IDS Alert Visualization and Monitoring through Heuristic Host Selection

Hadi Shiravi*, Ali Shiravi*, and Ali A. Ghorbani

Information Security Centre of Excellence
University of New Brunswick, Canada
{hadi.shiravi,ali.shiravi,ghorbani}@unb.ca

Abstract. Traversing through multiple pages of log entries, trying to detect malicious and anomalous behavior and being able to correlate events to address multiple use cases is a non trivial task for a security administrator. It requires resources, expert knowledge and time. In this paper, we present a novel security visualization system entitled Avisa. It accentuates fundamental matters of information visualization, namely interaction and animation and synthesizes it with intrusion detection audit traces. Visual constraints inspired the use of heuristic metrics to select and display hosts with irregular and variant behaviors. We thoroughly describe the ideas behind the heuristic metrics and perform an empirical analysis to individually evaluate each metric's functionality. Avisa's intuitive interface, accompanied by the power of the heuristic functions, allows the perception of patterns and emergent properties, facilitating in understanding the underlying data.

Keywords: Visualization, IDS Alerts, Animation, Interaction, Beta-Splines, Heuristic Function, Exponential Moving Average.

1 Introduction

Defending networks against potential attacks and intrusions is a delicate act. It demands extensive resources and adequate knowledge. Resources that can easily exhaust the security budget of organizations and knowledge that can mostly be obtained through the use of well trained administrators. Intrusion Detection Systems (IDSs), firewalls and Intrusion Prevention Systems (IPSs) are complementary security devices that are widely deployed and assist analysts in securing an enterprise's network. Although such devices play an undeniable role in securing a network, there is still no silver bullet in protecting a network against potential security breaches. This is all due to the limitations that are inherent in current security systems.

In multi layered security architectures, IDSs are deployed perpendicular to the communication link, as a second line of defense behind firewalls. In anomaly detection systems a normal activity model of the network is built using a statistical or machine learning approach. Any behavior deviating from this normal

* Hadi Shiravi and Ali Shiravi contributed equally to this work.

M. Soriano, S. Qing, and J. López (Eds.): ICICS 2010, LNCS 6476, pp. 445–458, 2010.

model is considered suspicious and a potential alarm is raised. In misuse detection, signature-based IDSs constitute a collection of pre-configured and predetermined attack patterns known as signatures. Whenever a signature matches the network traffic, an alert is generated. A major drawback of IDSs, regardless of their detection mechanism, is the overwhelming number of alerts generated on a daily basis that can easily exhaust security administrators [1,2,3]. Additionally, since signatures are not always written precisely enough or are written too specific and because deviation from normal behavior does not always correspond to malicious behavior, the phenomenon of false positives and false negatives also arises [3,4,5].

All aforementioned limitations has lead researches in the IDS community to not only develop better detection algorithms and signature tuning mechanisms, but to also focus on discovering various relations between individual alerts, formally known as alert correlation. The approach applied in this paper takes on a very different perspective. It builds upon the fundamental basics of information visualization. Visualization aims at taking advantage of the perceptual and cognitive powers of the human being through a fundamental and flexible pattern finder, the human visual system [6]. Visualization allows for inherent attributes of a dataset, not previously anticipated, to become apparent. The power of visualization through the human visual system allows for perception of patterns that can lead to new insights of the underlying data. It is this fascinating ability that provokes a user to pose new questions and to explore and discover unseen dimensions of data. Security Visualization is a very young term [7]. It expresses on the idea that common visualization techniques have been designed for use cases that are not supportive of security related data, demanding novel techniques fine tuned for the purpose of thorough analysis.

Visualization helps to comprehend and analyze large amounts of data, a fundamental necessity for network security due to the large volume of audit traces produced each day. Visualization allows for inherent attributes of a dataset, not previously anticipated, to become apparent. If such properties were known upfront, it would be possible to detect these incidents without visualization. However, they need to be discovered first, and visual tools are best suited to do so. The power of visualization through the human visual system allows for perception of patterns that can lead to new insights of the underlying data. It is this fascinating ability that provokes a user to pose new questions and to explore and discover unseen dimensions of data. Visualization is not only efficient but also very effective at communicating information. A single graph or picture can potentially summarize a month's worth of IDS alerts, possibly showing trends and exceptions, as oppose to scrolling through multiple pages of raw audit data with little sense of the underlying events.

Security Visualization is a very young term [7]. It expresses on the idea that common visualization techniques have been designed for use cases that are not supportive of security related data, demanding novel techniques fine tuned for the purpose of thorough analysis. Security visualizations should have an elegant and visually appealing design. They should be built upon the principles of information visualization. A factor that is often overlooked. Since security analysts, as the main developers of these tools, are not necessarily visual designers. At the same time, security visualizations should be informative, interactive, and have exploratory capabilities. It is these features that assist an analyst to first grasp an overall view of the data, allowing her to perceive areas of activity, and second to permit further explorations of irregular behavioral patterns, assisting in the detection of potential intrusions or identification of possible outliers.

In this paper, we present a novel security visualization system entitled Avisa. It accentuates fundamental matters of information visualization, namely interaction and animation and synthesizes it with IDS audit traces. The system utilizes three categories of heuristic functions, each composed of multiple heuristic measures, to collectively identify hosts of peculiar behavior.

In this paper, we make the following contributions;

- Design and implementation of a novel security visualization system for displaying a selective number of hosts and their corresponding alerts in an interactive manner.
- Utilization of heuristic functions to combat visual constraints by identifying hosts with irregular and anomalous activities. The heuristic functions can nonetheless be applied independently of the visual system itself.

The remainder of this paper is organized as follows. In Sect. 2, we present our system and discuss its design and architectural features. In Sect. 3, we detail the technicalities behind the heuristic functions. A thorough analysis of the heuristic functions is performed in Sect. 4. In Sect. 5, the functionality of the system is evaluated using several use-case scenarios of attacks. Section 6 looks at background and related work and we conclude our paper in Sect. 7 with details on future improvements of the system.

2 Avisa: A Network Security Visualization System

The high number of alerts generated from modern IDSs greatly reduces the capability of an analyst to correlate events. This combined with a high percentage of false positives effectively transforms log analysis into a tedious task. Avisa is a security visualization system that addresses the aforementioned problems. It is built upon an emerging information visualization paradigm, namely radial visualization. The paradigm is aesthetically pleasing, allows for data to be encoded on both the outer and interior parts of the ring, and has a compact layout for an effortless user interaction [8].

Figure 1 illustrates our systems initial design. It is composed of two main components, the radial panel and the interior arcs. The radial panel itself is composed of two inner and outer rings. Starting from the top left corner, a color band inside the inner ring, formally known as the *alert type panel*, is used to display IDS alert types. The outer ring located exactly above this color band is used for categorizing alert types and facilitating user interaction. One color is assigned to each alert type category and different shades of the same color are used for individual alert types inside the category. We believe that this color coding eases visual correlation. The greater portion of the radial panel is devoted to internal hosts residing inside a network. Hosts can be arranged in subnets or asset groups or even be manually arranged based on specific machines that an admin is interested in monitoring. The outer ring surrounding the individual hosts, formally known as the *subnet panel*, depicts these arrangements.

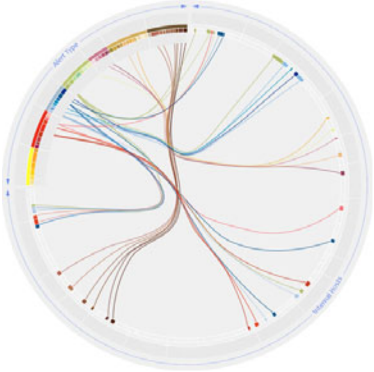


Fig. 1. Initial design of Avisa

The inner arcs residing inside the radial panel illustrate actual alerts. Alerts triggered by an IDS on ongoing traffic are stored and depending on various parameters, arcs are drawn starting from the alert type panel on the top left and ending at the host affiliated with the alert on the host panel. Exceptional care has been taken to avoid occlusion in the drawing of the arcs. As can be seen in Fig. 1, all arcs emerging from an alert type are laid out in a fashion that only a single exit point is illustrated.

Avisa also supports filtering through direct interaction with the user. By simply clicking on any of the hosts, subnets, alert types or alert categories the entire portion of the host or alert panels are devoted to them. This feature allows for filtering of hosts and alerts in any combination. If an admin, for example, would want to see the detailed activity of a particular subnet they would just need to click on the specific subnet and the entire host panel would be populated with the subnet's hosts. If a specific alert category seems rather unusual in the subnet, it can be further explored by clicking on it. These point and click features not only save an analyst's time but also allows for a smooth, thorough analysis.

An advantage of Avisa is its use of animation. Animation can facilitate the perception of change over time. In our case, animation is used not only to display transitions of one view to another, but to assist in highlighting system transitions from one state to another. Unfortunately due to space restrictions, this discussion has been omitted.

Table 1. Heuristic functions

Category	General Definition	Measure
Exponential Moving Average(EMA)	$S_t = \alpha Y_t + (1 - \alpha)\, S_{t-1}$	–No. of Alerts –No. of Distinct Sources –Non-shared Percentage of Sources Between each Update Period –No. of Distinct Alert Types –Non-shared Percentage of Distinct Alert Types Between each Update Period –No. of Alerts for each Distinct Alert Type
Standard Deviation(STD)		Alert Arrival Times
Difference Exponential Moving Average (DEMA)	$S_t = \alpha\,(Y_t - Y_{t-1}) + (1 - \alpha)\, S_{t-1}$	–No. of Alerts –No. of Alerts for each Distinct Alert Type

3 Heuristic Functions

The amount of information that can be displayed on a given canvas is subject to various constraints including canvas resolution, human perception, and visual clarity. Thus, it is ultimately necessary to limit the amount of information displayed. In this work we deal with two general categories of information; (a) hosts and (b) related alerts. While both inhibit a significant amount of noise, we have taken an approach to decrease the amount of visual clutter by both decreasing the number of hosts and also the number of alerts displayed at each interval.

A viable option to limit the number of displayed hosts is to prioritize them. This could be driven by allowing the user to specify the host priorities by asset importance as in the importance of servers over simple hosts. Apart from the burdensome task of labeling each host, in most large networks a high percentage of hosts will bear the same priority level and thus will make this option impractical. The approach taken in Avisa is to assign scores to hosts based on an collection of metrics that reflect the amount of change in a variety of aspects related to the alerts received by a certain host. Extra control is then given to the end-user to enable them to fine-tune the scoring process and to add or remove emphasis on a particular aspect. The proposed functions have been determined by observation, expert knowledge and experimentation. They are chosen to be efficient and effective to reflect change as much as required. The three categories of functions are as shown in Table 1 and are further elaborated below.

Exponential Moving Average (EMA). Several measures are monitored through the utilization of EMA. Here S_t is the current value of EMA, S_{t-1} the previous EMA, Y_t the current value of a measure, and α indicates the smoothing factor. EMA takes all past data into account by applying an exponential decay over time. Interestingly, it only requires the previous EMA value and current value to compute the new EMA.

The measures mentioned in Table 1 are required to *directly* affect the score of a host. In this respect, EMA has been employed to effectively provide the necessary means to represent change while smoothing it out exponentially over the previous samplings. Finally for each of the measures, S_t is normalized to a value between 0 and 1 over all hosts for the same measure (represented by $\overline{ema_i}$) and is subsequently used in determining a host's final score. The normalization was prevised as a means to prevent a certain measure from dominating the overall score.

Difference Exponential Moving Average(DEMA). Several of the required measure were necessary to *indirectly* affect a host's score. By indirect we intend to elaborate on the fact that instead of the value itself, its change over the previous intervals for a measure is desired. To exhibit the effect of sudden changes to these measures, we have applied an EMA to this change and have subsequently called this the DEMA of a measure. To calculate a DEMA, the value of a measure from the current and previous period is required. As in EMA, each S_t is normalized to a value between 0 and 1 (represented by $\overline{dema_i}$) and subsequently used in determining a host's final score.

Standard Deviation (STD). To provide a means to represent the dispersion of alert arrival times over a certain period of time, the standard deviation (σ_i) of a host's alert arrival times is calculated and subsequently normalized. The normalization of σ_i is done in two phases. It is initially calculated by dividing a host's σ value over it theoretically maximum standard deviation value.Considering the interim values of all host, they are subsequently normalized to a value between 0 and 1 (represented by $\overline{\sigma_i}$).

The final aggregation function is a summation of the mentioned measures as is illustrated in Equation 1, and detailed in Table 2. Here, several weights are introduced for each of the measures mentioned in Table 1, to enable the end-user to control the effectiveness of certain functions in a host's final score. These weights are represented by w_{ema_j}, w_{dema_j}, and w_{std} for EMA, DEMA and STD measures, respectively. In addition to the weights, a Gaussian function, G, is applied to the final $\overline{\sigma_i}$ value to enable the user to specify the intended amount of dispersion set for maximum score. The amount of dispersion is set through the γ parameter which takes a value between 0 and 1, where zero sets the preference to no dispersion and 1 to full dispersion.

$$\begin{aligned} Score_{Host_i} = & \sum_{j=1}^{6} w_{ema_j} \cdot \overline{ema_{ji}} \left(\alpha_j\right) + \\ & \sum_{j=1}^{2} w_{dema_j} \cdot \overline{dema_{ji}} \left(\alpha_j\right) + \\ & w_{std} \cdot G\left(\overline{\sigma_i}, \gamma\right) \end{aligned} \tag{1}$$

Table 2. Definitions of variables and functions of Equation 1

Variable	Definition
w_{ema_j}	User specified weight [0 1] for function j of category EMA
$\overline{ema_{ji}}\left(\alpha_j\right)$	Normalized EMA function j with user specified smoothing factor α_j for host i
w_{dema_j}	User specified weight [0 1] for function j of category DEMA
$\overline{dema_{ji}}\left(\alpha_j\right)$	Normalized DEMA function j with user specified smoothing factor α_j for host i
w_{std}	User specified weight [0 1] for Standard Deviation
$\overline{\sigma_i}$	Normalized Standard Deviation of Alert Arrival Times for host i
γ	Amount of dispersion ([0 1]) preferred by the user in relation to alert arrival times. Zero indicates no dispersion and 1 indicates full dispersion of alerts preferred over the time window
$G\left(x, \mu\right)$	$e^{\frac{-(x-\mu)^2}{0.18}}$

4 Evaluation of Heuristic Metrics

An empirical analysis is conducted to evaluate the influence and effect of each heuristic function on a host's final score. The analysis consists of the aggregated outcome of multiple heuristic functions which utilize a single measure. In other words, the effect of a measure, like number of alerts, is studied simultaneously for all categories to which it belongs to. In the case of the number of alerts, the EMA and DEMA heuristic functions are analyzed together as they are designed to counter-weight each other. In this respect, by assigning weights (in Equation 1) to only the heuristic functions under analysis and zero to all others, we are able to independently verify the outcome of a particular heuristic measure. In the coming section, the following measures are analyzed across multiple categories;

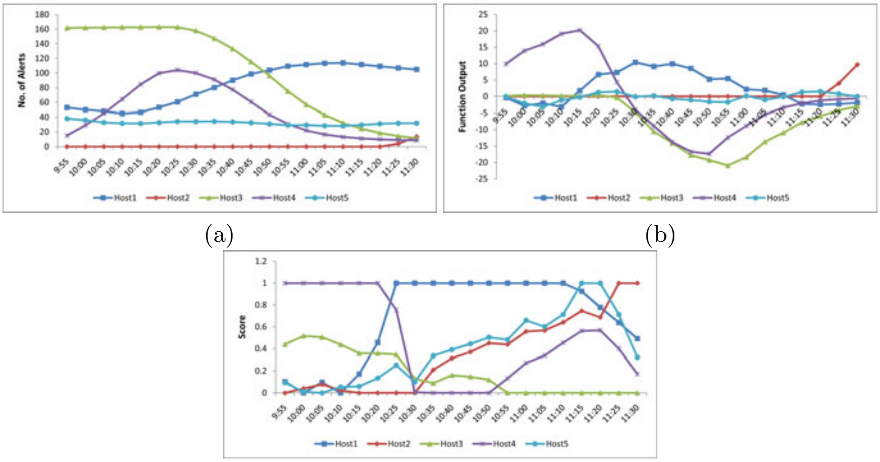
- EMA of number of alerts, DEMA of number of alerts
- EMA of number of distinct sources, EMA of non shared percentage of sources between each update period
- STD of alert arrival times

The analysis is performed on a private dataset available to our security center, composed of a collection of Snort alerts triggered on a network of over 850 nodes in a time span of 24 hours. The dataset contains over 100,000 events and 24 distinct alert types. To our knowledge, the dataset comprises benign and anomalous activities with several cases of real-world attacks. For practical reasons, five hosts have been selected, and the behaviors of the heuristic measures in regards to the overall score are analyzed. We believe that the results of these analyses demonstrate the effectiveness of the heuristic functions, as they are not limited to specific hosts and can be extended to all hosts in the dataset.

4.1 Number of Alerts

A six period EMA and DEMA representing the number of alerts for five hosts is illustrated in Figs. 2(a) and 2(b), respectively. Their final scores are also depicted in Fig. 2(c), with the number of alerts for the past 30 minutes sampled at 5 minute intervals. The EMA diagram in Fig. 2(a) clearly shows a constant rate of alerts for `host3` in the first 30 minutes of activity. This constant activity is clearly filtered in the DEMA diagram. In the same period, `host4` is experiencing an escalation in the number of alerts, which leads to the increase seen in the DEMA diagram. A similar situation regarding `host2` is seen in the final 10 minutes of activity in Fig. 2(a). While all hosts are experiencing a constant alert rate, there is a sudden small increase in the number of alerts for `host2` and this small shift is seen, in a higher magnitude, in the DEMA chart.

By further analysis we have come to the conclusion that assigning a higher weight (0.7) to the DEMA and a lower weight (0.3) to the EMA would result in the eventual filtering of hosts with a constant number of alerts, while enabling the hosts with varying activities to gain higher scores. In practice, hosts which generate a constant number of alerts are a strong sign of application, device, or IDS misconfiguration. Attacks of high concern often manifest themselves in only a small number of events, if any, leading to the necessity of identifying even the smallest of changes. The results of these assumptions are illustrated in Fig. 2(c). In the first 30 minutes of activity, since the change in the number of alerts is of higher interest than constant activity, a higher final score is assigned to `host4` despite a much lower alert count than `host3`. This is also true for `host2`, where only a small rise in the number of alerts (only 14) results in the host to be ranked first in the respective period.



(a)

(b)

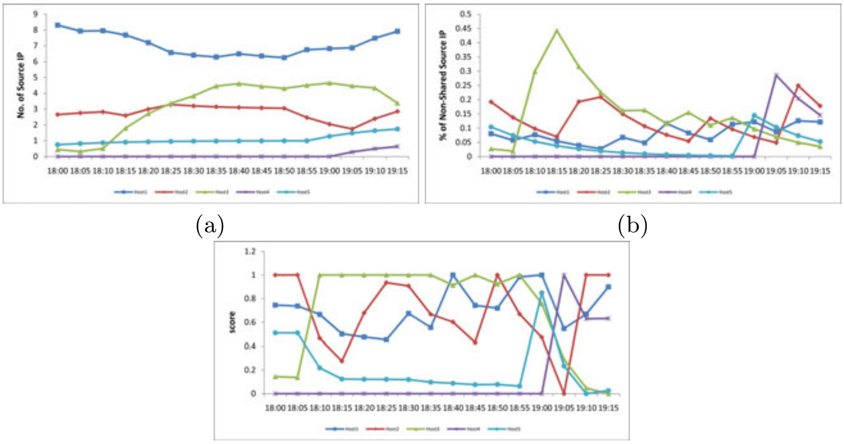
(c)

Fig. 2. (a) EMA of number of alerts. (b) DEMA of number of alerts. (c) Hosts' final score.

4.2 Number of Distinct Sources

A six period EMA representing the number of distinct source IPs and the percentage of non-shared source IPs between each update period is illustrated in Figs. 3(a) and 3(b), respectively. The final scores of the hosts are also depicted in Fig. 3(c), with the number of distinct sources for the past 30 minutes, sampled at 5 minute intervals. The EMA diagrams clearly portray an increase in the number of distinct source IPs regarding `host3`. But, as illustrated in Fig. 3(b), `host3` is not only receiving alerts from a larger range of IP addresses, the percentage of non-shared source IP addresses between the current and previous intervals has also increased. This means that in this specific time period, `host3` is receiving alerts from multiple IP addresses and that a high percentage of these IPs are changing in each interval. This situation illustrates a host with peculiar behavior, one that would make a great candidate for displaying. The respective weights of the heuristic functions must also be carefully tuned to prioritize hosts which exhibit such behavior.

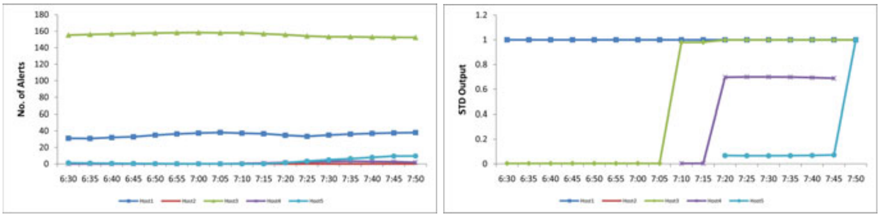
In contrast, consider the activity of `host1` in Fig. 3(a). Although the number of distinct source IP addresses in each interval is higher compared to other hosts, a lower percentage of these addresses are unique in regards to previous periods. This conveys the fact that `host1` is constantly receiving alerts from similar IP addresses, making it a candidate of lesser interest but of medium concern. Figure 3(c) illustrates the final scores with a weight of 0.3 assigned to the number of distinct IP addresses and 0.7 to the number of non-shared percentage of source IPs. As expected, in the first 35 minutes of activity, `host3` has gained the highest score due to its irregular and variant behavior while `host1` has a lower rank, expressing a medium level of concern.



(a) (b)

(c)

Fig. 3. (a) EMA of number of distinct source IPs. (b) EMA of percentage of non-shared source IPs between each update period. (c) Hosts' final score.



(a) (b)

Fig. 4. (a) EMA of number of alerts. (b) STD of alert arrival times.

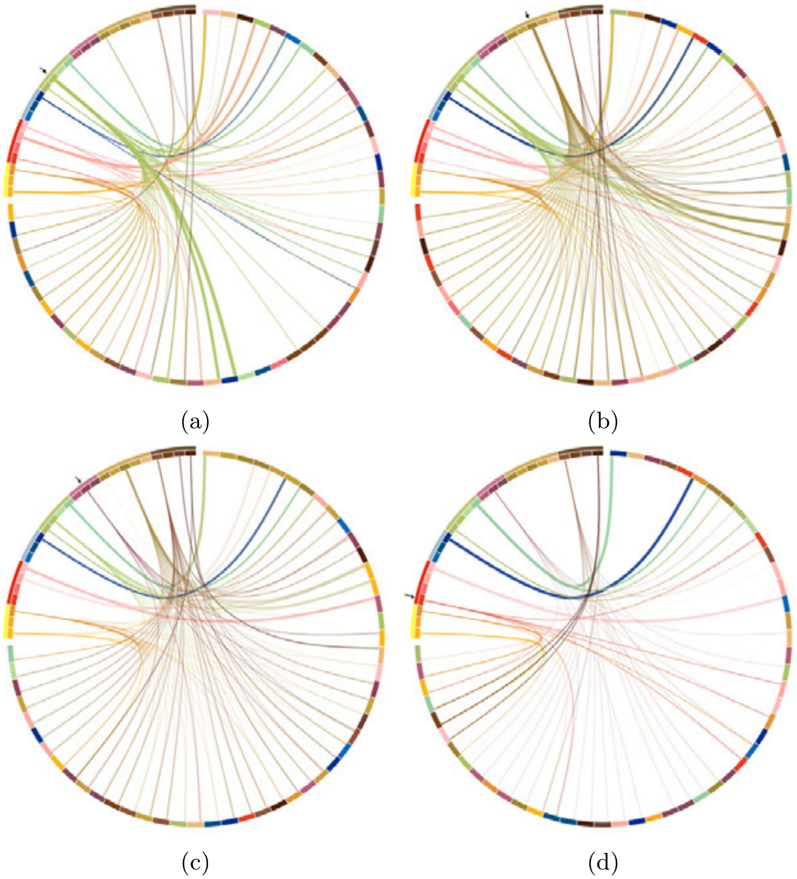
4.3 Standard Deviation of Alert Arrival Times

A six period EMA representing the number of alerts for 5 hosts is illustrated in Fig. 4(a). Figure 4(b) depicts the normalized value of the standard deviation of alert arrival times after being run through the Gaussian function presented in Table 2. A higher value in Fig. 4(b) indicates alerts arriving in a burst, depicting a significant amount of activity in a short period of time. Conversely, a lower value is indicative of disperse arrival times, expressing activity that is loosely distributed across the time window. Interestingly, both can be argued to be as important and thus we have designed the function to adequately adjust the scoring through the γ value respectively. In our particular dataset, we have observed that many of the intrusions were packed in small bursts of alerts and therefore we have assigned γ as 1 to represent such behavior.

Fig. 4(a) illustrates `host3` as it is receiving a constant number of alerts in the depicted time span. A slightly different behavior is seen in Fig. 4(b). In the first 40 minutes of activity, alert arrival times of `host3` are dispersed more evenly than other hosts, resulting in a much lower value. In contrast, in the final 45 minutes of activity, despite possessing the same number of alerts, `host3` has obtained a higher value. This is indicative of alerts arriving in a more compact manner in comparison to other hosts. The same situation is seen with hosts 4 and 5. As portrayed by Fig. 4(b), a sudden slight increase in the number of alerts is indicative of a sudden burst. This heuristic measure assures that such activity is flagged and represented by the system to the administrator.

5 Visual Correlation of Alerts

In order to demonstrate the capabilities of Avisa in visual correlation of alerts and identification of attack patterns, several use case scenarios are presented. Avisa, in its default settings is run on the dataset used for the evaluation of the heuristic functions. As Avisa runs in normal mode, alerts of all type start appearing on the screen. Some are false positives, due to misconfiguration of devices or services while others may seem more notable or suspicious, requiring further investigation. In Fig. 5(a) large number of Nmap reconnaissance events have been triggered, targeting multiple hosts on the network. Reconnaissance is



(a)

(b)

(c)

(d)

Fig. 5. (a) Nmap reconnaissance alerts are displayed in light green. (b) Failed login attempts along with previous reconnaissance alerts are displayed. (c) Buffer overflow alerts are displayed in dark purple. (d) TFTP GET passwd attempts are displayed in red.

a preliminary activity used by an attacker to gather information about a host or network. Security scanning tools such as Nmap are frequently used to identify network computers, running services, open ports and operating systems as they can lead the attacker to valuable information. In Fig. 5(b) multiple failed login attempts are seen targeted at the same range of IP addresses. Although these events are triggered at different time stamps, Avisa's time window facilitates visual correlation by displaying past and current events at the same time. The unusual number of attempts indicates that the intruder is attempting a brute force method of attack. In a real world scenario after an attacker gains enough information on its targets, having attempted a brute force attack without success, she then attempts to exploit vulnerabilities on running applications in order to acquire access to a system. The same procedure is seen in Fig. 5(c)that after a brute force attempt, which is likely to have been unsuccessful, the attacker has tried to gain administrator privileges through exploiting several FTP severs vulnerable to buffer overflow. The *bftpd chown overflow* alert is generated on multiple servers indicating a possible intrusion. This bug allows an attacker to execute unauthorized commands on the target system with a root access. Once the hacker has acquired access to the network, she would want to maintain that access. Typical actions would be to download password files so that reentering the system at a later time is possible or install keyloggers to monitor keystrokes of the victim. As can be seen in Fig. 5(d), the compromised FTP servers are receiving *TFTP GET passwd* alerts indicating a file transfer containing root level authentication information. These examples show that rather than traversing through multiple lines of audit traces and correlating alerts based on previous and current events, an administrator can easily use Avisa to interact, filter and visually correlate events.

6 Related Work

For IDS alert visualization, very limited work has been carried out. SnortSnarf [9] and ACID [10] constitute earlier work in this area as they are only simple interface layers on top of raw alerts with basic statistical analysis abilities. NIVA [11] uses haptic integration to display hosts and alerts in a 3D graphical environment. Nodes are positioned using gravitational equations, electromagnetics and fluid dynamics. Link color also represents severity of attacks. SnortView [12] uses a matrix view to display source IP addresses of alerts over time. Alerts are drawn as glyphs with different services and protocols displayed using a variety of shapes and colors. IP Matrix [13] uses a 2D matrix representation of IP space to display both the Internet and local networks. Actual alerts are color coded and drawn as pixels while histograms are used to show relative number of alerts in each address block. IDS RainStorm [14] consists of a main view which is depicted in eight columns each showing in a top to bottom matter a contiguous set of IP addresses. Alarms are also represented as color coded pixels allowing for a 2.5 class B IP address block to be represented onto a single display for a full 24 hour time period. VizAlert [15,16] is a novel paradigm for visual correlation of network alerts. The developed system displays the local network topology map in the center with the various alert logs on a surrounding ring. The ring's width also represents time and is divided into several history periods. A line is drawn from a specific attack type on the outer ring to a particular host on the topology map to represent a triggered alarm. AlertGraph [17] incorporates a very common 3D graph and assigns a combination of source/destination IP and port numbers to each axis.

All aforesaid systems have one issue in common. They tend to visualize and display every IP or host involved in a security event, resulting in an often occluded, overdrawn, and hard to perceive display. IP Matrix and IDS Rainstorm, for example, utilize pixels to display a large IP range. Firstly, pixels are not informative and cumbersome for the user to interact with and secondly, since not all IPs are involved in a security event the IP space is not used wisely and a larger portion is left intact. NIVA, SnortView, and VizAlert also suffer from similar issues.

To our knowledge, no prior work has been carried out on heuristic host selection based on IDS alerts. The use of exponential moving averages in regards to alerts has been limited to [18], where EMA control charts have been used for alert reduction. The system has a very poor performance and is only capable of reducing alert types that have over 10,000 alerts.

In comparison, what accentuates Avisa from the aforementioned systems is its ability to identify hosts with interesting and often irregular behaviors and to discard hosts that experience constant alert activities. Based on multiple heuristic functions described in Sect. 3, hosts with greater behavioral changes over multiple intervals often receive higher scores as appose to hosts that have little or constant activity associated with them. We consider this as a major contribution of our work and an advantage over previous proposals that can also facilitate in false positive reduction.

7 Conclusion and Future Work

In this paper we have presented Avisa, a network security visualization system that can assist in comprehending IDS alerts and detecting abnormal pattern activities within a network. We have also thoroughly described and evaluated the proposed heuristic functions as our solution to visual space constraints. The underlying concept behind the heuristics can nonetheless be reused in various other systems. We have evaluated the system with real-world attacks and have shown how Avisa can be used to illustrate the attacks and visually correlate the events. Future work will be focused towards enhancing the visual capabilities of the system; being able to see detailed information regarding alerts; optimizing Spline calculations to reduce overhead; refining the heuristic functions and applying more rigorous methods to evaluate the system, including an intense usability test.

References

1. Morin, B., Mé, L., Debar, H., Ducassé, M.: M2D2: A formal data model for IDS alert correlation. In: Wespi, A., Vigna, G., Deri, L. (eds.) RAID 2002. LNCS, vol. 2516, pp. 115–137. Springer, Heidelberg (2002)
2. Debar, H., Wespi, A.: Aggregation and correlation of intrusion-detection alerts. In: Lee, W., Mé, L., Wespi, A. (eds.) RAID 2001. LNCS, vol. 2212, pp. 85–103. Springer, Heidelberg (2001)
3. Shin, M., Kim, E., Ryu, K.: False alarm classification model for network-based intrusion detection system. In: Yang, Z.R., Yin, H., Everson, R.M. (eds.) IDEAL 2004. LNCS, vol. 3177, pp. 259–265. Springer, Heidelberg (2004)
4. Cuppens, F., Miege, A.: Alert correlation in a cooperative intrusion detection framework. In: 2002, IEEE Symposium on Security and Privacy, Proceedings (2002)

5. Valeur, F., Vigna, G., Kruegel, C., Kemmerer, R.: Comprehensive approach to intrusion detection alert correlation. IEEE Transactions on Dependable and Secure Computing (2004)
6. Ware, C.: Information Visualization: Perception for Design. Morgan Kaufmann Publishers Inc., San Francisco (2004)
7. Marty, R.: Applied Security Visualization. Addison-Wesley Professional, Reading (2008)
8. Draper, G., Livnat, Y., Riesenfeld, R.: A survey of radial methods for information visualization. IEEE Transactions on Visualization and Computer Graphics (2009)
9. Hoagland, J., Staniford, S.: Viewing IDS alerts: lessons from SnortSnarf. In: Proceedings DARPA Information Survivability Conference and Exposition II (2001)
10. Danyliw, R.: Analysis console for intrusion databases (acid) (January 2001)
11. Nyarko, K., Capers, T., Scott, C., Ladeji-Osias, K.: Network Intrusion Visualization with NIVA, an Intrusion Detection Visual Analyzer with Haptic Integration. In: International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (2002)
12. Koike, H., Ohno, K.: SnortView: visualization system of snort logs. In: Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security, vol. 29, pp. 143–147. ACM, New York (2004)
13. Koike, H., Ohno, K., Koizumi, K.: Visualizing cyber attacks using IP matrix. In: Proceedings of the IEEE Workshops on Visualization for Computer Security (2005)
14. Abdullah, K., Lee, C., Conti, G., Copeland, J., Stasko, J.: IDS rainStorm: visualizing IDS alarms. In: IEEE Workshop on Visualization for Computer Security, VizSEC 2005, pp. 1–10. IEEE, Los Alamitos (2005)
15. Livnat, Y., Agutter, J., Moon, S., Erbacher, R., Foresti, S.: A visualization paradigm for network intrusion detection. In: Proceedings of the IEEE Information Assurance Workshop (2005)
16. Foresti, S., Agutter, J.: VisAlert: From Idea to Product. In: VizSEC. Mathematics and Visualization (2007)
17. Musa, S., Parish, D.: Using Time Series 3D AlertGraph and False Alert Classification to Analyse Snort Alerts. In: Visualization for Computer Security (2008)
18. Viinikka, J., Debar, H.: Monitoring IDS Background Noise Using EWMA Control Charts and Alert Information. In: Jonsson, E., Valdes, A., Almgren, M. (eds.) RAID 2004. LNCS, vol. 3224, pp. 166–187. Springer, Heidelberg (2004)