05 16:22:01 EST 2009	Sat Dec 05 16:35:59 EST 2009	13 minutes, 58 seconds	44

#	Event Name	Src IP	Dst IP	Src Port	Dst Port	...	...
1	JavaScript_NOOP_Sled	54.228.188.74	200.172.27.61	80	38826		
2	JavaScript_NOOP_Sled	54.228.188.74	164.99.206.30	80	62817		
3	JavaScript_NOOP_Sled	54.228.188.74	164.99.206.30	80	62817		
4	HTTP_Unix_Passwd_File_Accessed	53.130.167.182	164.99.206.30	80	61256		
5	HTTP_Unix_Passwd_File_Accessed	53.130.167.182	164.99.206.30	80	55598		
6	HTTP_Unix_Passwd_File_Accessed	53.130.167.182	164.99.206.30	80	56695		
7	HTTP_Unix_Passwd_File_Accessed	53.130.167.182	164.99.206.30	80	50153		
8	HTTP_Unix_Passwd_File_Accessed	53.130.167.182	164.99.206.30	80	50591		
9	HTTP_Unix_Passwd_File_Accessed	53.130.167.182	164.99.206.30	80	53620		
10	HTTP_Unix_Passwd_File_Accessed	53.130.167.182	164.99.206.30	80	51411		
11	HTTP_Fields_With_Binary	164.99.206.30	199.219.111.150	64221	80		
12	HTTP_Fields_With_Binary	164.99.206.30	199.219.111.150	64221	80		
13	HTTP_Unix_Passwd_File_Accessed	53.130.167.159	164.99.206.30	80	49641		
14	HTTP_URL_Name_Very_Long	164.99.206.30	157.87.247.50	61184	80		
15	HTTP_URL_Name_Very_Long	164.99.206.30	157.87.247.50	61184	80		
16	HTTP_URL_Name_Very_Long	164.99.206.30	157.87.247.50	57395	80		
17	HTTP_URL_Name_Very_Long	164.99.206.30	157.87.247.50	57395	80		
18	SYNFlood	164.99.206.30	54.12.105.65	0	0		
19	HTTP_URL_Name_Very_Long	164.99.206.30	157.87.247.50	58383	80		
20	HTTP_URL_Name_Very_Long	164.99.206.30	157.87.247.50	58383	80		
21	HTTP_Unix_Passwd_File_Accessed	53.130.167.159	164.99.206.30	80	53176		
22	HTTP_Unix_Passwd_File_Accessed	53.130.167.159	164.99.206.30	80	51271		
23	HTTP_Unix_Passwd_File_Accessed	53.130.167.179	164.99.206.30	80	51965		
24	HTTP_Unix_Passwd_File_Accessed	53.130.167.179	164.99.206.30	80	50717		
25	HTTP_IIS_Hex_Evasion	164.99.206.30	25.245.172.31	64947	80		
26	HTTP_IIS_Hex_Evasion	164.99.206.30	25.245.172.31	64947	80		
27	HTTP_Unix_Passwd_File_Accessed	53.130.167.179	164.99.206.30	80	54439		
28	HTTP_Unix_Passwd_File_Accessed	53.130.167.179	164.99.206.30	80	63903		
29	HTTP_Unix_Passwd_File_Accessed	53.130.167.179	164.99.206.30	80	49994		
30	HTTP_Fields_With_Binary	164.99.206.30	199.219.111.150	57986	80		
31	HTTP_Fields_With_Binary	164.99.206.30	199.219.111.150	57986	80		
32	HTTP_GET_DotDot_Data	164.99.206.30	199.219.111.150	57986	80		
33	HTTP_GET_DotDot_Data	164.99.206.30	199.219.111.150	57986	80		
34	HTTP_IIS_Hex_Evasion	164.99.206.30	61.16.144.177	57179	80		
35	HTTP_IIS_Hex_Evasion	164.99.206.30	61.16.144.177	57179	80		
36	HTTP_Fields_With_Binary	164.99.206.30	199.219.111.150	57986	80		
37	HTTP_Fields_With_Binary	164.99.206.30	199.219.111.150	57986	80		
38	HTTP_Unix_Passwd_File_Accessed	53.130.167.179	164.99.206.30	80	52192		
39	HTTP_IIS_Hex_Evasion	164.99.206.30	61.16.144.177	52110	80		
40	HTTP_IIS_Hex_Evasion	164.99.206.30	61.16.144.177	52110	80		
41	HTTP_Fields_With_Binary	164.99.206.30	199.219.111.150	61030	80		
42	HTTP_Fields_With_Binary	164.99.206.30	199.219.111.150	61030	80		
43	HTTP_Fields_With_Binary	164.99.206.30	62.229.168.161	62074	80		
44	HTTP_Fields_With_Binary	164.99.206.30	62.229.168.161	62074	80		

Suggested Explanations

Issue Type	Priority	Match Score
SI Unauthorized Access	MEDIUM	38%
SI Inappropriate Use	LOW	30%
CSI - Acceptable Traffic	LOW	28%

Justifications Available

Time Remaining: 00:20

Issue Type:

Priority:

Figure 2: Screenshot of Tabular Display in Study Console

Figure 2 shows a screenshot of the NIMBLE user interface used in timed trials with analysts. The primary area of the display was filled with either the Interactive Incident Diagram for the alert or a table showing correlated IDS event details for the alert. The table of events showed a fixed number and order of sortable columns, including the event sequence number, signature, source IP address, destination IP address, source port, destination port, source asset information, and destination asset information (if any). When justifications were available, selecting a suggested explanation would shade the background of table cells in a manner analogous to the IID. Summary information such as the duration of the alert and the total number of included events was displayed in both cases.

The bottom right of the display contained a timer showing the time remaining in the trial, a drop-down list for selecting the incident category for the alert, a drop-down list for selecting the priority for the alert, and a "Commit Choice" button that allowed the analyst to signal completion of the trial. Between each trial, the user interface would enter a state in which no alert was shown. A large "Next Problem" button allowed the analyst to start the next trial when ready.

5. STUDY

Our user study tested the NIMBLE environment with professional cybersecurity analysts. The purpose of this study was to examine analysts' response to NIMBLE's visual display, its recommendation capabilities, and the visual mechanism for exposing system reasoning. Specific goals of this study included:

- Understand whether and how representing information in a visual display might affect analysts' comprehension of activity and performance on analysis tasks, relative to the more conventional tabular format for displaying such information.
- Determine how analysts might use and benefit from system-generated recommendations based on machine learning from the disposition of similar historical alerts, by comparing justified recommendations with cases where there are no justifications or no recommendations at all.

5.1 Participants

Nineteen analysts participated in the study. All had a minimum of three years experience in the job and most had worked as an analyst for over five years. Data from one of the analysts was removed from our dataset due to the analyst's lack of experience with the particular event signatures which were key to accomplishing the task.

5.2 Procedure

Each analyst was tested individually in a two-hour session. Sessions began with an introduction to the study and a detailed training on the NIMBLE test console, lasting about 30 minutes. During the training, participants had an opportunity to ask questions as they viewed an example of each of the display and suggestion conditions and completed two hands-on examples. Following the training, analysts completed 24 timed analysis trials, with a break at the midway point. They were instructed to complete each trial within two minutes and to give their best guess if they ran out of time. A chime sounded 15 sec before the end and again at the two minute mark. The alert, however, remained displayed until the analyst completed the task, even if it took longer then two minutes. The purpose of imposing a two minute limit was to mimic the limited time constraints under which analysts often operate. Pre-testing with analysts, who did not participate in the main study, confirmed that two minutes was realistic for completing the tasks.

The task had three parts. First the analyst determined the category of alert and its priority by selecting the alert category from a list of 11 items and the priority from a list of 2 items (Low, Medium). We did not provide "High" as a priority choice, as we had no examples of high-priority alerts in our dataset, so no suggested explanation could be high-priority. The analyst indicated their completion of this task by clicking on a button. They then indicated their confidence in their answer by selecting from a 5-point Likert scale ranging from "Very sure of my choice" to "Very unsure of my choice".

Analysts were asked to talk aloud during the trials about what they noticed in the displays and how they were solving the task. They were given an opportunity between trials to make additional comments and observations on the tasks and the user interface. We recorded audio from the entire session, with their permission. Individual sessions concluded with a survey in which analysts rated the value of the visual and tabular displays, suggestions and justifications, and provided general feedback and reflections on their experience. After all the analysts had completed their individual sessions, they attended a two-hour focus group to discuss their impressions of the study.

5.3 Research Variables

We tested the research goals with a fully balanced parametric design in which we independently varied 4 variables:.

- **Display**. Visual vs. tabular.
- **Recommendation**. No suggestion vs. three suggestions vs. three suggestions with justifications.
- **Suggestion Accuracy.** No correct suggestion vs. one correct suggestion among the three suggestions given.
- **Order**. First set vs. second set. The first 3 variables resulted in 12 unique conditions. Each of these conditions was presented as a complete set in a random order. The set of conditions was presented twice using a total of 24 unique items.

As participants completed each trial, the NIMBLE software logged their response, the time to complete each response, and the analysts' confidence level. These log data were converted into our primary dependent measures of a) task time, b) accuracy[1] of response and c) confidence level. The data were analyzed using ANOVA repeated measures design.

Quantitative measures from the trials and surveys were augmented by qualitative data from the audio recordings of each session and from the group debrief session at the end.

6. STUDY RESULTS

6.1 Quantitative findings

Before reporting the results, it should be noted that experimental requirements as well as privacy restrictions limited what information we could display, which could have impaired decision-making. While we provided a familiar environment and task, we were asking analysts to make decisions without access to typically available information such as event signature documentation, custom IDS event fields, web host URLs, and timestamps. Additionally, the decision to have the analysts talk aloud during the trials not only increased the overall time but probably also increased the variability of the response times. And the two-minute limit, which was less time than many of these analysts took in their regular job, may have increased the error rate.

6.1.1 Accuracy

Figure 3 below shows percent accuracy under conditions of no suggestion, suggestion, and justification for the visual and tabular display over the first and second set of trials.

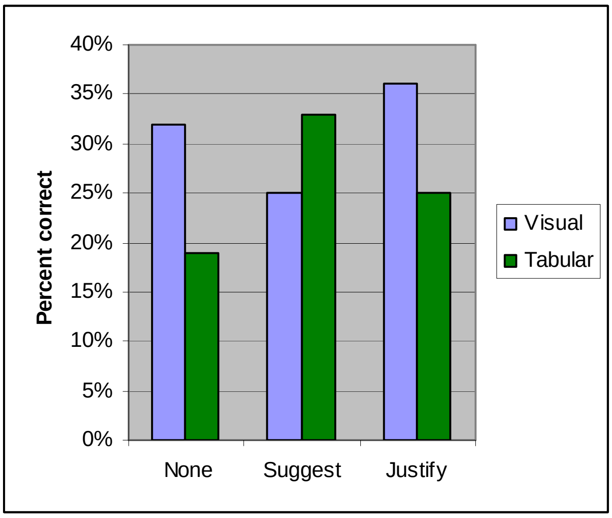


Figure 3: Percent Accuracy

Overall, the analysts were slightly more accurate with the visual display (31%) compared with the tabular display (26%) ($F_{1,17}$ = 3.2, p < 0.10). This effect was stronger in the second set of trials where accuracy for the visual display was 35% as compared with 20% for the tabular display ($F_{1,17}$ = 4.6, p < 0.05).

Across both tabular and visual displays, there was no overall difference between the three level of recommendation ($F_{2,34}$ = 1.0, p > 0.10). However, for the visual display, justification improved accuracy while for the tabular display the justification reduced accuracy ($F_{2,34}$ = 3.8, p < 0.05).

6.1.2 Response Time

The average response time overall was 85 seconds, well within the two minute period.

The average response time across all conditions is shown in Figures 4a and 4b. There was a significant main effect of order, with the second set taking less time on average than the first set ($F_{1,17}$ = 9.5, p < 0.01).

[1] The term "accuracy" is used as a shorthand to refer to agreement between the category and priority selection by the analyst in the study and the designation given to the same alert in the historical record.

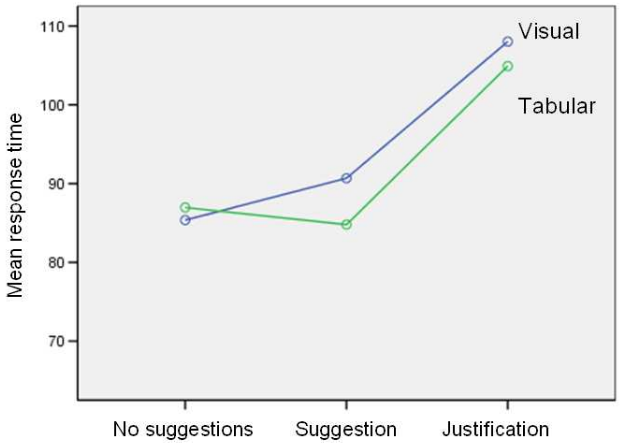


Figure 4a: Mean response time (secs) for first set of trials

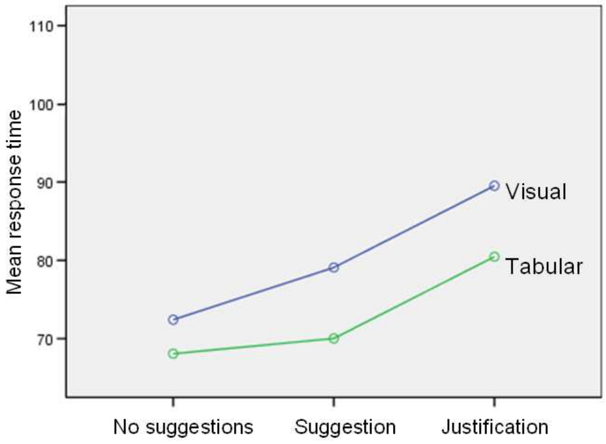


Figure 4b: Mean response time (secs) for second set of trials

There was also a main effect of recommendation with justifications taking longer to process than suggestions which took longer than no suggestions ($F_{2,34}$ = 6.7, p < 0.01). Response times for the visual displays were slightly longer than for the tabular displays ($F_{1,17}$ = 2.7, p = 0.12).

6.1.3 Confidence

There was no effect of any of the experimental conditions on confidence level. The average confidence level was uniformly high across all conditions.

6.1.4 Ratings

After the timed trials were completed, each subject was asked to rate the helpfulness of the displays and suggestions for their task, on a scale from 1 (very unhelpful) to 5 (very helpful). There was no significant difference in rating between the visual display (average rating = 3.47) and the tabular display (average rating = 3.67) (related t-test = -0.54, df = 17, p > .10). However, there was a small negative correlation between the two ratings (Pearson correlation = -0.38, p = 0.15) suggesting that analysts who liked one type of display did not like the other. There were no correlations between the ratings and performance, either in terms of accuracy or response time. In other words, analysts who gave higher ratings to the visual display were not more accurate nor faster than analysts who gave lower ratings to the visual display.

Analysts found the justifications to be significantly more helpful (average rating = 3.67) than the suggestions on their own (average rating = 3.06) (related t-test = 2.265, df = 17, p < 0.05).

There were no correlations between the individual differences (tenure, experience) and any of the ratings for display or suggestions, indicating that differences in ratings for visual or tabular was not a function of tenure or experience but of personal preference and perhaps differences in cognitive style.

In summary, analysts were able to understand and manipulate our Interactive Incident Diagrams despite their novelty, showing improved accuracy with minimal speed degradation on the incident classification task. Defensible recommendations in combination with the visual display were also associated with better accuracy.

In the next section, we turn to the qualitative data to provide further explanation for how and why the visual display and the recommendations contributed to better performance.

6.2 Qualitative Findings

6.2.1 Display

6.2.1.1 Visual Display

The Interactive Incident Diagram (IID) provides a machine-centric “picture” of event activity in a way that highlights many of the critical event relationships between source and destination machines; these relationships may be obscured in the tabular display. One of the unanticipated strengths of the visual diagram was its ability not only to represent the kind of network map some analysts reported mentally configuring, but also to support reasoning in new ways about familiar information.

“Graphically seeing the strays. You tended to see something that obviously clustered, and then you’d see other stuff out there and think, well, what is that? Then there’s one shot, off to the corner, and you think, who is that and why didn’t we see that? From looking at the tabular list, I saw it, but it was much quicker visually. The oddball stuff shows up a little better visually than it does with the table of events. That would be very useful, I think.”

Analysts pointed out particular features that they liked in the visual display, for instance:

“It was nice to see the subnet view and the ranges”

“The color coding, that's definitely helpful”

“As humans we are visual so when I am looking at a big list I am actually in some ways building that grid pattern in my mind. I have to visualize flow, direction, signatures”

Many analysts liked the visual displays, especially those who self-identified as visual thinkers. However, as noted earlier in reporting the ratings, there seemed to be clear individual differences with some people preferring the visual and others preferring the tabular. For instance, a person who preferred the tabular layout said about the visual design:

“I like the idea… But in back and forth traffic it [default layout] makes something look like a spoke – hub and spoke – but leads me

to the wrong conclusion at first glance if we are doing it quickly. There is no way of showing a 1:1 relationship there".

In summary:

- Analysts appreciated the system's clustering of events and machines.
- "Strays" or anomalies lost in large data sets stand out in the diagrams
- Some visual cues were considered insufficiently expressive, such as the use of subtle line thickness differences to represent event volume

Some IID manipulations presented challenges to efficient and effective exploration of layers of visual information. Enhancing the ease of graphical manipulations in the IID and supporting more efficient and rich functionality should amplify the benefits of working with a visual representation, improving both performance time and data exploration opportunities. In summary:

- Difficulties with zooming and panning the display slowed scanning and search for information and interfered with concurrent viewing of "big picture" and asset detail
- Node and edge distribution and font sizes were not optimized with zoom, amplifying the loss of context when zooming for asset detail
- Some manipulations were unfamiliar and tricky to control (e.g. zooming too far upon scrolling, difficulty centering zoom on a specific area)
- Making some information only available though tooltips slowed analysts down

6.2.1.2 Tabular Display

Analysts valued having access to "raw" data without any predetermined clustering or analysis.

"I felt more control in the tabular. For me, the graphic could have been more useful, but I couldn't control what was going on in the display."

The ability to view events in temporal sequence in the tabular format was also considered relevant for certain alert cases.

In summary:

- Some analysts express a greater sense of "control" and ability to manage attention by sorting columns and focusing on clusters of signatures
- Sorting also supported rapid identification of "noise" and "junk signatures"
- Ease in scanning signature names represented a significant advantage in the tabular interface
- Scrolling through sorted lists of events, analysts were better able to "get a feel" for total relative volumes of different signatures

6.2.2 Suggestions and Justifications

Because suggestion accuracy was one of our independent variables, and because we always offered exactly three suggested explanations, the presented suggestions were at times very weak matches to the viewed alert. While the system was able to illustrate these weak relationships, the analogies could be perplexing or unintelligible to analysts. Some analysts were also confused about whether the justification showed a single historical alert or a generalization of several previous alerts. Further, analysts had difficulty translating the color coding, often forgetting the meaning of each shade of highlighting. As one analyst said,

"I don't think I am trained in using the colors yet"

Analysts strongly value self-sufficiency, independent analysis, and individual judgment. Most participants in the study expressed a strong disinclination to follow a system-generated suggestion without confirming a diagnosis for themselves. There was explicit resistance to the idea of trusting in or relying on system interpretation alone.

"I would look at the suggestions, and if it didn't match my gut feeling, I would simply discard the suggestion. It became interesting when there was a justification, because then I could look, why are you suggesting this? It might be something I hadn't looked at or hadn't recognized. It might be completely bogus, but then I would see the reasoning, why are you suggesting this? What may I have missed? That's where the suggestions became valuable. If it matched my gut feeling then I would go for that option. So, the suggestion by itself was sort of worthless to me, whatever data was behind it. Only when there was a justification added, I had the intention to look at it and see, why did it come to this suggestion?"

Suggestions without justifications (unless they were of high statistical confidence) appeared to provide little support for analysts.

"The percentages never seemed really strong. If something came up and said 90% I had some comfort knowing it was there … but when it is coming up and saying things in 20s and 30s or 10s and 20s doesn't really mean much to me. That's no better than random."

Analysis of alert activity appears to occur in three phases:

- **Discovery** – becoming oriented, scanning, forming a mental model of the information available
- **Diagnosis** – reasoning about the relevance of different pieces of information, forming a hypothesis
- **Confirmation** – coming to a conclusion

Suggestions with justifications offered support, in particular, for the Discovery and Confirmation phases of analysis. Highlighting those features of the current activity which corresponded to features of previous activity served to support *attention management* in both the tabular and visual display modes, making salient for analysts the key information in the display.

In addition, after analysts had developed a hypothesis concerning the activity, they valued the suggestions with justifications as providing a "*second set of eyes*". When suggestions and justifications agreed with analyst diagnosis and reasoning, they provided confirmation, and when suggestions differed from analyst diagnosis, analysts often viewed justifications as stimulating consideration of potential alternatives.

7. DISCUSSION

In this section we consider some of the broader implications of our findings for future design of displays and recommendations for network intrusion tasks.

7.1 Displays

The complementary strengths of the visual and tabular displays suggest that analysts should have simultaneous access to both. This was something several analysts asked for directly, with one describing the value of multiple ways to view the same situation by stating:

"We might home in on one signature and automatically dismiss it, because we see it so often. A lot of our work is repetitive. You get very fast as you do it a lot, so those anomalies might slip by. I think anything like the visual/tabular thing that breaks our thought pattern up is useful."

Enhancing mechanisms for integration of views would support this need to explore information dynamically from different perspectives. As analysts focus attention on entities and relationships of interest (i.e. nodes and edges within the graphical display), they want these to serve as the mechanism through which to shift directly from one display mode to the other. They felt it would be insufficient simply to toggle between displays. Rather, analysts want to affect one by manipulating the other, for example, by selecting an arc to view the corresponding table rows, or filtering and selecting sections of a table to display as an interactive diagram. Advanced filtering capabilities may include a faceted interface for event exploration, which would additionally provide useful summary statistics for the viewed events, such as event counts by signature. [10]

Analysts call upon a wide range of information to inform their decision-making that extends beyond event signatures and the identity of source and destination machines. For privacy reasons and to maintain equality between the graphic and tabular displays we were restricted in what information could be displayed. The analysts, however, were quite vocal about what additional information they were accustomed to or desirous of, which included:

- **Asset Information** such as URLs, identification of proxy machines, and the result of queries for geolocation and WHOIS records.
- **Time Information** including the temporal sequence and pace of events, which might be displayed on a timeline that could also be used as a user control to filter events by temporal regions, or to request a visual playback of the sequence of events.
- **History**: Ability to do research on machines and alerts is critical to contextualizing current activity.
- **Signature Documentation** via a lookup feature to find signature definitions, classification, severity, and known false positives.

Analysts reported that lack of contextualizing details of this nature would make sophisticated analysis of the character and severity of a threat, including the dismissal of false positives, especially challenging.

We believe the IID could use color and region shading more effectively. The red font we used to indicate internal machines was chosen for consistency with the analysts' current environment but can be problematic for color-blind users. A continuous contour that indicated the machines that are internal to the protected network could help analysts quickly identify internal vs. external activity, and would provide a place to indicate available information about network hardware such as firewalls and the position of the source sensor(s) for events.

Analysts generally liked the IID's default layout algorithm, though we could improve layout for alerts that are particularly simple or complex. A bird's eye view may be useful when zoomed in to see detailed asset information and provide context for the current viewport position relative to the entire diagram. Conversely, a fish eye lens effect may be useful to allow detailed asset information to be visible even when zoomed out. Some analysts requested scrollbars in addition to the IID's panning mechanism when the current viewport could not encompass the entire diagram. It may be advantageous to automatically zoom the IID display in some cases, such as when new events are incorporated into a diagram, or upon selection of a suggested explanation. If more advanced navigation capabilities are provided, it would be useful to have a visual history mechanism to allow the analyst to quickly return to a previous diagram state, which would provide the additional benefit of allowing future analysts to recreate the analytical steps taken during the original analysis. [9]

7.2 Suggestions and Justifications

The current approach to representing system reasoning by means of highlighting corresponding attributes is useful. However, analysts expressed the wish to have reasoning made more expressive with the ability to toggle between highlighting matching attributes and highlighting anomalies or discrepancies against the model. Often, those elements of an alert inconsistent with a typical or characteristic case may be especially relevant for diagnosis.

Match scores provided the quantification of degree of similarity used by the system to identify the most similar cases in the knowledgebase. We were interested in discovering whether presenting match score values would influence analysts' evaluation of suggestions and justifications. However, these model similarity scores were particularly difficult for analysts to interpret. Instead, analysts want and expect something more like a confidence score, how accurate this suggestion has been in the past, or how often it has been previously accepted. If the recommendations are derived from generalizations of multiple incidents, they expect an indication of how robust the model is, how many incident diagnoses contributed to it and the detailed context around those incidents, in order to confirm the correctness of the generalization. As one analyst said:

"We're assuming that the data contributing into the suggestion presented is going to stay static, and in reality, some of those signatures can go from being false positives for a long time, making a pattern of commented security incident, and all of a sudden we get an update and now it's accurate. So, you can't base the learning on that previous data."

Analysts also wanted the ability to record comments concerning unique or important characteristics of an incident to help future analysis of similar situations. Several participants felt that the identity of the analyst working on a previous case was a strong determinant of the previous diagnosis' trustworthiness. However, it may be that as the visual language for exposing system reasoning becomes more expressive it will be able to convey enough detail that analysts shift away from their current focus on the identity and trustworthiness of the analysts behind the suggestions and become willing to evaluate the sophistication and character of the system reasoning itself.

There are several opportunities for integrating the environment we have designed for online analysis with the environments used for research and development tasks, global trend analysis, and other offline threat assessment. We interviewed several threat engineers who suggested that a variation on the IID might be useful to explicitly construct models to serve as the basis for recommendations, or to view rules generalized from analyst activity in order to vet candidates for fully automatic alert processing.

8. CONCLUSION

We have presented a two-fold approach to improving the performance of online cybersecurity analysts, combining a novel visualization of alert information with defensible incident classification recommendations generated from historical incidents of a similar nature. We evaluated the practicality of this approach by creating the NIMBLE prototype environment and testing it in a controlled empirical study with 18 professional analysts, leveraging alert data gathered from operational monitoring systems.

Our test framework and assigned task did not exactly replicate analysts' current working environment, but the participants felt they were a reasonable approximation. Analysts were able to understand and manipulate our Interactive Incident Diagrams with very little training, showing improved accuracy on an incident classification task with minimal speed degradation and no impact on confidence. It was not obvious that visualizing correlated event information would have any positive effect on performance; many of the analysts participating in our study had years of experience interpreting IDS event information presented in a tabular format.

We have discussed a number of possible improvements to NIMBLE's interactive visual display and recommendation features, such as offering multiple linked representations of alert information (e.g. visual, tabular, timeline, geographical) with integrated mechanisms for querying and filtering, or improving mechanisms for conveying recommendation relevance. It would be useful to revisit our findings with a refined NIMBLE user experience that incorporates alternate information displays or new methods for visualizing justifications.

There are also many opportunities for future study of interactive visualizations and defensible recommendations in other aspects of the online analysis vigilance task. Several analysts commented that the Interactive Incident Diagram would be helpful in training new analysts, or in communicating problems to customer stakeholders. We would like to further explore this notion of IIDs as boundary objects in synchronous and asynchronous collaboration tasks. Defensible recommendations may be useful not only for incident classification but also for other actions within the environment, such as suggesting which queries to execute or which remedial measures to propose. By attaching more explicit user metadata to recorded interactions with the environment it may also be possible to detect and compensate for analyst fatigue, or to automatically infer analysts' areas of expertise for more intelligent distribution of incoming alerts.

We found that analysts have a significant preference for recommendations that include justifications, which has repercussions for the types of machine learning algorithms that are appropriate for this use case. There are a variety of other factors whose impact on the usefulness of system recommendations bears exploration. Chief among these is the underlying accuracy of the recommendation engine, with its corresponding impact on analyst trust.

By their very nature, vigilance tasks tend to arise in the most critical of environments. These environments can be, with good reason, conservative about risk and cautious about change. We take this as a mandate for both user-centered design and the evaluation and validation of proposed improvements through both qualitative and quantitative user studies. Analysts' foremost concern is the successful completion of their mission, and they are well aware of the increasing sophistication and consequences of malicious activity. We found strong interest in both visualization and defensible recommendations as approaches for improving cybersecurity incident management, and we have high hopes that these techniques will enable analysts to respond to future threats more nimbly.

9. REFERENCES

[1] Abdullah, K., Lee, C., Conti, G., Copeland, J. A., and Stasko, J. 2005. IDS RainStorm: Visualizing IDS Alarms. In Proceedings of the IEEE Workshops on Visualization For Computer Security (October 26 - 26, 2005). VIZSEC. IEEE Computer Society, Washington, DC, 1. DOI= http://dx.doi.org/10.1109/VIZSEC.2005.8

[2] Bechhofer, S., van Harmelen, F., Hendler, J., Horrocks, I., McGuinness, D. L., Patel-Schneider, P. F., and Stein, L. A. 2004. OWL Web Ontology Language Reference. W3C Recommendation, 10 February 2004. Available at http://www.w3.org/TR/owl-ref/

[3] Bederson, B. B., Grossjean, J., and Meyer, J. 2004. Toolkit Design for Interactive Structured Graphics. IEEE Trans. Softw. Eng. 30, 8 (August 2004), 535-546. DOI=10.1109/TSE.2004.44 http://dx.doi.org/10.1109/TSE.2004.44

[4] Bilgic, M. and Mooney, R.J. Explaining recommendations: Satisfaction vs. promotion. In Proceedings of Beyond Personalization Workshop, IUI, 2005.

[5] Collins, C., Penn, G., and Carpendale, S. 2009. Bubble Sets: Revealing Set Relations with Isocontours over Existing Visualizations. IEEE Transactions on Visualization and Computer Graphics 15, 6 (Nov. 2009), 1009-1016. DOI= http://dx.doi.org/10.1109/TVCG.2009.122

[6] Fan, J., Xu, J., Ammar, M. H., and Moon, S. B. 2004. Prefix-preserving IP address anonymization: measurement-based

security evaluation and a new cryptography-based scheme. Computer Networks, Volume 46, Issue 2 (October 2004), 253-272, Elsevier.

[7] Goodall, J. R. 2009. Visualization is Better! A Comparative Evaluation. Proceedings of the Workshop on Visualization for Computer Security (VizSec), IEEE Press, 2009, 57-68.

[8] Goodall, J. R., Lutters, W.G., and Komlodi, A. 2004. The Work of Intrusion Detection: Rethinking the Role of Security Analysts. Proceedings of the Americas Conference on Information Systems (AMCIS), AIS Press, 2004, 1421-1427.

[9] Gotz, D. and Zhou, M. 2008. Characterizing Users' Visual Analytic Activity for Insight Provenance. IEEE Visual Analytics Science and Technology (VAST), Columbus, Ohio (2008).

[10] Hearst, M. 2008. UIs for Faceted Navigation: Recent Advances and Remaining Open Problems. Proc. 2008 Workshop on Human-Computer Interaction and Information Retrieval.

[11] Herlocker, J. L., Konstan, J. A., and Riedl, J. 2000. Explaining collaborative filtering recommendations. In Proceedings of the 2000 ACM Conference on Computer Supported Cooperative Work (Philadelphia, Pennsylvania, United States). CSCW '00. ACM, New York, NY, 241-250. DOI= http://doi.acm.org/10.1145/358916.358995

[12] Koike, H. and Ohno, K. 2004. SnortView: visualization system of snort logs. In Proceedings of the 2004 ACM Workshop on Visualization and Data Mining For Computer Security (Washington DC, USA, October 29 - 29, 2004). VizSEC/DMSEC '04. ACM, New York, NY, 143-147. DOI= http://doi.acm.org/10.1145/1029208.1029232

[13] Komlodi, A., Goodall, J. R., and Lutters, W. G. 2004. An Information Visualization Framework for Intrusion Detection. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems* (Vienna, Austria, April 24 - 29, 2004). CHI '04. ACM, New York, NY, 1743. DOI= http://doi.acm.org/10.1145/985921.1062935

[14] Leeson, P. and Coyne, C. 2005. The Economics of Computer Hacking. Journal of Law, Economics and Policy 1(2) 2005: 511-532.

[15] Livnat, Y., Agutter, J., Moon, S., Erbacher, R. F., and Foresti, S. 2005. A Visualization Paradigm for Network Intrusion Detection. Proc. IEEE Workshop on Information Assurance and Security. IEEE CS Press, 2005, pp. 92-99.

[16] Luo, B. and Hancock, E. R. 2001. Structural Graph Matching Using the EM Algorithm and Singular Value Decomposition. IEEE Trans. Pattern Anal. Mach. Intell. 23, 10 (Oct. 2001), 1120-1136. DOI= http://dx.doi.org/10.1109/34.954602

[17] Pietraszek, T. 2004 Using Adaptive Alert Classification to Reduce False Positives in Intrusion Detection. In Recent Advances in Intrusion Detection (RAID2004), volume 3324 of Lecture Notes in Computer Science, 102-124, Sophia Antipolis, France, 2004. Springer-Verlag.

[18] Pu, P. and Chen, L. 2006. Trust building with explanation interfaces. In Proceedings of the 11th international Conference on intelligent User interfaces (Sydney, Australia, January 29 - February 01, 2006). IUI '06. ACM, New York, NY, 93-100. DOI= http://doi.acm.org/10.1145/1111449.1111475

[19] Thompson, R. S., Rantanen, E. M., Yurcik, W., and Bailey, B. P. 2007. Command line or pretty lines?: comparing textual and visual interfaces for intrusion detection. In Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (San Jose, California, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, 1205. DOI= http://doi.acm.org/10.1145/1240624.1240807

[20] Tintarev, N. and Masthoff, J. 2007. A Survey of Explanations in Recommender Systems. In Proceedings of the 2007 IEEE 23rd international Conference on Data Engineering Workshop (April 17 - 20, 2007). ICDEW. IEEE Computer Society, Washington, DC, 801-810. DOI= http://dx.doi.org/10.1109/ICDEW.2007.4401070

[21] Vig, J., Sen, S., and Riedl, J. 2009. Tagsplanations: explaining recommendations using tags. In Proceedings of the 13th international Conference on intelligent User interfaces (Sanibel Island, Florida, USA, February 08 - 11, 2009). IUI '09. ACM, New York, NY, 47-56. DOI= http://doi.acm.org/10.1145/1502650.1502661

[22] Warm, J. S., Dember, W. N., and Hancock, P. A. 1996. Vigilance and Workload in Automated Systems. In R. Parasuraman & M. Mouloua (Eds), Automation and Human Performance: Theory and Applications, 183-200. Mahwah, NJ: Erlbaum.