Pianola - Visualization of Multivariate Time-Series Security Event Data

Alistair Thomson

ECS Security Ltd
Edinburgh, UK
e-mail: alistairthomson@ecssecurity.co.uk

Martin Graham and Jessie Kennedy
Institute for Informatics and Digital Innovation
Edinburgh Napier University
Edinburgh, UK
e-mail: {m.graham,j.kennedy}@napier.ac.uk

Abstract— **Monitoring log files for network intrusions is unwieldy. To build a mental model of the log, an analyst is required to recognise continuous timelines and attack patterns from a dataset that is essentially limited to an ordered list of events. Information Visualization techniques arrange data into directly perceivable visual patterns that may alleviate some overheads associated with interpreting these datasets and improve the ability of users, especially those in resource-stretched Small and Medium sized Businesses (SMBs), to make sense of activity patterns in Intrusion Detection System (IDS) event logs. To this end, we discuss existing network security visualizations for IDS logs and after examining the strengths and drawbacks of those applications we have prototyped a visualization tool, Pianola, that arranges events on multiple timelines to reveal patterns both in time and across a network. The tool was evaluated against the traditional use of command-line interface (CLI)-based tools for analyzing network security events and displayed significant improvements in both recognition and detection of attacks and reduction in the users' subjective workload, measured using the NASA Task Load index (TLX).**

Keywords-information visualization, security visualization

I. INTRODUCTION

The near ubiquity of internet-enabled devices in the workplace serves as a powerful attack vector for cyber criminals. As attack methodologies have evolved, it has become increasingly hard to maintain information security in the workplace. Organizations of all sizes come under attack daily, and recent research indicates an increase in cyber attacks on SMBs, with 50% of targeted attacks last year focused on them, and 18% on small businesses [1]. SMBs must improve their defensive capabilities to hope to withstand this onslaught.

An accepted best practice in information security is to employ layered preventative, detective and reactive controls to secure confidentiality, integrity and availability of resources within the domain [2]. This strategy is known as "defence in depth", of which Intrusion Detection (ID) is a key component. ID is used extensively by larger organizations but it requires a degree of time and investment in staff that smaller organizations may consider to be beyond their means.

An automated Intrusion Detection System (IDS) such as Snort [3] is able to monitor all Local Area Network communications for suspicious behaviour that has evaded other controls. Because the IDS operates passively, it can be configured to trigger alerts aggressively and at low thresholds, without causing degradation of service to network users. This has the benefit that attacks are more likely to be detected. The drawback is that many false-positive alerts are generated by the system, since it has only limited ability to determine the context of suspicious behaviour. It is not unusual for an IDS to generate thousands of discrete security events [4] which must be contextualized by the IDS operator each day.

Each IDS alert may contain over a dozen data dimensions. The scale and dimensionality of the IDS log means that it can be difficult for the operator to perceive patterns in the data. Unix pattern matching and text file manipulation programs such as sed, awk, grep and uniq, allow an analyst to usefully summarize and filter IDS event data, but cannot easily describe the continuously evolving data in a coherent (and current) way.

To perform optimally, the security analyst should be able to quickly perceive the state of their network, and investigate patterns of misuse from the IDS log data. The task of making sense of these data files is considerable, and can lead to cognitive overload, even amongst seasoned analysts [5]. Clearly an "information gap" exists between the data produced and the information needed by the analyst to make timely, informed decisions [6].

Information Visualization (InfoVis) [7] is a known strategy for contextualizing and finding patterns in large quantities of data. Our research shows that IV techniques can be successfully employed to present IDS event data interactively in such a way that facilitates monitoring and analysis, while reducing cognitive load.

II. BACKGROUND

A. Data Description

An IDS alert is a discrete event in time. It contains a number of attributes, most commonly:

Date, Time, Source IP, Source port, Destination IP, Destination port, Protocol, Priority, Classification, Message.

As demonstrated in Fig. 1, when a number of alerts are recorded in a log file, each line can be described as a discrete event mapped on a vertical "ordinally scaled time-axis [8]", meaning that the sequence of events is clear, but that the interval period between events is not explicitly encoded in the presentation of the data.

Using 12-point text on a 1600 x 1000 monitor, it is possible to present only ≃75 events on screen. Text must be read serially and so for the operator to get an overview of the data, they must individually process and then hold in memory each event from the log file. In addition to this, the log file must be regularly re-examined for the operator to keep on top of the latest events. To reliably perform such a task on a log file consisting of thousands of events is beyond the capabilities of most human beings.

Figure 1. Typical visual representation of the event log format. Each event's details are represented as one row, and the rows are ordered in time.

Statistical analysis and filtering methods are therefore extensively employed amongst IDS operators [5] to make sense of the data. The user may assemble an SQL query or a sequence of Unix commands to discover which host received the most attacks in a given period, or to isolate all alerts for a given attack type. The computer will return the relevant subset of the data, but the temporal/relational context of the events is lost. There is also a temporal and cognitive overhead associated with assembling commands to filter data in this manner.

B. The Intrusion Detection Process model

The human process of ID can be divided into 3 stages: Monitoring, Analysis/Diagnosis, and Response [9], although the monitoring and analysis phases may be hard to separate in practice since they are often conducted in tandem [10]. When thinking about this model, it is important to bear in mind that, depending on the operator's job function and the size of the organisation, the operator may have additional tasks to fulfil alongside ID. Since most businesses are small or medium-sized it can be reasoned that most operators monitoring network security for their firm only perform this role as one part of their overall job.

1) Monitoring. The monitoring phase is largely passive. The operator (constantly or intermittently) monitors the IDS log, identifying patterns and alerts that may warrant further analysis.

Certain elements of the data are of key importance to the operator at this stage: event severity, target and time, or "the W3 of What, Where, When" [11]. These represent the minimum dimensionality that can be known in order to start building an actionable model of the situation (the date, time, IP and priority fields in the IDS alert logs).

2) Analysis. During the analysis stage the operator will actively explore, filter and correlate alert data [9]. The operator extracts meaning from the data and improves the "resolution" of their mental model. Expert knowledge is employed by the operator to build a narrative of events, and potentially to correlate them against previously encountered attack scenarios.

3) Response. During the response stage, any confirmed attacks must be documented, categorized, and remediated according to the organization's operational policy. This process is largely procedural.

III. Related Work

Exploration of temporal data is a common theme in InfoVis [12] with applications in domains as diverse as genealogy [13] and text analysis [14], though here we concentrate on applications tailored to network security analysis and the domain-specific problems they attempt to tackle.

Shiravi et al. [15] proposed a taxonomy of network security visualizations based on use-case. They identified 35 security visualizations from the literature, of which 7 focus on visualizing attack patterns. Of these, 4 attempt to explicitly map the temporal dimension of IDS alerts:

A. SnortView

SnortView [16] uses a source-time matrix to plot IDS events over time. Colour is used to indicate event severity, and network protocols are given distinct glyph identifiers. Events are plotted at regular intervals on the timeline, and where two events occur in the same place, the glyph colour is changed to a blend of the two event colours to show that occlusion has occurred. Some statistical information is overlaid with coloured bars on the horizontal and vertical axes.

The visualization clearly shows patterns of events over time, and it successfully displays several dimensions of data on the overview, however its use of distinctive glyphs reduces the ability of the operator to preattentively absorb the information. A major limitation of SnortView is that it is only capable of displaying 40 alerts at any time, and it has limited capability to mitigate occlusion.

B. IDS Rainstorm

IDS Rainstorm [17] takes a novel approach to displaying IDS alert patterns on large-scale networks. The tool was developed for Georgia Tech, whose campus occupies a very large address space (equivalent to 2.5 Class B netblocks). IDS Rainstorm represents time, source and destination IP addresses and alert on a single display by mapping the entire address range in consecutive vertical columns. The width of each column is representative of 12 hours, and an alert is a single coloured pixel. This pixel-oriented technique [18] means the main display of IDS Rainstorm is close to a heat map in appearance. It effectively highlights patterns of activity over time within the vast address range. The user

may zoom in on a smaller block of alerts to investigate further and gain resolution. A drawback of the interface is that the alert colour-coding (green, yellow, red) is susceptible to colour-blindness effects, plus the secondary issue that pure yellow and green are perceived as brighter than red and thus the lower-status alerts may actually draw operation attention more readily. Additionally, the tool displays every IP address on the network, regardless of whether the address has associated alerts, or even whether it is active.

C. Vizalert

Livnat et al. developed Vizalert [19] to map alert data onto a radial event chart. Internal hosts are mapped onto a link node graph at the centre of the visualization, and alert types are grouped into areas around its circumference. Increments of time are described as concentric rings on the outside of the visualization. Vizalert is one of the few tools reviewed that correlates security event data from additional sources such as Windows event logs. These correlations can be helpful to the operator, improving decision-making capabilities.

Because the temporal dimension is depicted radially, the display would become unintelligible if all events were depicted in full. To mitigate this, most events are summarized, allowing only the most recent events to be reproduced in full. Even so, the display area would quickly become congested if, for example, a single host triggered a large number of distinct event types.

D. SpiralView

SpiralView [20] is a novel means of enabling the operator to see temporal patterns and anomalies. As with Vizalert, events are plotted onto a radial chart, but in this case one day is plotted as a single full revolution of the circle, in the manner of a clock. The tool has 31 concentric circles, each one representing a day and with the outer circle being most recent. In this way a whole month of event data is plotted, and events occurring at similar times over the month are quickly revealed. Statistical data is integrated with the main display through interactive bar and radial charts, providing a rich investigative landscape.

SpiralView's strength is in identifying patterns that recur at the same time daily, over extended time periods. It would be most useful for gaining a long-term overview of the state of the network, rather than day-to-day monitoring of IDS alerts. For intra-day monitoring, while there could be space for displaying alerts on the outer rings, it is doubtful whether a large number of alerts would compress onto the inner timelines with any clarity.

E. Summary

Network security visualization is still in its infancy [4]. A number of innovative visualizations have been developed over the last decade, but few have achieved traction in industry. The applications surveyed here are clearly valuable contributions to the field, but each has its individual drawbacks such as scalability in terms of number of alerts or events, relevance of information displayed, low-level perception issues, or focus on timescales that are not applicable in the day-to-day managing of network security. Analysts need tools that (a) have low cognitive overhead, (b) can be easily integrated into their workflow, and (c) supply the analyst with actionable information [9; 15]. It is the aim of Pianola to incorporate these elements

IV. Design

There is a body of thought [21; 9; 22] suggesting that these adoption issues can be resolved by employing user-centred design (UCD). UCD is an approach that puts the needs of the end-user at the centre of all design decisions. Its principles have been successfully employed in similarly mission-critical areas to improve operator performance [23].

A. Task Analysis

Task analysis is a fundamental part of UCD. Komlodi et al. [9] describe a task analysis study conducted with 16 ID expert participants, and their findings formed the basis of a process model for ID. They describe the analysis task requirements for each phase of the ID process, and suggest visualization methods to support them. These are briefly recapped below in Table I:

TABLE I. Phases of Komlodi *et al.'s* process model with associated visualization strategies.

Monitoring	Simple, overview of the data, pattern and anomaly recognition.
Analysis	Multiple views, zoom, drill-down, linked views, filtering.
Response	Suggest course of action, reporting, annotation, saving views.

Our goal of finding event patterns fits into the monitoring and analysis phases of this model, and the listed visualization techniques dovetail Shneiderman's visual information-seeking-mantra: "Overview first, zoom and filter, then details on demand" [24].

B. Attribute Mapping

Following Frank's time model [25], Muller and Schumann [8] suggested an approach to categorizing data as a precursor to creating a time-based visualization. From this, some key parameters were identified as being necessary to create a real-time IDS visualization:

Firstly, every IDS event is of individual significance, so each event should be visually indicated on an "event map" that shows the evolution of data over time. Secondly, the display must incorporate a dynamic representation of time; and time should be represented on a continuous line, rather than cyclically.

Finally, the IDS data are multidimensional. The principal idea of the project is to present a minimum of three dimensions of event data at any time. These are the "What, Where, When", i.e. the severity, target and time of the event. These three elements form the foundation for the visual display of security event data and should be visible at all times and assigned to the most prominent and appropriate visual attributes. Given this simple rule, we can assign the continuous data attributes of time and IP address to x,y

positions, and the categorical data attribute of severity to size, as described in Table II.

It should also be possible for the user to reveal data patterns by interactively layering additional dimensions through other visual attributes onto the event map. E.g., colour can be employed as a means of selectively layering such dimensions onto the basic visualization without diminishing the three fundamental dimensions' salience.

TABLE II. MAP OF EVENT DATA TO VISUAL ATTRIBUTES.

What	> Priority	> Glyph size
When	> Date / time	> Position on horizontal axis
Where	> Destination IP	> Position on vertical axis

C. *Pianola Implementation*

Following these design principles, we constructed a prototype tool for the visualization of IDS data which we named Pianola – after the automatic piano player whose sheet music closely resembles our prototype's main interface in appearance.

The tool was built using the Processing language [26] as it is suited for rapid prototype development, and we can take advantage of a range of reusable libraries and widgets for visualizations that have been built for Processing-based applications. At a lower level, a number of Python scripts were written to parse data logs for use by both Pianola and a command line IDS tool (used in a comparative evaluation).

Pianola's user interface can be seen in Fig. 2, and is divided up into a number of panels. A main panel (the 'Event Map') showing the timeline of events sits in the upper right-hand corner, and a number of smaller subsidiary panels showing related information are arranged on the left-hand side and bottom border.

Pianola's Event Map consists of a live event chart, with destination addresses ordered on the vertical axis and time represented on the horizontal. Each event in the log is represented as an ellipse glyph, and mapped on screen according to the target host and the time of attack. Fig. 3 shows a diagram of a single destination host's timeline:

Figure 3. Timeline of events on a single host. Time runs continuously from left to right and events are represented as circles; larger radii indicating more serious events in security terms

Each host's timeline is updated at two minute intervals, with new alerts appearing from the right-hand edge of the display (B), but although the display is in constant motion, each event is anchored in time. Destination hosts are displayed in descending order according to the number of events each has received. This acts as a visual cue to help the operator to quickly assess the situation.

Attacks may be classified as being of low, medium or high severity (priority). This is visually represented using small, medium and large ellipses (A,B,C), a method that

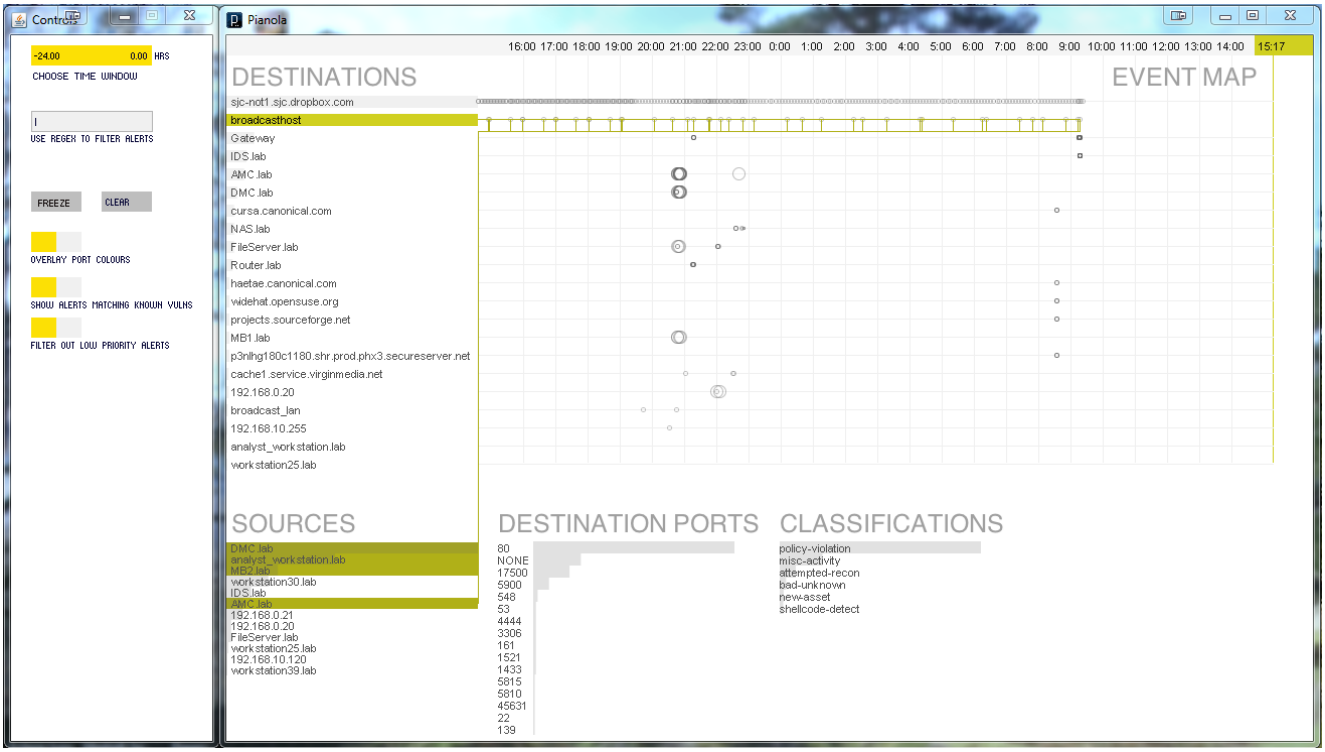

Figure 2. The main display of Pianola, highlighting events connected to one destination – 'broadcast host'. 'Event Map' is the main interface component, showing the timeline of multiple hosts on the network. Smaller sub-panels reveal information on Destinations, Sources, Ports and Classifications, and are linked to events in the Event Map through user actions such as selection and brushing. Finally, a control panel allows filtering operations and overlaying of colours on port data.

enables the user to quickly identify high priority alerts.

1) Statistical panels. Surrounding the Event Map are four panels, each containing statistical information on a single dimension of the data (destination hosts, source hosts, destination ports and snort classifications). Data points within the panels are ordered according to how frequently they appear in the event log and frequency is indicated by light grey bars underneath the text, thus helping the operator to identify trends and outliers.

When the user's mouse hovers over a data point, paths visually link it to other data points, thus revealing relationships in the data. This allows the operator to build on his "mental model" of the situation through exploratory analysis.

A user may select an element of statistical data by placing the cursor on it, upon which every matching event, source and destination host becomes highlighted on the event map. Fig. 2 shows the user selecting the "broadcasthost" timeline. The screenshot shows that several source hosts triggered broadcast alerts. The pattern of these events can be clearly seen, and also understood in the context of the broader event landscape. This operation would not be possible to achieve using other tools, but is typical of the exploratory possibilities that are facilitated by Pianola.

Finally, if the operator hovers the mouse over an event bubble, all recorded information about the event(s) is displayed in a message box beside it.

2) Global controls. A control panel containing global control widgets is situated to the left of the main panel (Fig. 2). The time axis can be zoomed and panned or constrained using a range slider. This is more flexible than a traditional slider, allowing the user to focus on any section of the timeline to facilitate more indepth analysis when necessary.

Dynamic filtering of the data can be easily achieved by entering search strings into the "filter" field; the display updates, hiding all events matching the filter criteria. The filter also takes regular expressions as input.

A toggle switch colours event bubbles according to their values for a given dimension, and also colours the selected bar chart for that dimension; transforming it into a visual key. This creates a quick visual overlay of the statistical information. The feature was implemented for the "Destination ports" chart, and could easily be applied to the other charts. Fig. 4 shows the feature in effect. Finally, the user can choose to hide low-priority alerts using a simple toggle switch. These filters can focus the display, removing nonessential data.

3) Minimizing occlusion. Often, many alarms occur in a very short period of time. One of the major challenges was figuring out how to present the alarms without causing visual occlusion on screen. If individual data points are hidden, the user's perception of the data can become distorted [27].

Pianola's default behaviour is to present event glyphs as outlines: making it possible to distinguish, as seen in Fig. 2, when a medium priority alert has occurred just before a high severity alert (C). The outlines are also drawn with a degree of translucency, so if many alerts of the same type occur on a destination host in a very small period of time, the overplotting of the ellipses results in a darker area of the display to indicate such an event (B). Occlusion can also be tackled through the use of the range slider in the global controls: by zooming in to a shorter period of time, closely bunched events can be resolved more easily.

Also, naively drawing paths as straight lines caused a great deal of occlusion on the event panel, which was allayed with a simple orthogonal routing technique which drew shared paths along the edge of the event map. Whilst not as aesthetic as other edge bundling techniques [28] it keeps the paths off the event map as much as possible and requires no iterative calculating. Together, these features allow the user to navigate the data efficiently and to see relationships that may otherwise remain hidden.

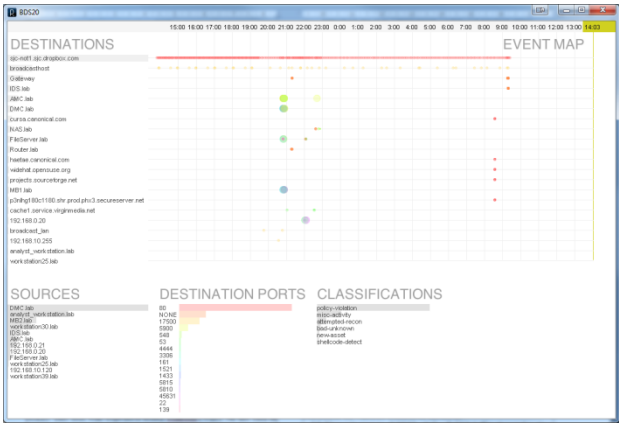

Figure 4. Port numbers encoded with colour, and overlaid onto events.

V. EVALUATION

An evaluation study was designed to measure user performance, situational awareness and workload when using Pianola. The study consisted of a repeated measures experiment, where subjects analysed IDS alert data using a suite of text extraction tools and scripts on a command-line interface (CLI). The test was then repeated using Pianola (VIZ). At the end of each treatment, the subject completed a subjective workload questionnaire. Practice effects were minimised by using a different log file for each treatment; each log contained similar events and patterns, but occurring at different times.

A sample group of 9 people was assembled from within the University. The group comprised 1 undergraduate, 1 graduate, 6 postgraduates, and 1 research fellow. They all had studied Intrusion Detection (ID) at university and all had experience with Unix command-line text manipulation tools that are commonly utilized for ID work.

There is a shortage of publicly available, well-annotated network traffic dumps that contain malicious traffic. Many organizations are naturally wary of releasing network traffic captures, even in anonymised form. The venerable DARPA datasets, dating from 1998-2000 were considered to be past their sell-by-date, and so a decision was therefore made to create a synthetic dataset for the purposes of testing the

program. A Class C laboratory network was built using physical and virtual machines and supplemented with a small honeynet. This created a flexible model on which different scenarios could be tested, and for the purpose of the evaluation generated an example dataset of nearly 1,000 suspicious events over 24 hours on a small network of 20 destination addresses.

A. Timed test

The timed test was designed to measure operator performance and situational awareness. During the test, the subject was tasked with answering 9 questions in 10 minutes:

1. Can you see any patterns or clusters in the activity?
2. If so, how many distinct patterns can you identify?
3. Which host on the network is regularly "calling home"?
4. Based on the log, which hosts would you monitor more closely?
5. Which host was the subject of the most security events?
6. Which hour showed the most activity?
7. Did most of the scanning activity occur during work (9-5) hours?
8. Which host was the source of most of the network scans?
9. Did a successful attack occur? Give details of your initial analysis.

Since Pianola was designed for smaller organisations, time was deliberately limited to simulate the real-world experience of a Sysadmin who has additional responsibilities in addition to ID.

B. Timed test results

Six of the nine questions asked (3, 5, 6, 7, 8 & 9) were Boolean in nature, in that there was a definite and singular correct answer to each. The answers to question 9 were filtered and normalised, so for instance, an answer of "No", "Maybe", or "Yes" did not score, but an answer of "Yes, NAS.lab at 23:00" scored positively, since it was a reasonable answer given the available evidence. The other three questions (1, 2 & 4) were open-ended in terms of the responses that could be supplied so these were omitted from the test for correctness of answers, though they supply useful information. This allowed the answers to these six questions to be aggregated over the user group, as shown in Fig. 5, and compared under each condition (VIZ or CLI) with various statistical methods.

McNemar's test ($p=0.0023$, $chi2 = 9.333(1df)$) showed a clear significant difference between the overall results for the two treatments. A paired t-test between the individual evaluators' performance in the two conditions indicated a significant improvement in performance under the VIZ condition ($p = 0.028$, 2-tailed). Similarly, the evaluators registered significantly more hosts on a watch list under the VIZ condition ($p=0.008$, 2-tailed).

A similar comparison per question (aggregating over the users as in Fig. 6) showed correct answers were given significantly more often in the VIZ condition than with the Command line Interface ($p < 0.01$, 2-tailed).

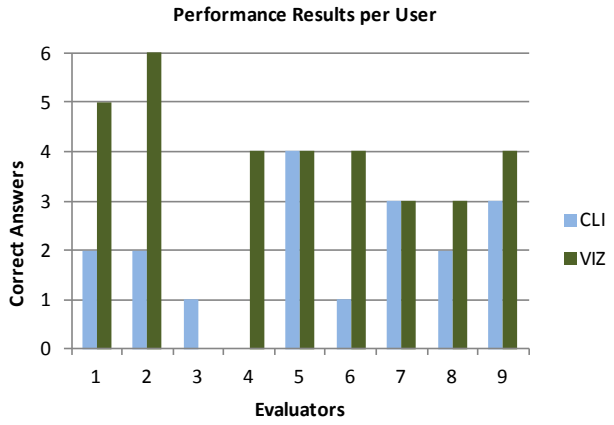

Figure 5. Performance results per user for both test conditions. Six questions were given as yes/no answers, while three others were open-ended in nature.

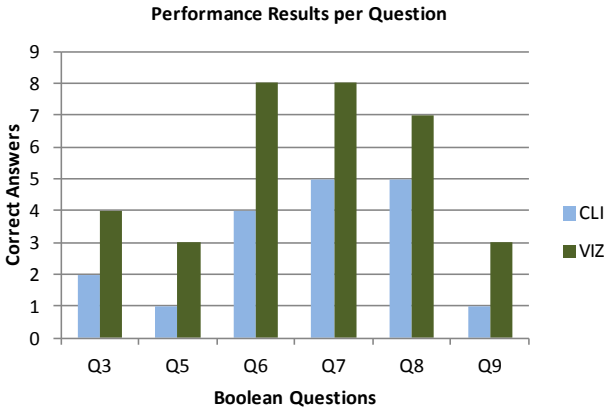

Figure 6. Performance results for each boolean-coded question over both test conditions. All showed an improvement in correct responses for the visualization (VIZ) when compared to the command line interface (CLI).

C. Measuring workload

A major goal of Pianola, and one of the stated benefits of InfoVis itself, was to attempt to reduce operator cognitive load when dealing with complex information. In order to measure workload, the subject was asked to subjectively assess their experience at the end of each treatment. The NASA Task-Load Index (TLX) [29; 30], a well-established method for measuring user workload, was used for this purpose. The index asks the user to subjectively grade each of: mental demand, physical demand, temporal demand, performance, effort and frustration. Subjects mark a score between 1 (very low) and 28 (very high) for each criterion.

An overall workload score was calculated, shown in Fig. 7, by combining scores from all of the TLX subcategories for each candidate [29]. The "Physical workload subcategory was not included in the results, since subjects registered very low in this category on both treatments. The overall score showed that the visualization reduced workload by a mean of 30% across the sample: VIZ mean=62.2, SD=27.7; CLI

mean=89.0, SD=25.8. A paired t-test across the evaluators revealed these differences to be significant: p= 0.004, 2-tailed).

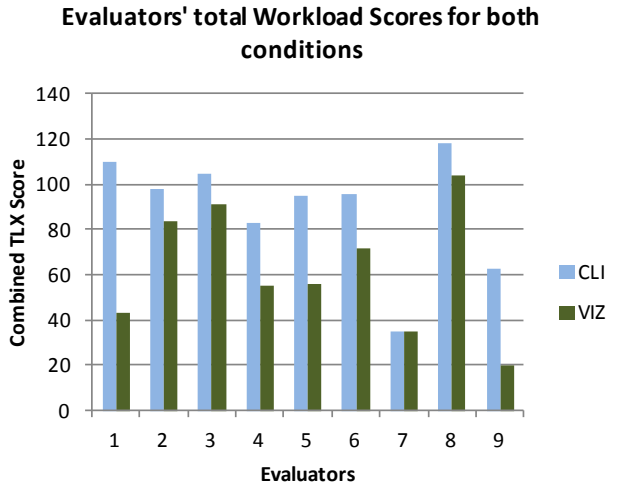

Figure 7. Evaluators' total TLX scores for both conditions. Lower values are preferable, indicating most users found the visualization to be less demanding than the command line interface.

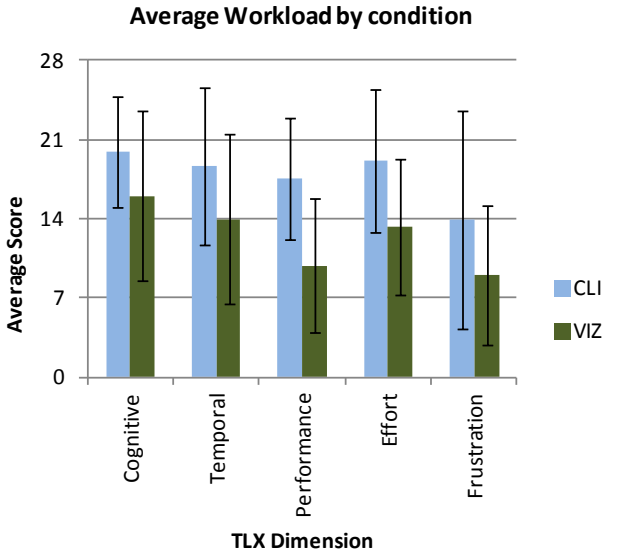

Figure 8. Average user workload per TLX dimension for each condition. Lower values are preferable, indicating the visualization was less taxing than the command-line interface for each of these dimensions.

Out of the five individual dimensions measured, the performance dimension individually was significantly better on a paired t-test (p=0.0005) (VIZ mean=9.89, SD=5.93; CLI mean=17.56, SD=5.39), whilst the Effort dimension was marginally significantly improved in the VIZ condition (p=0.049, VIZ mean = 13.33±6.00, CLI mean = 19.11±6.33). The strong performance preference for the VIZ condition in particular tallies with the objective test results. The average value and standard deviation for each dimension across all users is shown in Fig. 8.

D. *Other observations*

The test subjects tended to be comfortable with the visualization's interactivity, and quickly set about exploring the event landscape. Several people enthused about Pianola's ability to highlight subsets of the data.

As only two minutes was allotted to instruct each candidate on the use of the visualization tool before the timed test, some candidates seemed disoriented at the beginning and users typically took several minutes to assimilate all of the features of the visualization, but had gained a level of proficiency by the end of the test. This is reflected in the variance within the subcategory results, showing that there is a very short, but discernible learning curve associated with Pianola that should be taken into account in any future evaluation. The reciprocal conclusion is that the results, already in Pianola's favour, could have been even more significant if users were given more training.

Observation of users operating the visualization along with verbal feedback also indicated several improvements that could be made in terms of minor usability issues. Some users found using the time slider to be awkward, one suggested solution being that the slider should be lengthened and that the handles should be a different colour. Also, when hovering over a particularly congested part of the event matrix, the event message box would sometimes be longer than the available display, and thus the message box could be improved by being of a fixed size and scrollable. Despite these small issues, it was obvious from both the objective and subjective results that the users found the visualization to be more usable overall than the traditional command line interface for analysing the event logs.

VI. CONCLUSIONS

We have developed a novel visualization tool for monitoring and analysis of IDS event data. Pianola uses a simple visual taxonomy to solve the commonly encountered issues of visual occlusion and to improve the capability of the operator to preattentively process security event information.

Evaluation of Pianola has shown that it facilitates improved monitoring and analysis of IDS data, while significantly reducing workload compared to the traditional CLI's that network security practitioners are wedded to. Pianola addresses the needs from the task analysis described by Komlodi which mapped well to the IDS analyst situation. It also addressed limitations found in the alternative IDS visualization tools – scalability of number of displayed events, low-level perception issues, and focused on timescales and data of day-to-day managing of an SMB's network security.

Referring back to the requirements in III.E *("(a) have low cognitive overhead, (b) can be easily integrated into their workflow, and (c) supply the analyst with actionable information"*), the TLX workload surveys showed the visualization to have lower cognitive overhead, and the performance results showed the participants correctly found more actionable information. Integration into a workflow is no more problematic with a visualization than it is with a

disparate collection of scripts. As such, organisations that lack the resources for dedicated network security staff may feel empowered to use an IDS if there is a tool like Pianola to organize and interface with the data, and dedicated IDS operators may achieve better results with less effort compared to using traditional CLI-based approaches.

Based on the evaluation there are several additional features and improvements that could be made to Pianola: The tool was designed for use in smaller networks. Using a 1600x1000 monitor in portrait orientation, it is possible to display up to 50 sources plus 50 destinations without scrolling. However, it would be possible to either compress the grid, use focus+context effects [31] or increase the monitor size to display double the number of destinations. Even so, the limitation would be apparent if a network sweep scan were to trigger an alert for every host on a Class C network. A consideration could be made for events such as this: perhaps an "all hosts" destination timeline could be incorporated into the event map, allowing such "global" events to be displayed efficiently.

There is no reason why host-based intrusion detection system (HIDS) event data could not also be overlaid onto the event map using a different glyph type. This would equip the operator with deeper visual analysis capabilities.

Currently the destination hosts are ordered simply by the number of alerts that each has received. A large number of low priority alerts may put a host to the top of the event chart, whereas a host with a single, high priority alert may go to the bottom. This could be improved by introducing a weighted scoring system for different priority alerts. A future version of the tool would incorporate an algorithm to manage this.

Reporting and response features are not implemented in this version of Pianola, but it would be useful if the operator could annotate the event map with markers and notes. Also useful would be the ability to manually draw links between events while building attack scenarios.

ACKNOWLEDGMENT

We would like to thank the evaluators for taking part in the user experiment and to Richard MacFarlane for helping find those suitable ID-savvy participants in the first instance.

REFERENCES

[1] Symantec Corp. "Internet security threat report - 2011 trends", White Paper, Symantec Corp, April, 2012.

[2] National Security Agency. "Defense in depth: A practical strategy for achieving information assurance in today's highly networked environments". Retrieved 14 June, 2012, from http://www.nsa.gov/ia/_files/support/defenseindepth.pdf

[3] M. Roesch. "Snort - lightweight intrusion detection for networks," Proc. 13th USENIX conference on System administration, USENIX Association 1999, pp.229-238.

[4] G. Conti. Security data visualization: Graphical techniques for network analysis. San Francisco, California, USA: No Starch Press, 2007.

[5] G. Conti, K. Abdullah, J. Grizzard, J. Stasko, J.A. Copeland, M. Ahamad, et al. "Countering security analyst and network administrator overload through alert and packet visualization," IEEE Computer Graphics & Applications, vol. 26(2), March 2006, pp. 60-70, doi:10.1109/MCG.2006.30.

[6] M.R. Endsley. "Theoretical underpinnings of situation awareness: A critical review," in Situation awareness: Analysis and measurement, M. R. Endsley and D. J. Garland Eds. Mahwah, NJ, USA: Lawrence Erlbaum Associates Inc, 2000, pp.3-28.

[7] S.K. Card, J.D. Mackinlay and B. Shneiderman. "Information visualization," in Readings in information visualization: Using vision to think, S. K. Card, J. D. Mackinlay and B. Shneiderman Eds. San Francisco: Morgan Kaufmann, 1999, pp.1-34.

[8] W. Muller and H. Schumann. "Visualization methods for time-dependent data - an overview," Proc. 35th Winter Simulation Conference: Driving Innovation, ACM Press, 7-10 December 2003, pp.737-745, doi:10.1109/WSC.2003.1261490.

[9] A. Komlodi, J.R. Goodall and W.G. Lutters. "An information visualization framework for intrusion detection," Proc. ACM Conference on Human Factors in Computing Systems - extended abstracts, ACM Press, 24-29 April 2004, pp.1743-1746, doi:10.1145/985921.1062935.

[10] R.S. Thompson, E.M. Rantanen and W. Yurcik. "Network intrusion detection cognitive task analysis: Textual and visual tool usage and recommendations," Proc. Human Factors and Ergonomics Society 50th Annual Meeting, SAGE Publications, 16-20 October 2006, pp.669-673, doi:10.1177/154193120605000511.

[11] Y. Livnat, J. Agutter, S. Moon, R.F. Erbacher and S. Foresti. "A visualization paradigm for network intrusion detection," Proc. 6th IEEE Information Assurance Workshop, 15-17 June 2005, pp.92-99, doi:10.1109/IAW.2005.1495939.

[12] W. Aigner, S. Miksch, H. Schumann and C. Tominski. Visualization of time-oriented data. London: Springer-Verlag, 2011.

[13] N.W. Kim, S.K. Card and J. Heer. "Tracing genealogical data with timenets," Proc. Advanced Visual Interfaces, ACM Press, 26-28 May 2010, pp.241-248, doi:10.1145/1842993.1843035.

[14] D. Luo, J. Yang, M. Krstajic, W. Ribarsky and D. Keim. "Eventriver: An event-based visual analytics approach to exploring large text collections with a temporal focus," IEEE Transactions on Visualization and Computer Graphics, vol. 18(1) 2012, pp. 93-105, doi:10.1109/TVCG.2010.225.

[15] H. Shiravi, A. Shiravi and A. Ghorbani. "A survey of visualization systems for network security," IEEE Transactions on Visualization and Computer Graphics, vol. 18(8) 2012, pp. 1313-1329, doi:10.1109/TVCG.2011.144.

[16] H. Koike and K. Ohno. "Snortview: Visualization system of snort logs," Proc. 2004 ACM workshop on Visualization and data mining for computer security, ACM Press, 25-29 October 2004, pp.143-147, doi:10.1145/1029208.1029232.

[17] K. Abdullah, C. Lee, G. Conti, J.A. Copeland and J. Stasko. "Ids rainstorm: Visualizing ids alarms," Proc. IEEE Workshop on Visualization for Computer Security, IEEE Computer Society, 26 October 2005, doi:10.1109/vizsec.2005.8.

[18] D. Keim. "Designing pixel-oriented visualization techniques: Theory and applications," IEEE Transactions on Visualization and Computer Graphics, vol. 6(1), January-March 2000, pp. 59-78, doi:10.1109/2945.841121.

[19] Y. Livnat, J. Agutter, M. Shaun and S. Foresti. "Visual correlation for situational awareness," Proc. IEEE Symposium on Information Visualization, IEEE Computer Society Press, 23-25 October 2005, pp.95-102, doi:10.1109/infvis.2005.1532134.

[20] E. Bertini, P. Hertzog and D. Lalanne. "Spiralview: Towards security policies assessment through visual correlation of network resources with evolution of alarms," Proc. IEEE Symposium on Visual Analytics Science and Technology, IEEE Computer Society Press, 30 October - 1 November 2007, pp.139-146, doi:10.1109/VAST.2007.4389007.

[21] J.R. Goodall, W.G. Lutters and A. Komlodi. "The work of intrusion detection: Rethinking the role of security analysts," Proc. Americas Conference on Information Systems, AIS Press, 6-8 August 2004, pp.1421-1427.

[22] J.R. Goodall. "An evaluation of visual and textual network analysis tools," Information Visualization, vol. 10(2) 2011, pp. 145-157, doi:10.1057/ivs.2011.2.

[23] M.R. Endsley, B. Bolté and D.G. Jones. Designing for situtation awareness : An approach to user-centered design. London ; New York: Taylor & Francis, 2003.

[24] B. Shneiderman. "The eyes have it: A task by data type taxonomy for information visualizations," Proc. IEEE Visual Languages Symposium, IEEE Computer Society Press, 3-6 September 1996, pp.336-343, doi:10.1109/VL.1996.545307.

[25] A.U. Frank. "Different types of 'times' in gis," in Spatial and temporal reasoning in geographic information systems, M. J. Egenhofer and R. G. Golledge Eds. New York, NY, USA: Oxford University Press, 1998, pp.40-62.

[26] C. Reas and B. Fry. Processing : A programming handbook for visual designers and artists, 1st ed. Cambridge: MIT Press, 2007.

[27] M.C. Chuah, S.F. Roth, J. Mattis and J. Kolojejchick. "Sdm: Selective dynamic manipulation of visualizations," Proc. 8th Annual ACM Symposium on User Interface and Software Technology, ACM Press, 14-17 November 1995, pp.61-70, doi:10.1145/215585.215654.

[28] D. Holten. "Hierarchical edge bundles: Visualization of adjacency relations in hierarchical data," IEEE Transactions on Visualization and Computer Graphics, vol. 12(5), Sept/Oct 2006, pp. 741-748, doi:10.1109/TVCG.2006.147.

[29] S.G. Hart and L.E. Staveland. "Development of nasa-tlx (task load index): Results of empirical and theoretical research," in Human mental workload, A. Hancock and N. Meshati Eds. Amsterdam, Holland: North Holland Press, 1988, pp.139-184.

[30] S.G. Hart. "Nasa-task load index (nasa-tlx); 20 years later," Proc. Human Factors and Ergonomics Society 50th Annual Meeting, SAGE Publications, 16-20 October 2006, pp.904-908, doi:10.1177/154193120605000909.

[31] M. Krstajić, E. Bertini and D. Keim. "Cloudlines: Compact display of event episodes in multiple time-series," IEEE Transactions on Visualization and Computer Graphics, vol. 17(12), Nov/Dec 2011, pp. 2432-2439, doi:10.1109/TVCG.2011.179.