

A Scalable Aural-Visual Environment for Security Event Monitoring, Analysis, and Response*

Paul Z. Kolano

NASA Advanced Supercomputing Division, NASA Ames Research Center
M/S 258-6, Moffett Field, CA 94035 U.S.A.

`kolano@nas.nasa.gov`

Abstract. Intrusion detection systems gather large quantities of host and network information in an attempt to detect and respond to attacks against an organization. The widely varying nature of attacks makes humans essential for analysis, but the sheer volume of data can quickly overwhelm even experienced analysts. Existing approaches utilize visualization to provide rapidly comprehensible representations of the data, but fail to scale to real-world environments due to unrealistic data handling and lack of response facilities. This paper introduces a new tool for security event monitoring, analysis, and response called Savors. Savors provides suitable scalability by utilizing three additional areas of computing. High-end computing brings large amounts of on-demand processing to bear on the problem. Auralization allows both monitoring and analysis to be performed in parallel. Finally, grid computing provides the basis for remote data access and response capabilities with seamless and secure access to organization resources.

1 Introduction

Modern intrusion detection systems (IDS) utilize a collection of sensors to gather data about host and network activity within an organization. This information is then used to identify and analyse current and historical attacks. Sensor data must be correlated [18] to detect single attacks spread out across multiple hosts and networks. Data volume, especially in the network sensor case, can quickly consume even local storage resources, thus data must be significantly reduced before it can be sent to a centralized correlation engine. This creates a distributed information hierarchy where all the data necessary to analyze a given attack may not be present at the same location.

Once an intrusion has been verified, an appropriate response is necessary such as repairing existing damage, blocking traffic to prevent further damage, and adding filters to enhance detection of similar attacks. Taken together, the tasks of analyzing large security data sets, knowing where data resides and how to access it, knowing the network topology and how to access and configure devices within it, etc. can quickly lead to information overload for human analysts. To make intrusion detection more effective, it must be made scalable for human consumption.

* This work is supported by the NASA Advanced Supercomputing Division under Task Order NNA05AC20T (Contract GS-09F-00282) with Advanced Management Technology Inc.

A variety of tools have been proposed that use visualization techniques to provide greater amounts of information at once in a visual medium more easily grasped by humans. Existing tools, however, do not provide the scalability required for real-world security environments. In particular, they do not provide support for large data sets that may be split across multiple files or distributed across multiple hosts. Beyond basic filtering, they do not provide in-depth analysis capabilities to prune away the vast majority of normal traffic that can overwhelm visual displays. Finally, they do not provide a response capability to take actions based on results uncovered.

This paper presents a new tool called Savors, the **Scalable Aural-Visual Operations Room for Security**. Savors provides a scalable approach to monitoring, analyzing, and responding to arbitrarily sized and distributed security event data by combining four different areas of computing within five visualization tools, an auralization component, and supporting software. Visualization is used to provide quickly comprehensible graphical representations of the data that take advantage of the human ability to recognize visual patterns and anomalies. High-end computing is used to support computationally intensive anomaly calculations that highlight areas of interest. Auralization is used to increase human interface bandwidth by utilizing sound in parallel with sight. Finally, grid computing is used to seamlessly hook resources together, shielding the analyst from low level details while providing a strong security environment to protect resources.

This paper is organized as follows. Section 2 discusses related work. Section 3 gives an overview of Savors. Sections 4, 5, and 6 describe the monitoring, analysis, and response aspects of Savors. Finally, Section 7 discusses conclusions and future work.

2 Related Work

There are a variety of efforts related to the problem addressed by this paper. RUMINT provides a variety of security visualizations with DVR-like controls, which include binary rainfall displays [4] that map varying numbers of bits to pixels, parallel coordinate and scatter plots [3] between various protocol fields, and ASCII packet information.

IDS Rainstorm [1] shows the occurrence of IDS alarms across IP addresses and time. Multiple axes are used to visualize a large range of IP addresses with support for zooming in on a particular region to show increased detail. The IDS Rainstorm view is capable of utilizing every pixel for information, but is mainly useful for monitoring network state and not for providing additional insight into attacks.

The Visual Firewall [12] shows four simultaneous views of network traffic including parallel coordinate and animated glyph-based scatter plots between local ports and foreign IP addresses, a network throughput graph over time, and a quad-axis diagram mapping local IP address, foreign subnet, IDS alert type, and time. The concurrent display allows the user to consider four different perspectives of the same traffic at once, although only one view may be shown at full size.

The SIFT tool suite [20] consists of two main visualization tools. NVisionIP represents a complete class B network address space using a matrix of subnets and hosts. Users may perform hardcoded queries to see basic numeric characteristics, such as bytes per host, across all hosts. Users may also drill down to the host level to see more de-

tail. VisFlowConnect-IP shows a parallel coordinate plot between internal and external hosts while distinguishing between inbound and outbound traffic.

TNV [9] shows a grid of host activity over time with the middle portion of the grid devoted to a focus area and the outer portions showing the context. The focus area shows greater detail about hosts within a particular period of time including link communication and port activity. The context area shows color-based packet counts of the remaining time period to provide continuity between the focus area and the rest of the data. Surrounding the main grid are a historical summary and a control timeline as well as additional areas with detailed information about a specific host.

VisAlert [6] plots the time, location, and type of IDS alerts using a main circular area with a surrounding set of concentric circles. Event type is represented by regions along each outer ring where time increases with radius. Alert location is plotted from the types of the innermost ring to a graph of network topology in the main area. Persistent alerts and alert volume are indicated using color coding and graph node size.

Peep [8] auralizes network status using natural sounds that remain pleasing to the ear even when played continuously. Peep distinguishes between one-time events, periodic heartbeats, and specific states using representative sounds such as bird chirps, cricket chirps, and water flows, respectively. Peep clients generate sound information from local state such as system logs, which is mixed and played by the Peep server.

CyberSeer [15] incorporates auralization into its visualization component to enhance monitoring performance. The visualization component utilizes spectral analysis techniques, which are based on the observation of periodic behavior in network traffic. The auralization component synthesises sounds in response to changes in data patterns. CyberSeer further proposes an immersive 3D environment to take advantage of human spatial perception that would significantly increase aural and visual bandwidth.

3 Savors Overview

Savors consists of five visualization tools, an auralization component, and supporting software that together provide an integrated environment for monitoring, analyzing, and responding to security event data. Tools are split into client and server components, with the server responsible for locating and processing the requested data and streaming it to the client for display. Data is located using an administrator-defined function that returns the data file path corresponding to any given timestamp. Savors has built-in support for libpcap [13] and CSV data formats, but custom handlers can be easily added. Servers may be invoked by the client either locally, or remotely over SSH, which allows the client to be geographically separated from the data. Data is sent in batches and buffered at the client to maintain consistent display speed. Subsequent batches are transferred asynchronously while the current batch is being displayed.

The client is responsible for providing a visual (or aural) representation of the data along with playback controls and input fields to manipulate its behavior. All Savors visualization tools share the same basic interface. The bottom of each tool consists of buttons to play, pause, rewind, etc. the data stream as well as input fields to select/display the time period and other parameters. The left side of each tool shows the current mapping between colors and protocols. Protocols may be filtered from the display by click-

ing on the corresponding color box. The central region of each tool shows a specific visualization. All but the Savors Group tool of Section 4.1 have a set of panes below the central region to display details of any packet/flow over which the mouse pointer hovers (for the remainder of the paper, “flow” and “packet” will be used interchangeably).

Though not required, Savors was intended to be run on a small hyperwall-like device [16] with a dedicated display and back-end system for each tool. Tools can be run on the same system, but will not achieve maximum performance due to their greedy nature. Savors was written in Perl/Tk [14], which allowed for a very rapid and portable implementation. All tools were tested on a real 1.3 TB data set with over 10 billion flows captured during a 7.5 month period using three commercial network flow sensors. These sensors produce aggregated packet data in the form of network flows that contain information such as source/destination IP address/port, byte/packet count, etc. The vendor’s binary storage format is proprietary and undocumented, but a CSV export tool was available. Though non-optimal for performance, data was kept in compressed CSV form during prototyping for practicality. Note that all IP addresses from this data set displayed in the remainder of the paper have been anonymized by Crypto-PAn [19].

To support analysis over such large data sets, three of the visualization tools are designed to utilize high-end computing (HEC) resources such as clusters and supercomputers. Additionally, a response capability is integrated into all the visualization tools that allows users to instantaneously react to displayed data by blocking traffic, adding new filters, etc. The details of HEC usage, firewall configuration, and sensor updates as well as the underlying capabilities for locating data and controlling access to data and resources are hidden behind the Savors visual interface. This allows users to focus on their specific security tasks without the need to become experts in every aspect of their organization’s infrastructure. The following sections describe Savors in detail.

4 Monitoring

To assess the state of the network at any given time, Savors provides high, medium, and low level views of network flow data, which are described in the following sections.

4.1 Savors Group

High level monitoring is supported by the Savors Group tool, which displays an overview of network activity using a squarified treemap [2]. A treemap utilizes all available screen real estate while conveying hierarchical structure as well as relative size and content information. Savors Group uses a three level hierarchy of a set of sites that contain a set of hosts that perform a set of activities. The area of each site/host with respect to the total/site area represents the percentage of total/site bandwidth (in bytes or packets) that the site/host is utilizing. Totals are based on the hosts/sites displayed, thus accuracy may be improved by increasing the number of hosts/sites shown at once. Activities are represented by color, thus the dominant activity of all sites, one site, or one host is indicated by the most dominant color of the corresponding region at any given time.

Figure 1 shows sample output from Savors Group, which indicates that Site 2 is utilizing the most bandwidth, with HTTP being the most prevalent protocol. From a

security perspective, there are two regions of note. Site 2 has a host using significant bandwidth for BitTorrent traffic, which is often used for copyrighted content, thus indicating a possible policy violation of downloading or hosting inappropriate material. Site 1 has a host that is receiving both secure web traffic and some unknown traffic, which may be an indicator of a compromised host that is now accepting malicious commands.

4.2 Savors Flow

Medium level monitoring is supported by the Savors Flow tool, which displays host interactions using a parallel coordinate plot of the local and remote IP addresses and ports of each network flow, inspired by RUMINT's similar capability [3]. IP addresses are mapped along the vertical axes by dropping the decimal point and moduloing against the pixel height. Ports are mapped similarly. This scheme has the desirable property that hosts on the same subnet and numerically adjacent ports are visually adjacent.

Figure 2 shows sample output from Savors Flow. Natural nexuses form along the IP address axes where a single host serves many requests from many ports and along the port axes where the same port is used on different hosts for the same service. Port/host scans may be seen when the same remote host accesses many adjacent local ports/hosts. In the figure, two simultaneous host scans from different remote IP addresses may be seen along the top and middle of the local IP address axis (second of five from the left). These scans are still visible despite the considerable detail in the display.

4.3 Savors Content

Low level monitoring is supported by the Savors Content tool, which displays pixelized representations of the byte contents of each flow, inspired by RUMINT's similar capability [4]. To the right of each flow is a bandwidth bar, where the length of the bar is proportional to the logarithm of the byte/packet count, with red/green coloring for incoming/outgoing traffic. To the left of each flow is an area for visual flags that are displayed whenever a configurable set of regular expressions is matched in the content. When flow content is not transferred from the sensor to the correlation engine due to bandwidth constraints, the Savors Content server must be invoked on the sensor itself.

Figure 3 shows several features of Savors Content in action. A flag was generated when the content matched the regular expression "USER.*PASS". This indicates an unencrypted protocol such as FTP or POP that is exposing user names and passwords in the clear. Many such flags might indicate a brute force attack against the protocol. The black scan line a quarter of the way down is stopped at a flow that uses much higher bandwidth than those around it. Near the bottom of the pixelization region, there is a burst of HTTP traffic, which may indicate a reconnaissance operation against the organization. The bandwidth bars seem to indicate more inbound than outbound traffic for these requests, however, which does not support this hypothesis, but is itself suspicious for HTTP traffic and worthy of further investigation.

4.4 Savors Sound

A large part of intrusion detection involves analyzing historical events to identify sources of compromise, targets of attack, and the extent of any damage. Analysis demands full

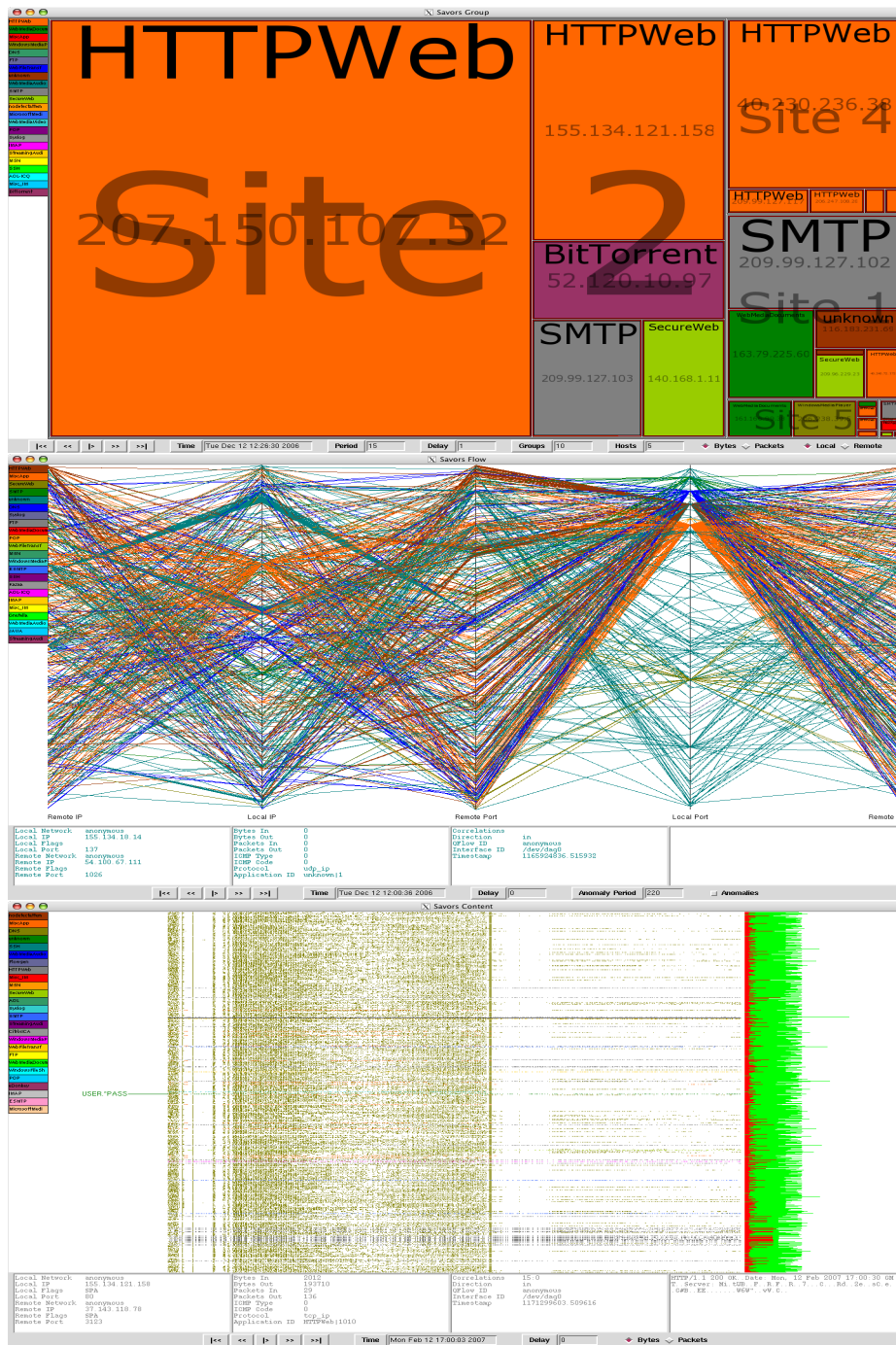


Fig. 1. (Top) Savors Group

Fig. 2. Savors Flow

Fig. 3. Savors Content

visual attention, thus impairing the ability to monitor current network activities. To improve monitoring capabilities during analysis and vice versa, Savors takes advantage of the human ability to process sound in parallel with sight. The Savors Sound tool auralizes both current and historical security data allowing the user to aurally monitor current data while visually analyzing historical data and vice versa. All Savors visualization tools can also sonically flag specific events, allowing them to focus attention when needed. All sounds are mixed dynamically by the Enlightened Sound Daemon [5], allowing multiple events from multiple clients to be auralized in parallel.

Savors Sound is based on Peep's three category model [8] of *events*, which are one-time occurrences, *heartbeats*, which are events that occur periodically over time, and *states*, which are the set of attributes that the system possesses at any given point in time. Events and heartbeats are defined based on configurable filters. Each match of a particular event filter triggers the playing of the associated sound. Heartbeat filter matches do not trigger sounds, but instead determine the frequency with which the associated sound should be played. States may be any quantifiable attribute, but only packets and bytes per time unit are currently implemented. Savors Sound can use the Peep sound library, which maps directly to the Peep model with sounds such as bird chirps for events, cricket chirps for heartbeats, and water flow for states.

5 Analysis

The Savors monitoring tools of the previous section are primarily concerned with conveying the state of the network through the display of raw and summarized flow information. As discussed, this information by itself is often enough to suggest systems that are compromised or under attack. The bulk of the displayed state, however, consists of normal activity that does not warrant further investigation. In addition, many indications of compromise and attack are received out-of-band from local system administrators or from individuals belonging to other organizations. Before an appropriate response can be initiated, it must be determined if, when, how, and from which host(s) a compromise has occurred and which other hosts may have been involved. Answering such questions accurately often demands the analysis of large volumes of historical data.

To support advanced analysis operations, Savors provides three HEC-assisted tools. The use of on-demand high-end computing has several advantages over a database model. The de facto standard format for packet trace storage, libpcap [13], and the emerging standard format for network flow storage, IPFIX [17], are both based on flat files. An on-demand HEC approach allows data to be analyzed in its native format exactly when needed, which simplifies transport and storage and frees up high-end resources for other purposes at other times. In contrast, a database solution requires dedicated resources in constant use just to insert continuously generated security data. All computations discussed in the following sections were performed on a 220 processor SGI Altix supercomputer with 2 GB of memory per processor.

5.1 Anomalous Savors Flow

As seen in Figure 2, visualizing all flows can result in a very cluttered display. Even though some details can be extracted, as discussed in Section 4.2, most flows repre-

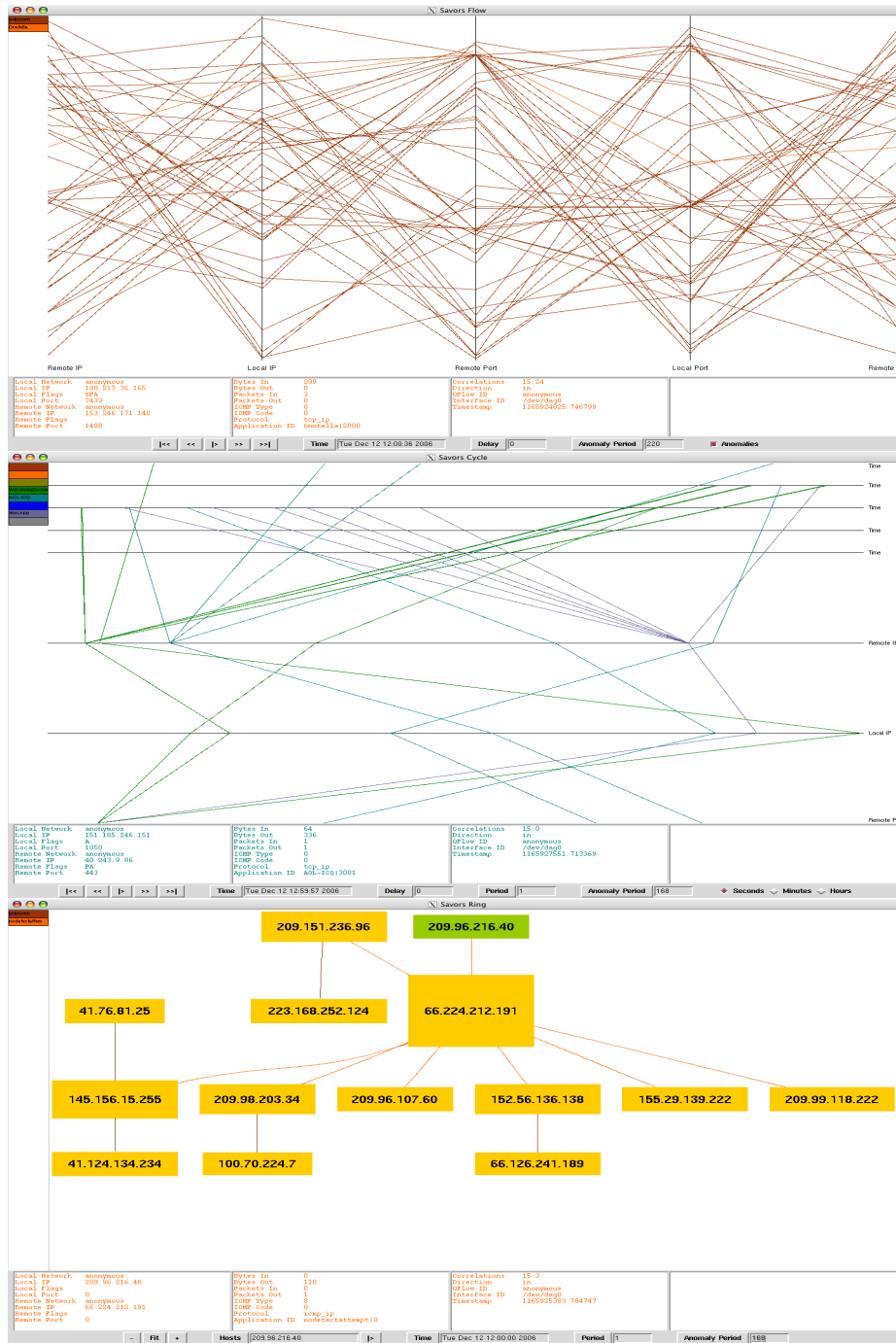


Fig. 4. (Top) Anomalous Saviors Flow Fig. 5. Saviors Cycle Fig. 6. Saviors Ring

sent normal activity, thus are not of particular interest. The flows deserving attention are those in which internal hosts are either accepting connections on ports they do not normally accept or initiating connections to ports to which they do not normally connect. To find such flows, Savors Flow supports anomaly detection, where flows in the display period are only shown if they are abnormal. Normalcy is derived using either an on-demand HEC computation or a cached instance of such a computation. Given an anomaly period P, an inbound flow not in P is normal if there is a flow in P with the same protocol and destination IP address/port. An outbound flow not in P is normal if there is a flow in P with the same protocol, source IP address, and destination port.

Figure 4 shows the anomalous flows over the same time period as Figure 2. As can be seen, the number of flows has been significantly reduced. This reduction is actually much greater than shown since in Figure 2, many flows have come and gone after the configured screen capacity has been reached, whereas in Figure 4, the display has not yet reached capacity, thus all flows are shown. The figure shows one Gnutella flow and several unknown flows. Of particular note is the inbound Gnutella flow, shown in the details pane, which should not be running within most organizations due to security vulnerabilities and misconfiguration that can lead to disclosure of sensitive information.

5.2 Savors Cycle

Organizations typically place significant restrictions on inbound access to protect internal resources, but have more permissive outbound policies to support access to Internet resources. Attackers use this fact by configuring compromised internal hosts to periodically connect outbound to retrieve further instructions. A human is often best-equipped to recognize such communication due to its widely varying and/or random connection patterns. The Savors Cycle tool supports such analysis by plotting periodic flows in a given time period. Many instances of periodic communication are legitimate such as queries to DNS servers or updates from time servers, thus to give meaningful results, Savors Cycle also supports anomaly detection in the same manner as Savors Flow. After the normal behavior profile is created, the computation counts the number of instances of anomalous connections and plots those past a configurable threshold.

Figure 5 shows sample output from Savors Cycle, with some protocols filtered for readability, which displays a parallel coordinate plot between the local IP address, the remote IP address and port, and the time of each network flow. Each pixel on the time axes may be configured to represent either a second, a minute, or an hour, where time increases with distance from the top of the display. Some connections only appear to have one instance as there is currently no minimum gap requirement, thus a single second may have many of the same connection.

5.3 Savors Ring

Compromised hosts typically become a launching point for additional attacks that take advantage of internal network access. Anomalous association with such hosts often indicates other sources or targets of attack. Given a set of starting hosts, the Savors Ring tool constructs a graph (rendered with GraphViz [7]) of all communication, either direct or indirect, involving those hosts for a given period of time. Because some hosts

are common rendezvous points such as DNS servers, which could result in a complete graph, only anomalous flows are considered, which are computed using an on-demand HEC computation or a cached instance. In this case, given an anomaly period P , a flow not in P is deemed normal if there is any flow in P involving the same two hosts.

Figure 6 shows sample output from Savors Ring. The green box (i.e. 209.96.216.40) represents the starting host. Boxes are sized according to the number of directly associated hosts, thus larger boxes represent hosts more likely to be either compromised or the source of compromise. Protocols deemed unlikely to be the medium for attack can be filtered using the color box on the left to break the graph into subgraphs with only the protocols of concern. In the figure, the initial line segment is a standard ping request, which, if filtered, completely segments the starting host from the rest of the graph, indicating no additional damage if the starting host were compromised.

6 Response

The ultimate goal of intrusion detection is not just to find attacks, but to respond to them. This includes verifying compromise, repairing damage, preventing further damage, and enhancing detection of similar attacks. Savors integrates response directly into all of its visualization components to allow the user to instantaneously react to displayed data. A right click on any flow-related visual element within the display brings up a list of actions that may be taken based on the local/remote hosts/ports of the given flow.

For local hosts, actions include downloading logs and opening a terminal, which can be used to verify compromise and repair damage, respectively. To perform these actions on a given system, the user must have appropriate login credentials. Since any network-accessible host within an organization may be subject to attack and require an appropriate response, users may need to juggle many different sets of credentials between large numbers of hosts. Different systems may also require different access paths through bastions, etc.

To support response capabilities in a scalable manner, Savors utilizes Mesh [11], which is a lightweight grid middleware that utilizes existing SSH infrastructure. Mesh provides single sign-on across distributed resources, allowing users to access any Mesh-enabled system with the same user identity and credentials. For organizations with different user domains and varying network access paths, Mesh provides automatic translation of user identities between domains and automatic location of the Mesh bastion responsible for each host. In addition, Mesh provides fine-grained authorizations that allow Savors users to be restricted to only the capabilities defined by the administrator. Not only does this allow each Savors action to be permitted or denied on a per host and/or per user basis, it also allows additional restrictions to be enforced such as only permitting actions from specific systems (e.g. the system(s) running Savors).

Besides downloading logs and opening terminals, Savors supports blocking of traffic to/from any subset of the local/remote hosts/ports of a flow to prevent further damage. For example, inbound traffic from any port of a malicious remote host can be blocked to any port of any local host to prevent additional attacks from the same host. Blocking is achieved through integration with Diaper [10], which is a framework for dynamically controlling the access control lists (ACLs) of network devices. Diaper sup-

ports firewalls, routers, and switches from major network vendors, allowing traffic to be blocked both at the perimeter as well as between internal hosts. ACL changes are requested through SSH remote commands, which are authenticated and authorized by Mesh. Diaper supports its own fine-grained authorizations that allow the administrator to define the scope of what may be blocked on a per user basis. Thus, Savors users can be given as much or as little control over network policies as desired.

Finally, to enhance detection of similar attacks, Savors allows users to define new IDS signatures and make them operational on existing IDS sensors (currently, only Snort sensors are supported). Savors constructs a signature template using the details of the selected flow, which may be edited before activation. A signature is activated by transferring it to the sensor and forcing a reload of the corresponding ruleset, both of which are authenticated and authorized by Mesh. This feature can be utilized to apply knowledge of attack profiles gained during analysis to tune the IDS.

Through the integration with Mesh and Diaper, Savors hides low-level operational details behind its visual interface and allows actions to scale across an entire organization with just a single set of login credentials. Users do not need to know how to access a particular host, which network device is responsible for blocking/monitoring which traffic, nor how to update the ACLs/signatures of that device. Once configured by the administrator, the appropriate actions are automatically taken behind the scenes by Mesh and Diaper, allowing users to focus on monitoring and analysis tasks.

7 Conclusions and Future Work

This paper has described a new tool for security event monitoring, analysis, and response called Savors, the **Scalable Aural-Visual Operations Room for Security**. Savors scales to real-world environments by utilizing four distinct areas of computing. Visualization-based monitoring components display high, medium, and low level representations of security event data in a form that conveys significant information in an easily understandable form. Visualization-based analysis tools utilize high-end computing resources on-demand to compile behavior profiles that point to anomalous behavior. Auralization allows both monitoring and analysis to be performed in parallel and draws attention to critical events in one tool when utilizing another. Remote data access and response capabilities across distributed resources are enabled using grid computing that provides a secure, single sign-on environment. Savors is portable across operating systems and was tested on a real data set containing over 1.3 TB of network flow data.

There are a number of directions for future work. Savors clients need an annotation capability to record interesting periods of activity. Servers could be modified to support multiple clients with a single shared view. A timeline like TNV's will be added to provide rapid time manipulation and a high-level summary of past activity. A drill down capability will be added to Savors Group to provide more detail about a specific site or host. A number of performance improvements are possible. The HEC analysis code must be optimized to eliminate file I/O bottlenecks by using shared memory capabilities. A minimum gap requirement must be added to the Savors Cycle analysis code to eliminate flows in close proximity. Finally, alternatives to SSH for streaming and Perl/Tk for rendering should be investigated to see if performance can be increased.

Acknowledgments

Thanks to Justin Cassidy and Matt Linton for supplying the flow data and suggesting the idea behind the Savors Ring tool. Thanks to Jeff Ward for suggesting the anomaly model for Savor Flow and the idea behind the Savors Cycle tool.

References

1. Abdullah, K., Lee, C., Conti, G., Copeland, J., Stasko, J.: IDS RainStorm: Visualizing IDS Alarms. IEEE Wkshp. on Visualization for Computer Security, Oct. 2005.
2. Bruls, M., Huizing, K., van Wijk, J.J.: Squarified Treemaps. 2nd Joint Eurographics - IEEE TCVG Symp. on Visualization, May 2000.
3. Conti, G., Abdullah, K.: Passive Visual Fingerprinting of Network Attack Tools. ACM Wkshp. on Visualization and Data Mining for Computer Security, Oct. 2004.
4. Conti, G., Grizzard, J., Ahamad, M., Owen, H.: Visual Exploration of Malicious Network Objects Using Semantic Zoom, Interactive Encoding and Dynamic Queries. IEEE Wkshp. on Visualization for Computer Security, Oct. 2005.
5. Enlightened Sound Daemon. <http://www.tux.org/~ricdude/Esound.html>.
6. Foresti, S., Agutter, J., Livnat, Y., Moon, S., Erbacher, R.: Visual Correlation of Network Alerts. IEEE Computer Graphics and Applications, vol. 26, no. 2, Mar. 2006.
7. Gansner, E.R., North, S.C.: An Open Graph Visualization System and Its Applications to Software Engineering. Software: Practice and Experience, vol. 30, no. 11, Aug. 2000.
8. Gilfix, M., Couch, A.: Peep (The Network Auralizer): Monitoring Your Network With Sound. 14th USENIX Large Installation System Administration Conf., Dec. 2000.
9. Goodall, J.R., Lutters, W.G., Rheingans, P., Komlodi, A.: Focusing on Context in Network Traffic Analysis. IEEE Computer Graphics and Applications, vol. 26, no. 2, Mar. 2006.
10. Kolano, P.Z.: Maintaining High Performance Communication Under Least Privilege Using Dynamic Perimeter Control. 12th European Symp. on Research in Computer Security, Sept. 2007.
11. Kolano, P.Z.: Mesh: Secure, Lightweight Grid Middleware Using Existing SSH Infrastructure. 12th ACM Symp. on Access Control Models and Technologies, Jun. 2007.
12. Lee, C., Trost, J., Gibbs, N., Beyah, R., Copeland, J.A.: Visual Firewall: Real-time Network Security Monitor. IEEE Wkshp. on Visualization for Computer Security, Oct. 2005.
13. Libpcap format. wiki.wireshark.org/Development/LibpcapFileFormat.
14. Lidie, S.: Perl and the Tk Extension. The Perl Journal, vol. 1, no. 1, Spr. 1996.
15. Papadopoulos, C., Kyriakakis, C., Sawchuk, A., He, X.: CyberSeer: 3D Audio-Visual Immersion for Network Security and Management. ACM Wkshp. on Visualization and Data Mining for Computer Security, Oct. 2004.
16. Sandstrom, T.A., Henze, C., Levit, C.: The Hyperwall. 1st Intl. Conf. on Coordinated and Multiple Views in Exploratory Visualization, Jul. 2003.
17. Trammell, B., Boschi, E., Mark, L., Zseby, T., Wagner, A.: An IPFIX-Based File Format. IETF Internet Draft, Jul. 2007.
18. Valeur, F., Vigna, G., Kruegel, C., Kemmerer, R.A.: A Comprehensive Approach to Intrusion Detection Alert Correlation. IEEE Trans. on Dependable and Secure Computing, vol. 1, no. 3, Jul. 2004.
19. Xu, J., Fan, J., Ammar, M.H., Moon, S.B.: Prefix-Preserving IP Address Anonymization: Measurement-based Security Evaluation and a New Cryptography-based Scheme. 10th IEEE Intl. Conf. on Network Protocols, Nov. 2002.
20. Yurcik, W.: Visualizing NetFlows for Security at Line Speed: The SIFT Tool Suite. 19th USENIX Large Installation System Administration Conf., Dec. 2005.