

A Framework for Effective Alert Visualization

Uday Banerjee

Jon Ramsey

SecureWorks

11 Executive Park Dr

Atlanta, GA 30329

{ubanerjee, jramsey}@secureworks.com

Abstract

Any organization/department that provides security typically deals with a large volume of alerts and logs generated from a variety of sources. These could be from firewalls, intrusion detection/prevention devices and agents, vulnerability scanners, etc. It would seem like a good idea to apply as much correlation as possible to this data in order to be able to see things from a bird's eye perspective. Even at this point, a human could use some additional help in deciphering the situation. The authors believe that visualization is a key component to this end. This paper describes general methods and principles that allow the use of visualization as an efficient tool for alert analysis.

Keywords: visualization, alerts, intrusion detection, correlation

Introduction

While correlation can take raw alert data analysis to a much higher level, it is important to ensure that the correlated data be presented in the most lucid manner possible. While text-based reports are a start, data could be presented visually for a more effective approach. There has been previous research that has focused on visualizing alert data but as Komlodi, et al. [1] point out, many of these studies did not take into account the specific needs that may be unique to an organization. There is no "one size fits all" visualization that will suit every organization's purposes, but there are some basic features that any visualization system must have in order to be effective. Although we do discuss one, the focus of this paper is not another network/alert visualization tool, but in fact, an attempt to create a framework of requirements that will aid not only in tool design, but visualization design as well. The paper is organized as follows: Section 1 talks about related work in the field of visualization to aid alert analysis and network flows. Section 2 details some fundamental requirements and considerations that must be incorporated into the design of visualizations and related tools. Section 3 discusses a visualization tool used within our organization to aid in alert and anomaly analysis – while highlighting its place within the framework of requirements. Section 4 discusses a sample visualization, and how its design allows for intuitive analysis. Finally, we present our conclusions.

Section 1: Related Work

There has been a significant amount of research conducted on the use of visualization in incident analysis. There are several tools currently being used to visualize alerts and

network flows. NVisionIP [2], VisFlowConnect [3], VISUAL [4], SnortView [5] are just a few of them. NVisionIP and VisFlowConnect are part of NCSA's Security Incident Fusion Tool (SIFT) [6]. NVisionIP is designed to represent network traffic on a class B network. It allows for multiple levels of granularity and has many desirable features for visualization. VisFlowConnect is another component of SIFT that most notably uses a parallel axes view to visualize network flows. There have been studies on using visualizations to detect and respond to internet-wide phenomena [7] like worms as well as tools (VISUAL) that support home-centric visualizations for security administration [4]. SnortView is another tool designed especially for visualizing snort [8] logs. All these tools conform to many of the basic requirements that make up a good visualization tool. Except in a few papers like Komlodi, et al., [1] there has not been a thorough treatment on the design requirements for visualization tools and visualizations themselves. In the following section, we will discuss these details.

Section 2: What makes a good visualization?

The answer to this question depends on what kind of organizational needs one has. Entities interested in monitoring their own network(s) may require a single network or home-centric set of visualizations. Entities that provide managed services would be interested in a system that allows for correlated visualization of data (alerts, netflows) from a variety of platforms across multiple networks. Irrespective of what type of visualization is required, in our opinion, the following are desirable features of any visualization tool:

Data driven display: This allows for effective visuals, for example, coloring all events by timestamp. This gives the operator a sense of event progression across all the events.

Multiple views: The operator should have the ability to look at the data set through multiple 'lenses' or views. This would afford the operator an opportunity to literally, see something from a different viewpoint. This may lead to an observation not easily apparent in another view.

Data linkage across multiple views: Data selected in one view should remain selected when the user switches to another view. This is a fundamental requirement, without which, it would be cumbersome to perform the same selections repeatedly when an operator switches views.

Customizable views: In many cases, it will be seen that the 'out of the box' views will not suffice for thorough analysis. A good visualization tool should allow operators to create their own custom views in order to achieve better results.

Drill-down/Zoom-out capability: The visualization tool should allow the operator to reach the detail of the raw alert data itself, if required. This allows the operator great flexibility while responding to incidents while presenting all the necessary data in one unified system. Similarly, once the operator has drilled down, the system should allow for zooming back out.

Data suppression: The tool should allow for efficient data suppression, for example, “Suppress all alerts from this IP address”, or in general “Suppress the selected data”. The corollary, “Suppress everything BUT the selected data” should also be an option. Such options would allow the operator to quickly eliminate extraneous data from the visualization.

Animation: This feature allows the operator to replay scenarios, for example, by using “animate by timestamp”, and can prove to be quite useful in some cases.

Volume-based representation: If the visualization tool uses a solid circle to represent an IP address, the tool should have the ability to increase the size (area) of the circle proportional to the number of alerts it is involved in. This visual representation of a larger area is a valuable aid to the operator’s cognitive capabilities.

Provide statistics on selected events: The tool should be able to provide some basic statistics on the selected data, like count, mean, median, unique data points, etc. This supplemental information is extremely useful in situations where it is not easy to determine such information from the visuals. For example, when one wants to find out how many times a particular IDS signature is tripping in order to assess the threat at a particular location, a glance at the visualization will not yield a quantitative answer – but a statistical view will.

Speed of rendering: While this may depend heavily on the host processor and video card, this is an important feature to have. A slow system most often results in a loss of productivity.

Interoperability with other systems (reporting, ticketing): This may or may not be a requirement for most people, but it would be nice to be able to work incidents from within the visualization system itself. From a reporting perspective, being able to use the images generated by the visualization system while preparing a report for illustrating how your preventative measures staved off the latest internet worm, would be nice.

Real-time capability: This is a very important feature to have. Most of the visualization tools that exist today are not truly real-time. Having the option to visualize data in real-time would give the operator the ability to react in real-time.

Stability under load: The visualization system should be stable when rendering extremely large quantities of data. Both the underlying hardware and software should be capable of supporting heavy loads.

Platform independent: It would be particularly useful to be able to use the visualization tool across multiple platforms. Ideally, the system would also be available for all major operating systems, and would also have a web interface for remote access.

Operator based considerations: It is also very important to bear in mind some other things such as providing the operator with capable hardware and ample display area to be able to work with. The operator's vision and ability to discern colors may need to be evaluated if working with visualizations is a crucial part of his or her job.

Many of the features discussed above apply to both the tool and the visualization itself. But the layout and interface visuals are of utmost importance as well. In order to make cognition an easy task for the operator, the screens should be logically designed and well laid out. A good layout should make the anomalies stand out – like top offending IP, most attacked host/network, most tripped signature, etc.

Data considerations:

The authors' experience has been that better visualizations result from richer data sets. More correlation can be performed if there are more data fields to work with. More data fields allow the creation of more views. The data to be visualized may differ between organizations. Our organization deals with security alerts, and figure 1 shows an example set of data fields that could be fed into a security alert visualization system. Since the raw alert data may not be always suitable for certain types of visualizations, it may be a good idea to store the alert data in an exclusive database that feeds the visualization system. This will give us the option to manipulate data if needed, while not impacting production systems. One example would be the ability to separately query the ASN for an IP address for correlation purposes and store it in the visualization database.

Figure 1 shows an example of the alert data fields and supplemental data fields that could be stored in the visualization database for a managed security provider scenario. The fields marked with an asterisk indicate unique identifiers across all events. This set of data could be visualized in multiple ways because of the number of fields we are feeding the system. The nature of the supplemental data, such as client information, associated vulnerabilities, etc., gives the operator access to pertinent information that he or she can use to perform activities like calling a client, changing signatures, etc. This ability makes the visualization tool a centralized place where a security analyst can work. This relates to a point made earlier stating that it would be desirable for the visualization system to tie in with the ticketing and reporting systems.

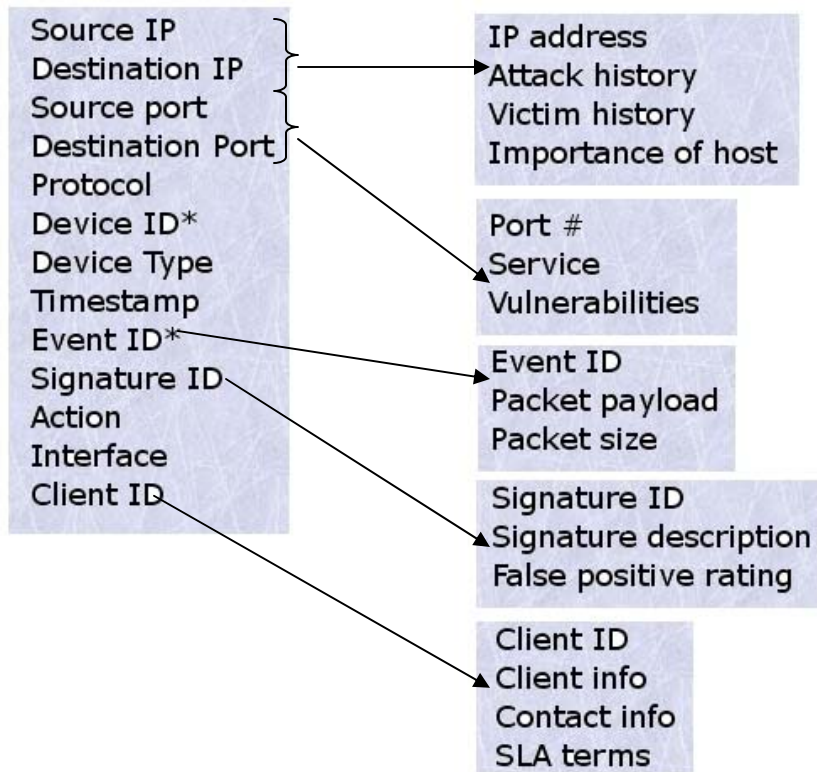


Figure 1: Example data that could be fed to an alert visualization system

Section 3: Advanced Analytics

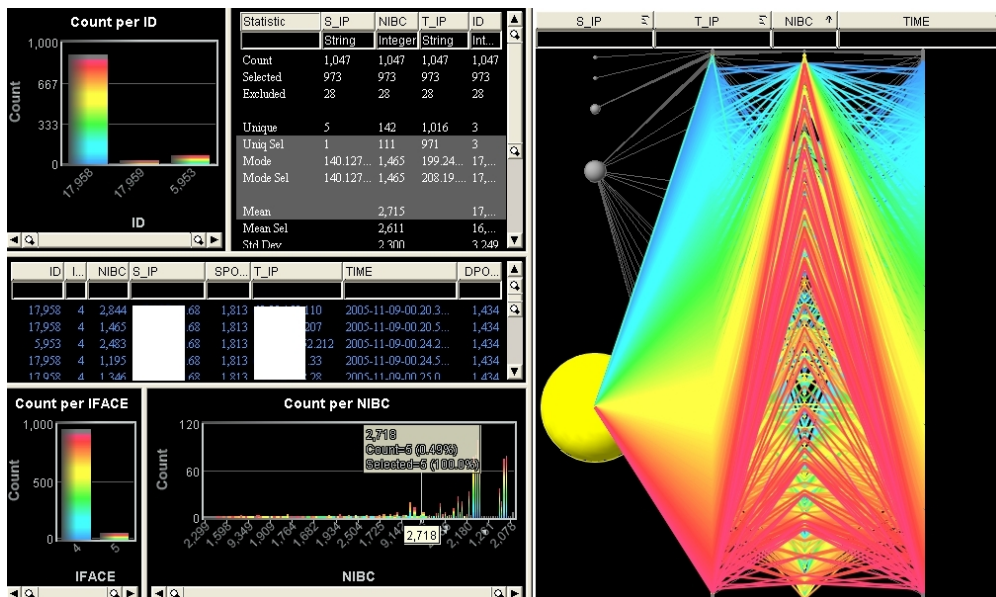


Figure 2: Layout of the visualization tool used within our organization.

Figure 2 shows one of the views displayed in the tool called Advanced Analytics [9]. This is a tool we use within our organization to perform visualizations with alert data.

Please note that sensitive information has been removed. Advanced Analytics essentially allows the user to render a variety of charts and graphs based on input data. The strength of this tool lies in its easily customizable interface and powerful data visualization engine. The layout above is divided into several panes, which detail signature ID counts, statistics on selected events, a raw datasheet, interface counts, device counts, and a parallel axes view showing source IP, destination IP, device ID and timestamps. The entire display is color-coded by event timestamp. This explains the rainbow of colors on the time axis in the parallel axes view. The source IP with the most traffic is rendered as a larger sized solid circle than the others. This lets us know right away that this is the top offender. The color-coding is present in all the panels, so it is easy to track the progression of details like when each device was attacked.

The statistics pane is extremely useful when quantitative analysis needs to be done. It presents details like means, counts, unique counts, standard deviations, etc. The tool supports features like drill down, animation, multiple customizable views, data linkage across multiple views, and data suppression. The tool is relatively quick at rendering visuals, and is quite stable under heavy loads. The system interacts with our ticketing system in a limited fashion, but is not near the level we would like. The version that we use today does not support real-time data feeds, but we were informed that support for this feature was forthcoming. Presently, we are using a client software that needs to be installed on the local machine, but versions of this product that support a web browser interface are available. Wood [10] also conducted a study using this product in an attempt to visualize IDS data but focused on select visualizations.

Section 4: Visualization scenario

A few visuals from Advanced Analytics are presented that allow us to investigate a simple spike in activity. As shown in figure 3, some prior correlation alerts a security analyst that the alert level for a particular IP address has crossed a preset threshold. The analyst then pulls up advanced analytics, which fires off a query equivalent to “show me the activity for this IP address over the last 24 hours”.

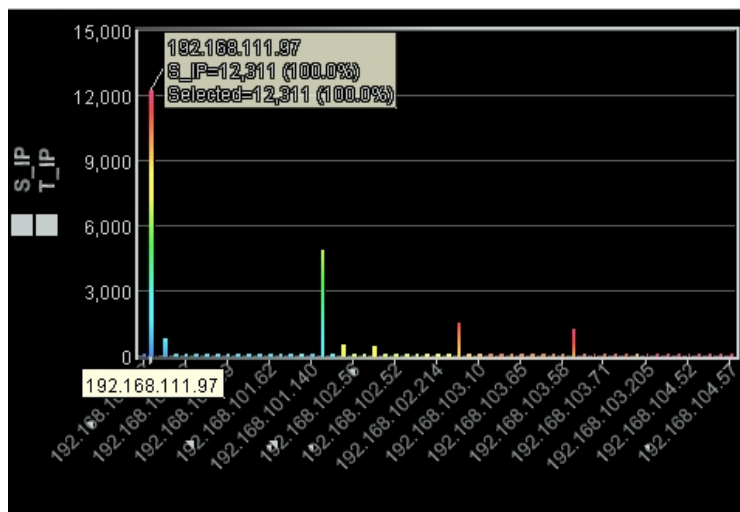


Figure 3: Activity per IP address

This query is customizable, but this is the default. The resulting data set is visualized as shown in figure 4.

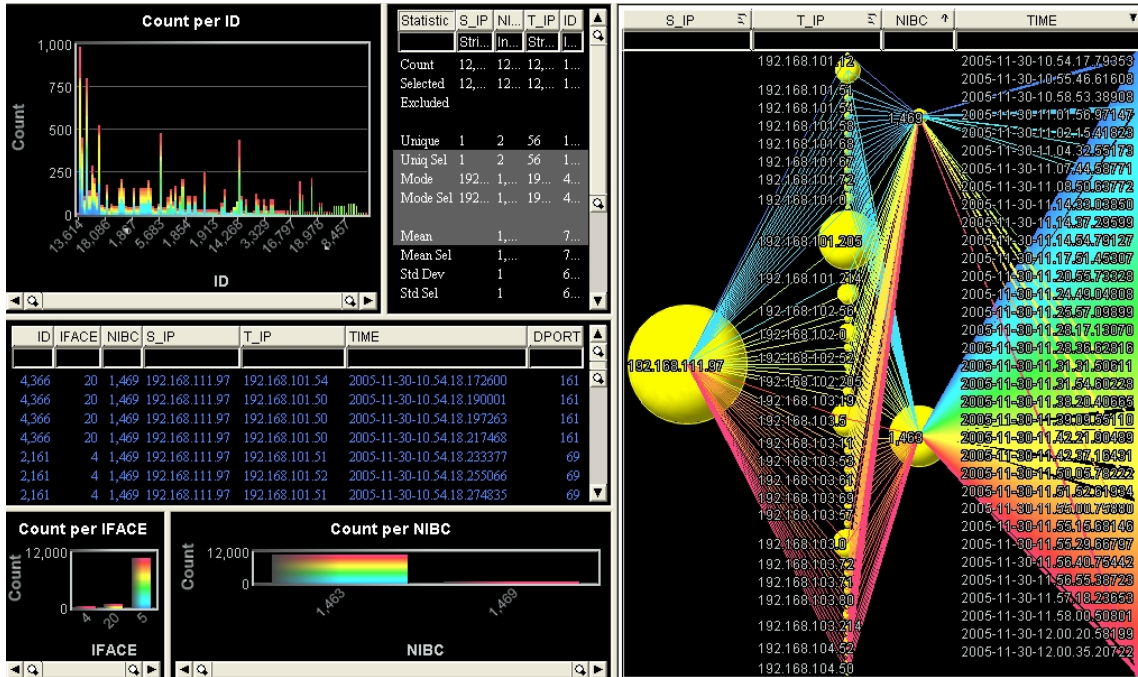


Figure 4: Network scans visualized

The images above clearly show the offending IP address performing the scan. The parallel axes view tells us which IP addresses were attacked more than others. The “Count per ID” pane in the top left gives us an idea of what signature Ids were tripped during this scan and their counts. The datasheet gives us a textual view of the alerts selected. The fact that time is embedded into the visual is quite helpful to understand the progression of attacks. It is important to keep in mind that building a set of visualizations requires good design, so as to be able to help the operator’s cognitive and intuitive capacities.

The incident shown above is a simple scenario. This tool has been used to perform extensive investigations into very large amounts of data in an efficient manner. Supplemental data, as shown in figure 1 have not yet been incorporated into our visuals, but can easily be done. This promises to make this tool a centralized console for a security analyst to work incidents.

Conclusion

This paper attempted to list a feature set that every visualization tool should have, while talking briefly about the importance of design of the visualization itself. The tool presented above is a step in the right direction. While it possesses many of the desirable features of an alert visualization tool, it still has some room for improvement. It fares well in comparison with the other visualization tools available today, but it still has a way to go in order to meet our recommended feature set. Future work should involve working

toward unifying visualization systems with other systems. The types of visualizations available from most tools today are simplistic two-dimensional figures. More research needs to be done to see if data can be represented more effectively using different models, possibly using three dimensions.

References:

- [1] Komlodi, A; Goodall, J.R.; Lutters, J.G. An Information Visualization Framework for Intrusion Detection, CHI '04 extended abstracts on Human factors in computing systems, April 2004.
- [2] Lakkaraju, K; Yurcik, W; Lee, A.J. NVisionIP: netflow visualizations of system state for security situational awareness, Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security, October 2004
- [3] Yin, X; Yurcik, W; Treaster, M; Li, Y; Lakkaraju, K. VisFlowConnect: netflow visualizations of link relationships for security situational awareness, Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security, October 2004.
- [4] Bali, R; Fink, G.A.; North, C. Home-Centric Visualization of Network Traffic for Security Administration, Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security, October 2004.
- [5] Koike, H; Ohno, K. SnortView: Visualization System of Snort Logs, Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security, October 2004.
- [6] Security Incident Fusion Tools (SIFT).
<http://www.ncassr.org/projects/sift/>.
- [7] Valdes, A; Fong, M. Scalable Visualization of Propagating Internet Phenomena, Proceedings of the 2004 ACM workshop on Visualization and data mining for computer security, October 2004.
- [8] Snort – the de facto standard for intrusion detection/prevention.
<http://www.snort.org/>.
- [9] Advanced Analytics.
<http://www.visualinsights.com>
- [10] Wood, A. Intrusion Detection: Visualizing attacks in IDS data, SANS GIAC Practical, 2003.