Situational Assessment of Intrusion Alerts: A Multi Attack Scenario Evaluation

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. In this research study, we focus on intrusion alerts and the burden of analyzing numerous security events by network administrators. We present Avisa$_2$, a network security visualization system that can assist in the comprehension of IDS alerts and detection of abnormal pattern activities. The quantity of security events triggered by modern day intrusion systems, accompanied by the level of complexity and lack of correlation between events, limits the human cognitive process in identifying anomalous behavior. This shortcoming induces the need for an automated process that would project critical situations and prioritize network hosts encountering peculiar behaviors. At the heart of Avisa$_2$ lies a collection of heuristic functions that are utilized to score, rank, and prioritize internal hosts of the monitored network. We believe this contribution elevates the practicality of Avisa$_2$ in identifying critical situations and renders it to be far superior to traditional security systems that solely focus on visualization. The effectiveness of Avisa$_2$ is evaluated on two multi-stage attack scenarios; each intentionally focused on a particular attack type, network service, and network range. Avisa$_2$ proved effective and accurate in prioritizing hosts under attack or hosts in which attacks were performed from.

Keywords: Visualization, IDS Alerts, Situational Awareness, Heuristic Function, Exponential smoothing.

1 Introduction

Intrusion signatures are continuously updated to generalize the behavior of a known exploit rather than to be targeted towards a specific malware. As a consequence of this behavior, a higher volume of legitimate traffic is flagged as malicious, generating a higher number of intrusion alerts leading to the phenomenon of false positives. A major issue that false positives create is that they can easily distort legitimate alerts from being seen by an administrator. Network security visualization is an emerging field that has been developed with these shortcomings in mind. Security visualization accentuates fundamental matters

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

S. Qing et al. (Eds.): ICICS 2011, LNCS 7043, pp. 399–413, 2011.

of information visualization and synthesizes it with security audit traces, demanding novel techniques for the purpose of exploratory analysis [1]. The main goal of security visualization is to give insight with which the ability to identify, process, and comprehend malicious behavior is achieved. However, the human visual system has rules of its own. We can only perceive patterns, trends, and exceptions if they are displayed in certain ways, or in better words, if they obey the rules of the human visual system [2]. In order to design visualizations that exploit this fast and powerful processor we need to find features that can be perceived rapidly, properties that are good discriminators, and characteristics that abide by the laws of our visual system. This in return allows for a more effective analysis of complex data while enhancing the situational awareness of the security analyst.

Situational awareness is viewed in [3] as a "state of knowledge that results from a process" and must be distinguished from the process used to acquire that state. Subsequently, situation assessment is "the process" used to achieve that knowledge and is considered an aid in the cognitive process of situation awareness. A situation assessment process must automatically identify and evaluate the impact of underlying events and relate them to assets of the monitored network.

In this paper, we utilize a collection of time based, parameterized heuristic functions as the basis of our situation assessment component to collectively identify and prioritize hosts of peculiar behavior. The output of the situation assessment component- a collection of hosts within the monitored network with a higher abnormality score- is then visualized through a novel security visualization system. The situation assessment component is evaluated on two multi-step attack scenarios executed on our Centre's benchmark dataset, each carefully crafted and aimed towards recent trends in security threats.

In this paper, we make the following contributions;

- Formalization of parameterized heuristic functions as the basis of a situation assessment component to combat constraints imposed on conventional security visualization systems.
- Design and implementation of a novel security visualization system for displaying a selective number of hosts and their corresponding alerts in an interactive and exploratory manner.

The remainder of this paper is organized as follows. Section 2 looks deep into security visualization as issues and concerns regarding modern network security visualization systems are elaborated on. Section 3 articulates the philosophy of incorporating heuristic functions as an automated process of estimating and projecting critical situations. Section 4 introduces seven distinct features utilized in identifying hosts with malicious behavior. In Section 5 the host selection algorithm is proposed and its functionality is described in length. In Section 6, we express the proposed visualization system, Avisa$_2$, with details regarding its design. The visualization system and its underlying heuristic functions are evaluated in Section 7. The paper is summarized in Section 8 with suggestions for future work and further extensions.

2 Limitations of Security Visualization Systems

A class of visualization systems, namely [4,5], have focused greatly on not only visualizing the state of one or a limited number of hosts as seen in [6], but on depicting the interaction of a large number of internal hosts with respect to external sources. This class solely focuses on visualization techniques to combat large quantities of network related data, often resulting in occlusion as most systems are largely faced with scalability issues. This fact reiterates the need for a process that can identify hosts with anomalous behavior and to project the processed results on a visualization system. In this manner, the load on the visualization system is reduced considerably; allowing for a near real time analysis of events and thereby a more responsive system is accessible.

As apposed to the aforementioned systems, where the emphasis is mainly on higher level activity of hosts, a collection of visualization systems are geared towards visualizing the port activity of a single or a collection of hosts within a monitored network [7]. Developers of these system assert that various malware programs often manifest themselves in abnormal port activity which can be detected through visualization systems. This argument may have been correct in the past, but as applications tend to evolve over time and adjust how they communicate over the Internet, they become increasingly evasive. Almost two thirds of all enterprise traffic is currently routed through ports 80 (HTTP) and 443 (HTTPS) [8]. This change in behavior greatly influences the objectives of port activity visualization systems as their focus should shift towards in-depth analysis of only a predominant number of ports rather than depicting the activity of the full port range.

The fascinating ability of visualization in providing insight into the attack detection process should be considered as the main contribution of a security visualization system. Current visualization systems devised for the process of detecting attack patterns [9], are in most cases used independent of other security products in a network. Visualization systems should be thought as systems that provide insight into areas that other security systems fail to enlighten. Any malicious behavior detected should then be analyzed and automated, if possible, so that an automated application can handle the task in future; conserving human time and attention.

3 Enhancing Situation Awareness via Automated Heuristic Functions

In this study we have taken an approach to decrease the amount of visual clutter by decreasing the number of hosts and consequently reducing the number of alerts displayed at each interval through a situation assessment process comprised of multiple heuristic functions. These functions are further elaborated below.

3.1 Exponential Smoothing

In an exponential moving average the effect of recent values is expected to decline exponentially over time to mitigate the effects of extreme observations. Let $\{y_t\} = \{\cdots y_{t-1}, y_t, y_{t+1} \cdots\}$ denote a sequence of non-negative real numbers indexed by some time subscript t. Call such a sequence of variables a time series. An n-period exponential smoothing of a time series y_t is defined as

$$\tilde{Z}_t\left(n\right) = \sum_{j=0}^{n-1} w_j \cdot y_{t-j} \ , \quad w_j = \frac{\alpha^{j-1}}{\sum_{j=0}^{n-1} \alpha^{j-1}} \tag{1}$$

where $0 \leq \alpha < 1$ is the smoothing constant. As $n \to \infty, \alpha^n \to 0, w_n \to 0$, Equation 1 can be defined independently of the window width n. The geometric decline in Equation 1 can be calculated efficiently using recursion

$$\tilde{Z}_t\left(\alpha\right) = \left(1 - \alpha\right) y_t + \alpha \tilde{Z}_{t-1}\left(\alpha\right) \tag{2}$$

where y_t is the observation at time t, $\tilde{Z}_{t-1}$ is the value of the exponential smoothing in the previous period, and $\tilde{Z}_t$ is the value of exponential smoothing at time t. The smoothing constant, α, controls the memory of the process such that the smaller the smoothing constant, the more weight is given to recent observations.

3.2 Exponential Smoothing Difference

In this category of heuristic function, emphasis is put toward the changing behavior of a feature's value rather than the absolute value itself. Running an exponential moving average over the difference of the current and previous values of a feature provides a means to filter constant activity and to reward increasing values. An n-period exponential smoothing difference of a time series y_t is defined recursively as

$$\tilde{D}_t\left(\alpha\right) = \left(1 - \alpha\right)\left(y_t - y_{t-1}\right) + \alpha \tilde{D}_{t-1}\left(\alpha\right) \tag{3}$$

where $0 \leq \alpha < 1$ is the smoothing constant, y_t is the observation at time t, y_{t-1} is the observation at time $t - 1$, $\tilde{D}_{t-1}$ is the value of the exponential smoothing difference in the previous period, and $\tilde{D}_t$ is the value of the exponential smoothing difference at time t.

3.3 Dispersion

A measure of dispersion can give a numerical indication of how scattered, or concentrated, a collection of events are over a certain period of time. The most commonly used measure of dispersion is the sample standard deviation, s, the square root of the sample variance given by

$$s = \sqrt{\frac{1}{n-1} \sum_{i=1}^{n} \left(x_i - \overline{x}\right)^2} \tag{4}$$

where $x_1, x_2, \cdots, x_n$ are the n samples observations and $\overline{x}$ is the sample mean.

4 Distinctive Set of Features

Let $r(R)$ denote a relation on the relation schema $R(A_1, A_2, \cdots, A_n)$, where $\{A_1, A_2, \cdots, A_n\}$ is a set of attributes. Also, let D_i denote the domain of permitted values of attribute A_i. A relation r is a set of n-tuples $(a_1, a_2, \cdots, a_n)$ where each $a_i \in D_i$. This paper models alerts as relation $\varLambda$, where $\varLambda$ is a subset of the Cartesian product of the domains of its attributes. Based on this definition and for the attribute set $\{ID, Category, SrcIP, Time, \cdots\}$, current time window $\tau_1 = t_a \leq Time < t_b$, and prior time window $\tau_0 = t_0 \leq Time < t_a$ the following seven features have been defined:

(1) $\mathcal{A}^i_{\tau_1} := F_{count(ID)} \left(\sigma_{Time=\tau_1} \left(\varLambda_i\right)\right)$
(2) $\mathcal{AC}^i_{\tau_1} :=_{Category} F_{count(*)} \left(\sigma_{Time=\tau_1} \left(\varLambda_i\right)\right)$
(3) $\mathcal{S}^i_{\tau_1} := F_{count(SrcIP)} \left(\pi_{SrcIP} \left(\sigma_{Time=\tau_1} \left(\varLambda_i\right)\right)\right)$

(4) $\mathcal{PS}^i_{\tau_1,\tau_0} := \frac{F_{count(SrcIP)}(\pi_{SrcIP}(W^i_{\tau_1} - W^i_{\tau_0}))}{F_{count(SrcIP)}(\pi_{SrcIP}\left(W^i_{\tau_1}\right))}, \; W^i_\tau := \pi_{SrcIP} \left(\sigma_{Time=\tau} \left(\varLambda_i\right)\right)$

(5) $\mathcal{C}^i_{\tau_1} := F_{count(AlertType)} \left(\pi_{AlertType} \left(\sigma_{Time=\tau_1} \left(\varLambda_i\right)\right)\right)$

(6) $\mathcal{PC}^i_{\tau_1,\tau_0} := \frac{F_{count(Type)}(\pi_{Category}(V^i_{\tau_1} - V^i_{\tau_0}))}{F_{count(Category)}(\pi_{Category}(V^i_{\tau_1}))}, \; V^i_\tau := \pi_{Category} \left(\sigma_{Time=\tau} \left(\varLambda_i\right)\right)$

(7) $\mathcal{AT}^i_{\tau_1} := \pi_{Time} \left(\sigma_{Time=\tau_1} \left(\varLambda_i\right)\right),$

5 Heuristic Host Selection Algorithm

Algorithm 1 describes the heuristic host selection procedure. The algorithm takes as arguments a set of IDS generated alerts $\varLambda$ from the current time window $t_a \leq \tau_1 < t_b$ and a set of features $\mathcal{F}$ accompanied with their respective user defined weights $\mathcal{W}$. The procedure outputs the top n hosts with the highest abnormality scores. The host selection procedure is performed in two major steps:
(1) For each host $i \in \mathcal{H}$ within the IDS alert stream input in the current time window (τ_1), three heuristic functions as defined in Section 3 are calculated on the set of features $\mathcal{F} = \{\mathcal{A}, \mathcal{AC}, \mathcal{S}, \mathcal{PS}, \mathcal{C}, \mathcal{PC}, \mathcal{AT}\}$ (Lines 3-14). The exponential smoothing is calculated on the first six features $(\widetilde{Z^{ij}_{\tau_1}})$ (Lines 4-6), the exponential smoothing difference on the first two features $(\widetilde{D^{ij}_{\tau_1}})$ (Lines 7-9), and the dispersion heuristic on the last feature $(s^{ij}_{\tau_1})$ (Lines 10-14).
(2) The final score of each host is composed of the sum of three components: sum of exponential smoothings $(S^{\tau_1}_{\tilde{Z}_i})$, sum of exponential smoothing differences $(S^{\tau_1}_{\tilde{D}_i})$, and sum of dispersion $(S^{\tau_1}_{\tilde{s}_i})$ (Lines 15-25). The value of each component is calculated by multiplying the normalized value of a feature $(\overline{Z^{ij}_{\tau_1}}, \overline{D^{ij}_{\tau_1}}, \overline{s^{ij}_{\tau_1}})$ by its respective user defined weight$(\mathcal{W}_j)$ and subsequently summing them for all features of a heuristic category (Line 24).

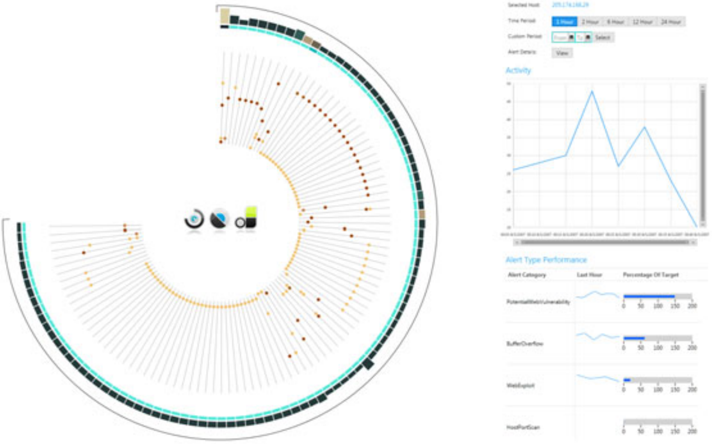
In the final step the algorithm outputs the top n hosts with the highest scores.

Algorithm 1. Heuristic Host Selection Algorithm

```
Input: Set of Alerts $\Lambda$, Set of Features $\mathcal{F} = \{\mathcal{A}, \mathcal{AC}, \mathcal{S}, \mathcal{PS}, \mathcal{C}, \mathcal{PC}, \mathcal{AT}\}$, Set of
       user defined weights $\mathcal{W} = \left\{w^Z_{\mathcal{A}}, w^Z_{\mathcal{AC}}, w^Z_{\mathcal{S}}, w^Z_{\mathcal{PS}}, w^Z_{\mathcal{C}}, w^Z_{\mathcal{PC}}, w^D_{\mathcal{A}}, w^D_{\mathcal{AC}}, w^s_{\mathcal{AT}}\right\}$,
       $t_0 \leq \tau_0 < t_a$, $t_a \leq \tau_1 < t_b$, $n$.
Output: Top $n$ hosts with highest scores.
begin
    $\mathcal{H} \longleftarrow \pi_{DIP}(\sigma_{Time=\tau_1}(\Lambda))$
    // Calculate heuristic function for each host
    foreach $i \in \mathcal{H}$ do
        // Exponential smoothing of first six features
        for $j \longleftarrow 1\ to\ 6$ do
            $\tilde{Z}^{ij}_{\tau_1} \longleftarrow (1-\alpha)\,\mathcal{F}^{ij}_{\tau_1} + \tilde{Z}^{ij}_{\tau_0}$
        end
        // Exponential smoothing difference of first two features
        for $j \longleftarrow 1\ to\ 2$ do
            $\tilde{D}^{ij}_{\tau_1} \longleftarrow (1-\alpha)\,(\mathcal{F}^{ij}_{\tau_1} - \mathcal{F}^{ij}_{\tau_0}) + \tilde{D}^{ij}_{\tau_0}$
        end
        // Dispersion of last feature
        $j \longleftarrow 7$
        $k \longleftarrow \left|\mathcal{F}^{ij}_{\tau_1}\right|$
        $\overline{x} \longleftarrow \frac{\sum_{x \in \mathcal{F}^{ij}_{\tau_1}} x}{k}$
        $s^{ij}_{\tau_1} \longleftarrow \sqrt{\frac{1}{k-1} \sum_{x \in \mathcal{F}^{ij}_{\tau_1}} (x - \bar{x})^2}$
    end
    // Calculate score for each host
    foreach $i \in \mathcal{H}$ do
        // Sum score for normalized value of exponential smoothings
        for $j \longleftarrow 1\ to\ 6$ do
            $S^{\tau_1}_{\tilde{Z}_i} \longleftarrow S^{\tau_1}_{\tilde{Z}_i} + (\mathcal{W}_j \cdot \overline{Z^{ij}_{\tau_1}})$
        end
        // Sum score for normalized value of exponential smoothing
           differences
        for $j \longleftarrow 7\ to\ 8$ do
            $S^{\tau_1}_{\tilde{D}_i} \longleftarrow S^{\tau_1}_{\tilde{D}_i} + (\mathcal{W}_j \cdot \overline{D^{ij}_{\tau_1}})$
        end
        // Sum score for normalized value of standard deviation
        $j \longleftarrow 9$
        $S^{\tau_1}_{\tilde{s_i}} \longleftarrow S^{\tau_1}_{\tilde{s_i}} + (\mathcal{W}_j \cdot \overline{s^{ij}_{\tau_1}})$
        // Sum final score of host
        $Score^i_{\tau_1} \longleftarrow S^{\tau_1}_{\tilde{Z}_i} + S^{\tau_1}_{\tilde{D}_i} + S^{\tau_1}_{\tilde{s_i}}$
    end
    // Return top n hosts with highest scores
    return $top(Score_{\tau_1}, n)$
end
```

6 Avisa$_2$: A Network Security Visualization System

A screen shot of Avisa$_2$ in action is illustrated Fig. 1. The system is composed of two main components, the *radial visualization* on the left and the *information stack* on the right. Both components work in collaboration with each other to maintain effective security situational awareness. This enables a rapid assessment and investigation of relevant security events through direct user interaction and analysis. Avisa$_2$ presents an up-to-date display of network state by providing an interactive visualization of real-time security events. This, combined with the automatic situational assessment powers of the heuristic functions presents an ideal visualization system that is capable of displaying prioritized situations to security analysts for a better situation awareness.



Selected Host: 205.174.168.29

Time Period: 1 Hour | 2 Hour | 6 Hour | 12 Hour | 24 Hour

Custom Period: From | To | Select

Alert Details: View

Activity

Alert Type Performance

Alert Category	Last Hour	Percentage Of Target
PotentialWebVulnerability		
BufferOverflow		
WebExploit		
HostPortScan		

Fig. 1. A screen shot of Avisa$_2$ in action. The radial visualization on the left is the focal point of the display, while the information stack on the right illustrates detail.

6.1 Radial Visualization

The radial visualization component of Avisa$_2$ constitutes the focal point of the display. As the output of the heuristic functions are calculated in 5 minute intervals, the top n hosts of the monitored network are selected and are piped as input to the visual component. The radial visualization creates an interactive environment for analysts to perceive patterns, trends, and exceptions within the already prioritized data. Primitive attributes are the unique properties that allow a visual element to be seen from an image. Primitive attributes such as hue, motion, size, length, intensity, and spatial grouping are used extensively to establish visual prominence. The radial visualization itself is composed of two subcomponents, namely the *network host radial panel* and the *alert category dot panel.*

Network Host Radial Panel. The *network host radial panel* is designed to represent the output of the heuristic functions along with several attributes in a perceivable fashion. The prioritized network hosts are arranged along a radial panel while the final quarter of the panel is reserved for displaying additional information. As hosts are added or removed from the panel, they are animated in place to assist in highlighting system transitions from one state to another. The primitive attributes *length* and *color* are utilized to visually differentiate hosts with higher and lower scores. At each time interval the height, color, and position of each host is animated from its previous value to its current value. This is a feature that is rarely seen in security visualization systems due to the selected development framework and complicated implementation issues surrounding animation.

Alert Category Dot Panel. The *alert category dot panel* is designed to encode the alert activity of a host. For each alert category a circle shaped element is displayed along a vertical line. Currently the panel is capable of displaying twenty alert categories with room for further extensions. In order to encode the number of alerts in each category as a color, the number of alerts in each alert category is normalized over all hosts. Accordingly, the values are arranged into equal length intervals while each interval is assigned a color. In this manner, it is very clear for an analyst to see the different types of alerts a host or a collection of hosts are experiencing in one glance. Consequently, based on the assigned colors an analyst can also grasp an idea on the number of alerts and if further detailed information is required, she can use the *information stack* on the right side of the system for further analysis.

6.2 Information Stack

The *information stack* works in collaboration with the radial visualization component to provide a greater insight into the underlying data. When a user selects a host on the radial panel, the information stack queries for the required data and displays it in an informative and interactive manner. The stack is composed of three main components: the *date time selector*, the *activity graph*, and the *alert type performance table*. The *date time selector* displays a panel of predefined or custom time periods for an analyst to select. The toggle buttons are data bound to the underlying data and when pressed, the stack is updated dynamically. The *activity graph* displays a zoomable line chart and is used to display alert activity in greater detail. The *alert type performance table* displays an overview of the alert categories experienced by the selected host by incorporating sparklines and bullet graphs. Sparklines provide a bare-bones and space efficient time-series graph. Selecting an alert category from the performance table displays its respective sparkline graph in greater detail on the *activity graph*. A simplified version of the bullet graph is also used to provide a comparison mechanism for the number of alerts in each category. Avisa$_2$ also provides low level details of the actual IDS alerts through the *Alert Detail* button of the stack.

7 Experimental Results and Evaluation

7.1 Scenario 1: Network Infiltration from the Inside

It is very common for computers on a network to access the Internet through a NAT server. This attack scenario is designed to show how a network with all workstations located behind a NAT can be infiltrated. In this case, while the target computers will be able to make connections to the Internet, it will not be possible to establish a connection from outside to the target network. Thereby, client-side techniques such as executable encoding, host pivoting, social engineering, and shell migration are utilized to exploit vulnerabilities on internal hosts and servers. Figure 2 provides a detailed illustration of the attack scenario. Each stage of the attack scenario accompanied with the analysis of Avisa$_2$'s results is articulated below and depicted in Fig. 3. The output score of each host within the time window of Scenario 1 is also detailed in Appendix A.

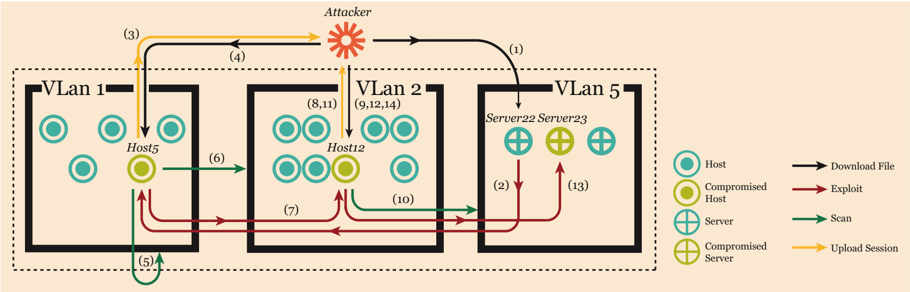


Fig. 2. An infographic detailing the multiple stages of attack scenario 1

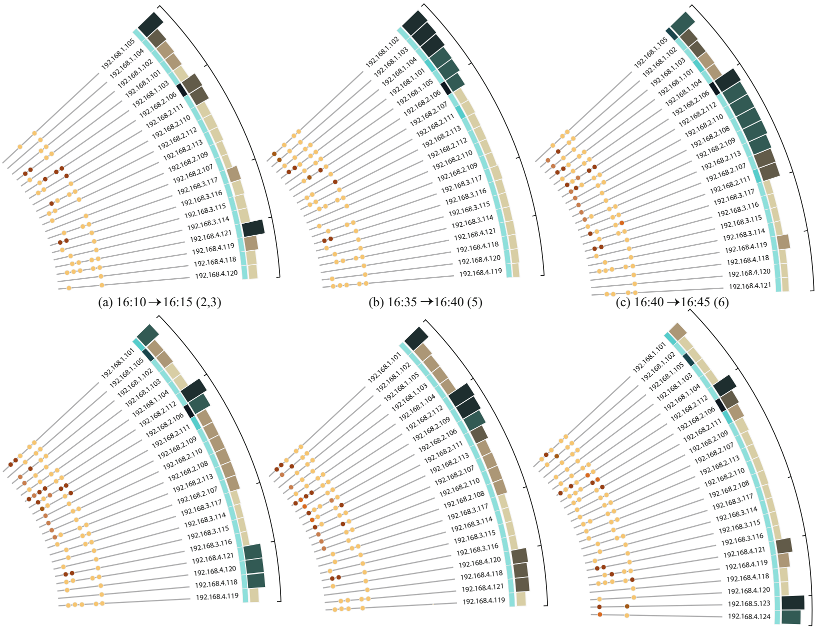
(1) 15:30→15:35: A corrupt PDF file, containing a TCP connection binary, is sent as an email attachment to all testbed users.
(2,3) 16:10→16:15: Host 192.168.1.105 opens the corrupted PDF file, an Adobe PDF vulnerability is exploited and a Meterpreter session is generated back to the attacker. Avisa$_2$ detects this and as shown in Fig. 3(a) and Table 1(a), host5 is ranked first and has received the highest score.
(4) 16:25→16:30: Nmap directory is download on host 192.168.1.105.
(5) 16:35→16:40: Nmap is run to scan subnet 192.168.1.0/24 of testbed from host 192.168.1.105. Port scans often trigger numerous alerts and as illustrated in Fig. 3(b) and Table 1(b), Avisa$_2$ has prioritized users of subnet 1.
(6) 16:40→16:45: Nmap is run to scan subnet 192.168.2.0/24 of testbed from host 192.168.1.105. Avisa$_2$ picks up on this behavior and as illustrated in Fig. 3(c) and Table 1(c), users of subnet 2 have received the highest scores.
(7,8) 16:45→16:50: SMB vulnerability on host 192.168.2.112 is exploited and a Meterpreter session is generated back to the attacker. Avisa$_2$ detects this peculiar

behavior and as illustrated in Fig. 3(d) and Table 1(d), host12 has received the highest score and is ranked first.

(9,10) 16:55→17:00: Nmap directory is downloaded on host 192.168.1.112; Nmap is run to scan subnet 192.168.5.0/24 of testbed from host 192.168.2.112. Although snort correctly alerted on the exploit and the port scans, it failed to detect the Nmap directory being downloaded on host12 and only a number of false positives were generated. But as illustrated in Figures 3(e),(f) and Tables 1(e), Avisa$_2$ ranks host12 the highest in two consecutive time periods due to its peculiar behavior. Due to the port scans performed on subnet 5, servers 23 and 24 are ranked in the top 5 in Table 1(f) even though they have received no alerts in previous intervals.

(13) 17:10→18:30: Browser on host 192.168.1.112 connects to the internal web application running on 192.168.5.123 and starts performing SQL injection attacks. The attacker iterates through database tables and creates a new user account with administrator privileges. The *user* table of the database is subsequently deleted to disallow further logins. The SQL injection attacks were not detected by snort, but due to the remote desktop connection accessing server23, Avisa$_2$ was able to rank server23 as the first or second host in this period. This is seen in detail in Tables 1(j) and (k).

(14) 20:45→21:00: Attacker downloads a backdoor on host 192.168.2.112 and disconnects all established sessions and finishes the attack scenario.



(a) 16:10→16:15 (2,3)

(b) 16:35→16:40 (5)

(c) 16:40→16:45 (6)

(d) 16:45→16:50 (7,8)

(e) 16:50→16:55 (9,10)

(f) 16:55→17:00 (10)

Fig. 3. Screen shots of Avisa$_2$ displaying the stages of attack scenario 1

7.2 Scenario 2: HTTP Denial of Service

The second attack scenario is designed towards performing a stealthy, low bandwidth denial of service attack without the need to flood the network. We will be utilizing `Slowloris` as the main tool in this scenario as it has proven to make Web servers completely inaccessible using a single machine. Slowloris starts by making a full TCP connection to the remote server. The tool holds the connection open by sending valid, incomplete HTTP requests to the server at regular intervals to keep the sockets from closing. Since any Web server has a finite ability to serve connections, it will only be a matter of time before all sockets are used up and no other connection can be made. This scenario picks up where scenario 1 left off by connecting to the backdoor created in the final stage. A detailed description on each stage of the attack scenario accompanied with the analysis of Avisa$_2$'s results and subsequent screen shots are given below. The output score of each host within the time window of Scenario 2 is also illustrated in Appendix B.

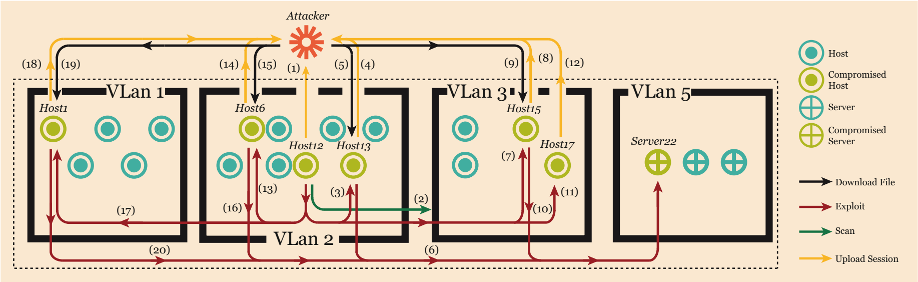


Fig. 4. An infographic detailing the multiple stages of attack scenario 2

(1) 16:55→17:00: Host 192.168.2.112 makes an outbound connection to the attacker through a backdoor. Snort detects host12 connecting to the attacker's machine and subsequently, and as illustrated in Table 2(a), Avisa$_2$ ranks host12 the top host.
(2) 17:15→17:20: Nmap is run to scan subnet 192.168.3.0/24 of testbed from host 192.168.2.112. Due to the scan on subnet 3, hosts 14,15,16, and 17 have received higher scores and as shown in Fig. 5(a) and Table 2(b), they are ranked amongst the top 5 hosts.
(3,4) 17:20→17:25: SMB vulnerability is exploited on host 192.168.2.113 and a remote desktop connection is returned to the attacker. This exploit is partially detected by snort, but since the number of alerts generated in the previous period is substantial, hosts 17,12, and 14 remain top hosts while host13 is ranked 4th. Figure 5(b) depicts this behavior.
(5) 17:25→17:30: Host 192.168.2.113 downloads malicious files from remote server.

(6) 17:30→17:35: Slowloris is run from host 192.168.2.113 against server 192.168.5.122. Even though the required signatures for detecting `Slowloris` attacks are turned on, Snort is unable to detect this attack. As a result of shutting the server down, hosts are unable to access the site and Snort triggers a collection of alerts. Avisa$_2$ is able to pick up on this behavior and as illustrated in Fig. 5(c) and Table 2(d), server22 is ranked second in the period under attack.

(7,8) 17:35→17:40: SMB vulnerability is exploited on host 192.168.3.115 and a remote desktop connection is returned to the attacker. Avisa$_2$ detects this behavior and as illustrated in Table 2(e), host15 is ranked second.

(9) 17:40→17:45: Host 192.168.2.115 downloads malicious files from remote server.

(10) 17:45→17:50: Slowloris is run from host 192.168.3.115 against server 192.168.5.122. Avisa$_2$ is able to pick up on this behavior and as illustrated in Table 2(f), host15 and server22 are amongst the top 3 hosts in the period under attack.

(11,12) 17:50→17:55: SMB vulnerability is exploited on host 192.168.3.117 and a remote desktop connection is returned to the attacker-Conncetion Lost. Avisa$_2$ detects this behavior and as illustrated in Table 2(g) host17 is ranked first.

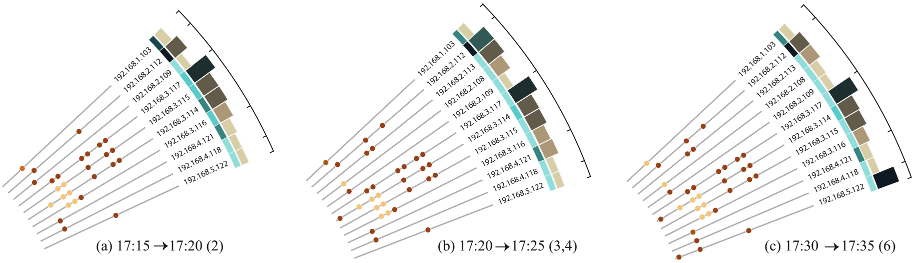
(13,14) 18:00→18:05: SMB vulnerability is exploited on host 192.168.2.106 and a remote desktop connection is returned to the attacker. Avisa$_2$ detects this behavior and as illustrated in Table 2(h) host6 is ranked second.

(15,16) 18:05→18:10: Slowloris is run from host 192.168.2.106 against server 192.168.5.122. Avisa$_2$ is able to pick up on this behavior and as illustrated in Table 2(i), host6 and server22 are amongst the top 3 hosts in the period under attack.

(17,18) 18:10→18:15: SMB vulnerability is exploited on host 192.168.1.101 and a remote desktop connection is returned to the attacker. Avisa$_2$ detects this behavior and as illustrated in Table 2(j) host1 is ranked second.

(19)18:15→18:20: Host 192.168.1.101 downloads malicious files from remote server.

(20) 18:20→18:25: Slowloris is run from host 192.168.1.101 against server 192.168.5.122. Avisa$_2$ is able to pick up on this behavior and as illustrated in Table 2(k), host1 and server22 are amongst the top 3 hosts in the period under attack.



(a) 17:15 →17:20 (2)

(b) 17:20→17:25 (3,4)

(c) 17:30 →17:35 (6)

Fig. 5. Screen shots of Avisa$_2$ displaying the stages of attack scenario 2

8 Conclusion

In this research we presented Avisa$_2$, a network security visualization system that can assist in comprehending IDS alerts and detecting abnormal pattern activities within a network. Visual constraints, complexity of relations between intrusion alerts, and limitations on perceiving situational awareness in high volume environments were the driving force behind the development of the heuristic functions. Three categories of heuristic functions along with seven heuristic features were introduced and formalized. The effectiveness of Avisa$_2$ in detecting malicious and abnormal behavior was evaluated on two multi-step attack scenarios, each intentionally focused on a particular attack type, network service, and network range. Avisa$_2$ was capable of prioritizing hosts that were the subject of attacks or hosts on which the attacks were executed. The effectiveness of Avisa$_2$ is reliant primarily on the detection of the underlying IDS, or in formal terms, its true positive rate. However, this does not mean that the false positive rate of the IDS must also be low, as the heuristic functions of Avisa$_2$ are capable of filtering and eliminating recurring events and prioritizing hosts receiving alerts from multiple sources and types.

Acknowledgment. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) through a grant STPGP-381091 to Dr. Ghorbani.

References

1. Shiravi, H., Shiravi, A., Ghorbani, A.A.: A survey of visualization systems for network security. IEEE Transactions on Visualization and Computer Graphics 99(PrePrints) (2011)
2. Few, S.: Now You See It: Simple Visualization Techniques for Quantitative Analysis, 1st edn. Analytics Press (2009)
3. Endsley, M.: Toward a theory of situation awareness in dynamic systems: Situation awareness. Human Factors 37(1), 32–64 (1995)
4. Ball, R., Fink, G.A., North, C.: Home-centric visualization of network traffic for security administration. In: Proceedings of the ACM Workshop on Visualization and Data Mining for Computer Security, pp. 55–64 (2004)
5. Goodall, J.R., Lutters, W.G., Rheingans, P., Komlodi, A.: Preserving the big picture: visual network traffic analysis with tnv. In: IEEE Workshop on Visualization for Computer Security (VizSEC 2005), pp. 47–54 (2005)
6. Erbacher, R., Walker, K., Frincke, D.: Intrusion and misuse detection in large-scale systems. IEEE Computer Graphics and Applications, 38–48 (2002)
7. McPherson, J., Ma, K., Krystosk, P., Bartoletti, T., Christensen, M.: PortVis: a tool for port-based detection of security events. In: Proceedings of the ACM Workshop on Visualization and Data Mining for Computer Security, pp. 73–81 (2004)
8. PaloAltoNetworks: Re-Inventing Network Security (2010), `http://www.paloaltonetworks.com/literature/whitepapers/Re-inventing-Network-Security.pdf` (online; accessed July 12, 2011)

9. Shiravi, H., Shiravi, A., Ghorbani, A.: Ids Alert Visualization and Monitoring through Heuristic Host Selection. In: Soriano, M., Qing, S., López, J. (eds.) ICICS 2010. LNCS, vol. 6476, pp. 445–458. Springer, Heidelberg (2010)

Appendix A: Output scores of Avisa$_2$ in attack scenario 1

Table 1. Output scores of Avisa$_2$ in attack scenario 1

TIME: 16:10 → 16:15	
HOST	SCORE
192.168.1.105	3.62
192.168.4.121	3.35
192.168.2.106	2.53
192.168.2.111	2.32
192.168.1.104	2.19
192.168.4.119	1.99
192.168.1.102	1.94
192.168.1.101	1.74
192.168.3.117	1.73
192.168.2.110	1.48
192.168.3.116	1.48
192.168.2.112	1.48
192.168.1.103	1.48
192.168.4.118	1.48
192.168.3.115	1.48

(a)

TIME: 16:35 → 16:40	
HOST	SCORE
192.168.1.102	3.37
192.168.1.103	3.10
192.168.1.104	2.95
192.168.1.101	2.60
192.168.1.105	2.48
192.168.2.106	2.27
192.168.2.107	1.00
192.168.3.117	0.92
192.168.2.111	0.92
192.168.4.121	0.87
192.168.2.113	0.86
192.168.3.116	0.76
192.168.2.112	0.76
192.168.2.110	0.76
192.168.4.118	0.76

(b)

TIME: 16:40 → 16:45	
HOST	SCORE
192.168.2.106	3.27
192.168.2.112	2.63
192.168.2.110	2.63
192.168.2.108	2.63
192.168.2.109	2.63
192.168.1.105	2.52
192.168.2.113	2.27
192.168.2.107	2.13
192.168.1.102	2.11
192.168.2.111	1.82
192.168.1.103	1.70
192.168.1.101	1.62
192.168.1.104	1.62
192.168.4.119	1.60
192.168.3.117	0.75

(c)

TIME: 16:45 → 16:50	
HOST	SCORE
192.168.2.112	3.07
192.168.1.101	2.35
192.168.2.106	2.34
192.168.4.118	2.05
192.168.4.120	2.04
192.168.4.121	2.03
192.168.1.105	1.27
192.168.1.102	1.21
192.168.2.110	1.20
192.168.2.108	1.20
192.168.2.109	1.20
192.168.2.111	1.20
192.168.2.113	1.18
192.168.2.107	1.09
192.168.1.103	0.93

(d)

TIME: 16:50 → 16:55	
HOST	SCORE
192.168.2.112	2.60
192.168.2.109	2.53
192.168.1.101	2.39
192.168.2.106	2.12
192.168.4.120	1.60
192.168.4.118	1.45
192.168.4.121	1.45
192.168.1.102	1.26
192.168.2.111	1.23
192.168.2.110	1.00
192.168.2.108	1.00
192.168.1.105	0.97
192.168.2.107	0.97
192.168.1.103	0.94
192.168.2.113	0.92

(e)

TIME: 16:55 → 17:00	
HOST	SCORE
192.168.5.123	3.06
192.168.2.112	2.66
192.168.5.124	2.48
192.168.2.106	1.96
192.168.1.101	1.77
192.168.4.121	1.69
192.168.2.111	1.15
192.168.2.109	0.99
192.168.4.119	0.93
192.168.1.102	0.92
192.168.1.103	0.66
192.168.1.104	0.62
192.168.2.113	0.58
192.168.4.118	0.57
192.168.2.110	0.56

(f)

TIME: 17:00 → 17:05	
HOST	SCORE
192.168.2.112	2.29
192.168.1.101	2.28
192.168.5.123	2.20
192.168.4.121	2.19
192.168.2.106	2.01
192.168.5.124	1.94
192.168.2.111	1.38
192.168.1.102	1.04
192.168.2.109	1.03
192.168.1.103	0.78
192.168.1.104	0.73
192.168.1.105	0.71
192.168.2.113	0.70
192.168.2.110	0.67
192.168.2.108	0.67

(g)

TIME: 17:05 → 17:10	
HOST	SCORE
192.168.2.112	2.74
192.168.1.101	2.35
192.168.4.119	2.27
192.168.2.106	2.10
192.168.4.121	1.68
192.168.5.123	1.67
192.168.2.111	1.60
192.168.5.124	1.44
192.168.2.109	1.24
192.168.1.102	1.10
192.168.2.113	1.05
192.168.1.105	1.02
192.168.2.110	1.01
192.168.2.108	1.01
192.168.4.118	0.88

(h)

TIME: 17:10 → 17:15	
HOST	SCORE
192.168.5.123	3.16
192.168.2.106	2.05
192.168.1.104	1.97
192.168.2.112	1.66
192.168.1.101	1.57
192.168.4.119	1.30
192.168.4.121	1.11
192.168.2.111	1.02
192.168.5.124	0.94
192.168.1.102	0.87
192.168.2.109	0.74
192.168.4.118	0.63
192.168.2.113	0.62
192.168.1.103	0.60
192.168.2.110	0.57

(i)

TIME: 17:15 → 17:20	
HOST	SCORE
192.168.5.123	3.20
192.168.1.104	2.33
192.168.2.106	2.23
192.168.2.112	1.94
192.168.1.101	1.85
192.168.1.105	1.70
192.168.2.111	1.55
192.168.4.119	1.51
192.168.5.124	1.40
192.168.2.109	1.31
192.168.4.121	1.29
192.168.1.102	1.28
192.168.2.113	1.20
192.168.2.110	1.15
192.168.2.108	1.15

(j)

TIME: 17:20 → 17:25	
HOST	SCORE
192.168.4.121	2.84
192.168.5.123	2.74
192.168.1.101	1.90
192.168.2.106	1.82
192.168.2.112	1.60
192.168.2.111	1.23
192.168.1.102	1.22
192.168.1.104	1.22
192.168.1.105	1.13
192.168.5.124	1.02
192.168.1.103	0.97
192.168.4.119	0.92
192.168.2.109	0.82
192.168.2.107	0.81
192.168.4.118	0.76

(k)

Appendix B: Output scores of Avisa$_2$ in attack scenario 2

Table 2. Output scores of Avisa$_2$ in attack scenario 2

TIME: 16:55 → 17:00

HOST	SCORE
192.168.2.112	3.96
192.168.4.121	3.27
192.168.5.122	2.58
192.168.1.103	2.23
192.168.3.114	1.26
192.168.2.109	1.23
192.168.4.118	1.22

(a)

TIME: 17:15 → 17:20

HOST	SCORE
192.168.3.117	6.34
192.168.3.115	3.76
192.168.3.114	3.74
192.168.2.112	3.02
192.168.3.116	2.59
192.168.4.121	1.02
192.168.1.103	0.86
192.168.5.122	0.12
192.168.4.118	0.10
192.168.2.109	0.10

(b)

TIME: 17:20 → 17:25

HOST	SCORE
192.168.3.117	5.43
192.168.2.112	3.50
192.168.3.114	2.93
192.168.2.113	2.87
192.168.3.115	2.84
192.168.2.108	2.72
192.168.3.116	1.67
192.168.4.121	1.32
192.168.1.103	0.54
192.168.2.109	0.10
192.168.4.118	0.10
192.168.5.122	0.10

(c)

TIME: 17:30 → 17:35

HOST	SCORE
192.168.3.117	4.53
192.168.5.122	3.56
192.168.3.114	2.64
192.168.3.115	2.46
192.168.2.112	2.00
192.168.2.113	1.70
192.168.3.116	1.26
192.168.4.121	0.65
192.168.2.108	0.59
192.168.1.103	0.50
192.168.2.109	0.22
192.168.4.118	0.22

(d)

TIME: 17:35 → 17:40

HOST	SCORE
192.168.3.117	4.68
192.168.3.115	4.59
192.168.5.122	3.65
192.168.3.114	2.49
192.168.2.112	2.32
192.168.3.116	1.43
192.168.4.121	1.39
192.168.2.113	1.17
192.168.2.108	0.76
192.168.2.109	0.40
192.168.4.118	0.40
192.168.1.103	0.35

(e)

TIME: 17:45 → 17:50

HOST	SCORE
192.168.3.115	4.97
192.168.3.117	4.63
192.168.5.122	4.40
192.168.3.114	3.12
192.168.2.112	2.12
192.168.2.113	1.81
192.168.3.116	1.66
192.168.2.108	1.43
192.168.4.121	1.19
192.168.2.109	1.14
192.168.4.118	1.14
192.168.1.103	1.02

(f)

TIME: 17:50 → 17:55

HOST	SCORE
192.168.3.117	6.68
192.168.5.122	3.85
192.168.3.115	3.73
192.168.3.114	3.36
192.168.2.112	2.64
192.168.2.113	1.61
192.168.2.109	1.53
192.168.4.118	1.53
192.168.3.116	1.51
192.168.1.103	1.47
192.168.2.108	1.42
192.168.4.121	1.34

(g)

TIME: 18:00 → 18:05

HOST	SCORE
192.168.3.117	4.84
192.168.2.106	4.00
192.168.2.112	2.65
192.168.3.116	2.53
192.168.5.122	2.35
192.168.2.109	2.26
192.168.4.121	2.24
192.168.3.115	2.17
192.168.3.114	1.95
192.168.4.118	1.49
192.168.1.103	1.48
192.168.2.108	1.48
192.168.2.113	1.42

(h)

TIME: 18:05 → 18:10

HOST	SCORE
192.168.3.117	4.98
192.168.5.122	4.25
192.168.2.106	4.08
192.168.2.112	2.70
192.168.2.109	2.43
192.168.3.116	2.34
192.168.4.121	2.02
192.168.4.118	1.74
192.168.1.103	1.74
192.168.2.108	1.73
192.168.3.115	1.72
192.168.2.113	1.71
192.168.3.114	1.45

(i)

TIME: 18:10 → 18:15

HOST	SCORE
192.168.3.117	4.42
192.168.1.101	3.37
192.168.5.122	3.36
192.168.2.112	2.30
192.168.2.106	1.62
192.168.2.109	1.35
192.168.3.116	1.32
192.168.4.121	1.29
192.168.3.115	1.22
192.168.4.118	1.13
192.168.1.103	1.13
192.168.2.108	1.13
192.168.2.113	1.13
192.168.3.114	1.11

(j)

TIME: 18:20 → 18:25

HOST	SCORE
192.168.3.117	4.32
192.168.5.122	4.13
192.168.1.101	2.77
192.168.2.112	2.04
192.168.2.106	1.48
192.168.2.109	1.22
192.168.4.121	1.20
192.168.3.116	1.16
192.168.2.108	1.06
192.168.4.118	1.06
192.168.1.103	1.06
192.168.2.113	1.06
192.168.3.114	1.06
192.168.3.115	1.05

(k)