

# Improving Intrusion Analysis Effectiveness

Robert F. Erbacher and Karl Sobylak  
University at Albany - SUNY  
Department of Computer Science, LI 67A  
Albany, NY 12222  
\*erbacher@cs.albany.edu, ksobylak@hotmail.com

## Abstract

Analysis vs. analysis

The volume of data available to the analyst for the forensic analysis of an intrusion or other form of successful attack is enormous. Clearly, analyzing the textual data would be prohibitive as a networked environment will generate tens of thousands of log messages a day. In complex cases, where events must be correlated both temporally and spatially, the task is daunting. Many techniques are applicable to aid the analyst, including: data mining, machine learning, and visualization. Currently, no technique is the end all be all of forensic analysis. Consequently, this paper discusses our research towards the development of visualization techniques to aid the analysis process. These techniques are geared towards incorporation of all intrusion detection and analysis data, including both the original log data as well as the results of other intrusion detection and analysis tools. Incorporating all results into a single environment greatly increases the analyst's effectiveness. This will have the effect of reducing the lost time examining false positives, allowing identification of true anomalies, their sources, and their impact.

## 1 Introduction

One of the primary problems with the current forensic analysis process is the wasted time following false leads generated from the high false positive and false negative rates of intrusion detection tools. This results from the fact that most of the tools providing information to the analyst focus only on single events and single hosts. Typical log messages will generally flag individual suspicious events. Portsentry [20] will identify suspicious port connection requests and identify that host's activity within the system log file. Other intrusion detection and analysis tools behave similarly. This focus on single events is the limiting factor of most such tools. If Portsentry has identified a probe on one system then it will likely detect probes on all systems. However, since most systems within an organization are configured similarly, once an experienced hacker has identified the configuration of said machines the hacker may use this

information to break into other machines within the same organization other than the one probed. While snort [22] can collect all network traffic and can be applied to more complex scenarios, the filters associated with snort only apply to known attack signatures. Additionally, if the attack is sufficiently distributed temporally or spatially, most snort filters will fail. Thus, the typical information provided within system log files and snort filters in and of themselves are insufficient to efficiently analyze the network's activity and identify intrusion details.

Analyses of individual events is extremely limiting in the monitoring and analysis of activity as geared towards forensic intrusion analysis. It provides insufficient detail to separate out true attacks from necessary activity or casual exploration, leading to a high false positive rate. It's lack of correlation over a distributed environment or temporally over time leads to a high false negative rate as activity that should be flagged are missed. Attempting to perform the correlation manually from the textual log files as is often done when forensic analysis is incredibly time consuming and frustrating.

The key to improving the analyses effectiveness is the reduction of the high false positive and false negative rates through the correlation of activity both spatially and temporally. Thus, we must aid in directing the analyst's attention in the most promising directions and avoiding misdirections. To this end, we must accept that the starting point of this analyses process will be from the information provided within system log files, consisting of standard system log information as well as the results of intrusion detection tools. Consequently, system log files will contain thousands of messages for each day for even the least used systems. The number of messages that must be correlated for an entire compute environment over a substantial period of time is daunting. Given the size of such files, attempting to read through them and correlate the available information textually is a truly futile effort.

Various techniques can be applied to assist in the correlation activities. This includes machine learning, data mining, and visualization techniques. Each of these techniques has its advantages and disadvantages. For

example, machine learning techniques are subject to the training data, which will likely be insufficient given the adaptability of hackers. Data mining techniques tend to take substantial amounts of time to execute and won't provide the needed analysis of identified anomalies. Visualization techniques require continuous operator intervention. Given that the operator will be required to examine the results of any of the techniques at some point we have chosen to focus our efforts on visualization techniques. Visualization has the added benefit in that the results of any other technique, data mining or machine learning, can be incorporated as additional parameters.

Consequently, we have developed visualization and interaction techniques which when applied to the available data greatly aids in the spatial and temporal correlation and analysis of activity. More specifically, we have developed an integrated environment incorporating several visualization techniques and interaction techniques. Multiple visualization techniques are critical given the fact that no single display or view of the data will provide a complete understanding of all of the intrusion data. The resultant environment provides an exploratory data analysis environment geared towards allowing the analyses of the large volume of available data to locate anomalies and identify their source and meaning. This subsequently should greatly improve the analyst's effectiveness by directing the analyst's focus on the true attacks, separating out isolated anomalies.

## 2 Background

This work describes advancements to our prior research on the development of a visual intrusion monitoring and analyses environment [9, 10, 11]. This prior work was limited in that it was limited to a single monitored host, ignoring the correlation of distributed activity within a network environment, only provided a single visualization technique, and was extremely limited in its application of interaction capabilities. The current environment allows the exploratory data analysis of a large scale environment through the incorporation of multi-parametric data visualization techniques, extensive user-interaction capabilities, multiple linked views, and perceptually-based node positioning algorithms. The environment is capable of supporting both online monitoring as well as forensic analysis needs.

### 2.1 Multi-Parametric Data Visualization

The intrusion data we have been analyzing incorporated a large number of parameters, including: connection and disconnection time, type of connection, number of

connections, number of users, system load, IP Address, etc. Representing most if not all of these parameters simultaneously within a single display is crucial to fully understand and analyzing the data at hand.

Multi-parametric techniques [13], such as glyphs [3], provide a proven method for representing large data sets with large numbers of parameters. Other techniques have applied such visualization techniques to general aspects of network monitoring but not towards forensic analysis, leaving many open research issues.

Examples of such techniques include the representations of network performance and bandwidth usage [1, 15, 5, 16], even down to the router [5], individual packets [12], and individual e-mail messages [7]. The techniques developed for these purposes do not provide sufficient detail or handle sufficient numbers of nodes and attributes in combination for our needs.

The SeeNet environment, Becker et al. [2], provides linkmaps for visually representing the amount of data being sent between two network nodes. It is limited to identifying when a node is overloaded, showing the network's behavior, and showing how data moves between locations.

Livelihood [4] is an environment for visualizing and measuring the web, gathering statistics on the number of hits web sites are receiving. This statistical information is presented in visual form as charts, graphs, and geographically correlated glyph-based visualizations.

Netmap [6] is a generic visualization tool for the representation of relationships within a data set. The environment is principally geared towards showing known relationships of static data sets. Netmap is not geared towards exploratory data analysis.

### 2.2 Human Visual Perception

When developing visualization techniques, we generally attempt to gear those techniques to work effectively with the human visual system. In particular, we attempt to generate representations in which anomalies will be discerned through pre-attentive vision [14]. Essentially, this means that the human visual system will draw attention to characteristics of the display without the viewer's conscious thought or need to search through the display. Examples of pre-attentive characteristics include: line orientation, texture, size, and shape. Viewers do not need to consciously examine a display to notice changes in texture patterns or anomalous line orientations.

### 2.3 Exploratory Data Analysis

Our approach with the development of the given tools is to provide new exploratory data analysis tools [21, 18].

```

2 3 4 2001-04-22 08:40:52 SYS:  inetd[189]: ftp[28103] from 63.59.18.29 1378 169.226.1.101
/var/adm/messages System
2 3 4 2001-04-22 08:40:52 SUNY:      User  aa1111  logged  in  on  ftp28103  from
1Cust29.tnt1.glenn-falls.ny.da.u  169.226.1.101 /var/adm/wtmpx System
2 3 4 2001-04-22 08:41:58 SYS:  inetd[189]: telnet[28114] from 169.226.74.74 1204
169.226.1.101 /var/adm/messages System
2 3 4 2001-04-22 08:42:03 SUNY:      User  bb2222  logged  out  of  pts/20  from
h212-80.stuyvesant.albany.edu 169.226.1.101 /var/adm/wtmpx System
2 3 4 2001-04-22 08:42:49 MAIL:  sendmail[28134]: 169.226.1.101 /var/log/syslog      Mail
2 3 4 2001-04-22 08:42:49 MAIL:  sendmail[28136]: 169.226.1.101 /var/log/syslog      Mail
2 3 4 2001-04-22 08:43:09 USERS: 0 169.226.63.24 who -q Who
2 3 4 2001-04-22 08:43:09 LOAD: 0.03 0.03 0.02 169.226.63.24 uptime Uptime
2 3 4 2001-04-22 08:43:09 BDF: /proc 0 169.226.63.24 /usr/bin/df -k      BDF
2 3 4 2001-04-22 08:43:09 BDF: /dev/dsk/c0t0d0s0 52 169.226.63.24 /usr/bin/df -k      BDF

```

**Figure 1:** Sample postgres dump fragment.

The complexity and ever changing nature of intrusion related data ensures that no single technique will always be able to identify an intrusion and provide the details needed to analyze characteristics related to said intrusion, segregating true attacks from false positives within the morass of data. Exploratory data analysis is an approach to data analysis geared towards revealing the intrinsic nature of the data. Exploratory data analysis is particularly effective when it is not known apriori what is being sought within the data. With the intrusion data, known simple attacks can be identified quickly through known representational and interaction processes. This is seen with snort filters. Unknown attacks, as well as more complex attacks, and their methodologies can be identified through the application of provided visual and interaction capabilities, thus allowing the complete exploration of the data.

The exploratory data visualization techniques within our intrusion analyses environment are generated through the inclusion of multiple visualization techniques, each incorporating extensive multi-parameteric capabilities and user interaction facilities. The interaction is crucial for the exploration process, allowing the analyst to interact with the environment and linking the analysis performed within each visualization display, creating a single stream of analysis.

### 3 Data Collection and Access

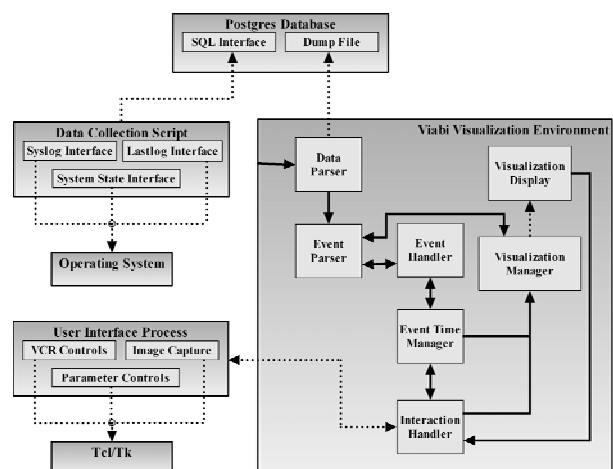
Our environment collects data through a set of bash scripts [8] which is then stored in a postgres database [17]. Our techniques are derived from the work on Hummer by Frincke et al. [19]. We are working to incorporate online retrieval of events as they are inserted into the database. However, the environment is currently geared towards forensic analysis of database dumps. Figure 1 shows an example dump fragment. The current database will incorporate details on connections, disconnections, daemon messages, system statistics, as well as Portsentry identified port probes. The results of any other intrusion detection tools or analysis tools,

such as data mining or machine learning tools, can easily be incorporated into the database format.

These data collection scripts will run on Solaris and Linux -based platforms. Porting the scripts to other UNIX platforms should be straight forward. Porting the scripts to MS Windows -based platforms remains an issue. The scripts support stunnel and ssh -based encryption and tunneling capabilities.

### 4 Environment Architecture

The high level architecture for the environment is shown in Figure 2. As can be seen from the diagram, the environment is made of several principal components. This includes: the postgres database, the data collection scripts, the visualization environment, and the environment user interface. The user interface is kept in a separate process from the visualization environment itself to ensure effective response time to user directed interaction even while the environment is executing and processing data. Synchronization between the processes is performed using semaphores and shared memory.



**Figure 2:** High-level architectural overview.

Many of the components are designed to be easily replaceable. For example, the data parser could be easily replaced to provide support for new database formats or

for online retrieval of data through direct SQL commands.

An important component to mention is the event time manager. As we are concerned with temporal relationships between events we must make an attempt to maintain a persistent representation of time. This is critical in our animated environment which is event-based. The importance occurs when there is a gap between events. If we do not maintain a persistent representation of time then we will merely display one event right after the next. If we do maintain a persistent representation of time we will need to include a representative delay in the visual simulation.

## 5 Visualization Techniques

The visualization techniques provide a visual representation of the events within the database for forensic analysis. Parameters associated with each event are represented visually as visual attributes. The key visualization capabilities include:

- A glyph-based animated representation
- Perceptually-based node layout algorithms
- A static histogram-based historical overview

Associated with the visualization techniques are extensive interaction capabilities. These interactive capabilities are critical for the exploratory capabilities of the environment.

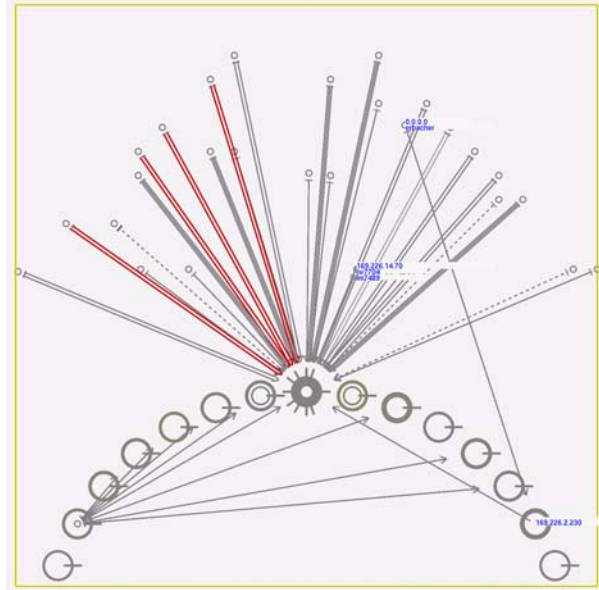
### 5.1 Glyph-based Techniques

The visualization techniques we have developed include an animated glyph-based environment which will show the current state of the compute environment at any particular point in time, figure 3. The environment incorporates many parameters as visual attributes, as described below.

A border surrounding the entire display window provides a visual representation of the time of day. The border is black at midnight and white at noon. An additional yellow border is indicative of PM vs. AM, a necessity given the difficulty of determining if intensity is increasing or decreasing when changing in small increments.

The monitored system has additional information attached to its glyph. Each spoke is representative of ten users and the thickness of the inner circle is representative of system load. Connecting nodes have cross hashes representing the number of different users connecting from that node, number of individual hashes, and the number of connections by that user, thickness of the hash.

The directed lines are themselves glyphs, showing the direction of the connection, the state of the connection, and the type of connection. Two parallel lines is indicative of an unauthenticated connection. If the parallel lines are red then the authentication has failed. Solid lines are telnet connections, long dashed lines are privileged FTP connections, and short dashed lines are anonymous FTP connections. Lines with multiple arrows are indicative of NFS connections. A lost NFS connection is represented by highlighting the node in yellow. Thick red lines are portsentry identified attacks.



**Figure 3:** User interaction, selection, and feedback example with multi-node monitoring. IP Address is shown above user list when available. Identifiers are in blue to differentiate from other characteristics.

### 5.2 Node Positioning Algorithms

The layout has been designed to effectively show remote vs. local connections. In fact, the layout is also designed to highlight connections to workstations in contrast to servers. In figure 3, there is a single server that individuals are expected to connect to and perform most of their work on. This system is at the apex of the bottom semi-circle. The remaining systems are UNIX workstations in a public access lab. Generally, you would not expect the workstations to be accessed remotely at all. However, the figure shows several anomalies in this respect.

The node layout is designed to maximize the viewability of interconnections and their relationships. For example, most of the remote nodes (top of the window) are connecting to a single local server (bottom of the window). However, one remote node has connected to a

local workstation directly, rather than connecting to the server. Perceptually, the eye is drawn to this line due to its deviation in angles and the generated line intersections. While other line intersections exist, such as the cross hatches for number of users per interconnection, the deviant line has a much stronger presence due to its size and overwhelms the lesser intersections.

The node layout algorithm is based on modifications of force-based graph layout algorithms. In a sense we are considering there to be two gravity wells, one at the top of the window and one at the bottom. Local nodes are attracted to the bottom well using circular orbits. Once the first ring is filled a new ring is created.

The remote nodes are placed in the top ring. Initial ring selection is based on the difference between the remote IP Address and the local IP Address. The least different addresses are placed in the ring nearest the local nodes. In this way, we provide a sense of locality with respect to hosts.

### 5.3 Histogram-Based Techniques

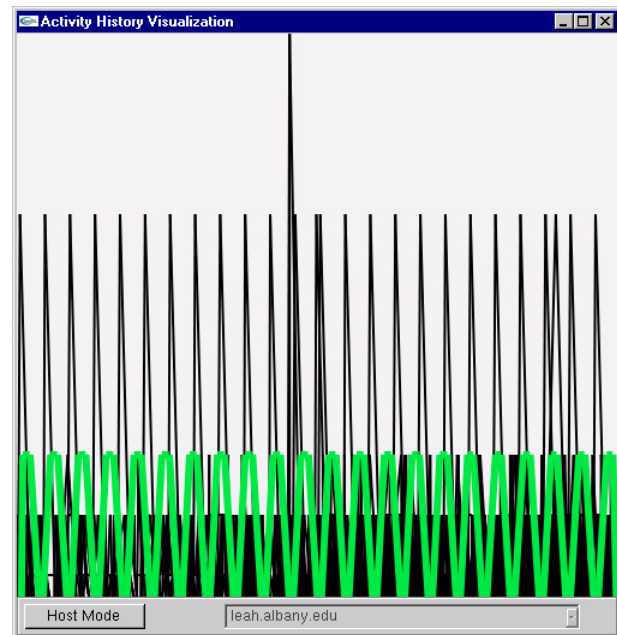
An additional display incorporates a histogram-based display in which the activity of a remote host is measured over time on a scale of 1-10. A value of zero essentially represents no activity. An example such mapping is shown in figure 4. Each event appearing in the system log files is given a score in this range, mapped to the vertical axis. The horizontal axis represents time. Essentially, this display provides a representation of the systems “activity signature”. Within this activity signature can be identified potential intrusion signatures. This view does not provide all of the details shown in the animated view and thus cannot resolve the questions as to what is occurring, why, and most importantly, is it acceptable? A second view of the data is selectable with a push button toggle at the left of the interface bar, appearing on the bottom of the display. This essentially switches between showing a histogram view based on hosts vs. a histogram based on users. The user view of the histogram is limited as it obviously will not show activity by unknown users, such as portscentry identified port scans.

The lack of detail within the histogram view limits its applicability. However, it provides a historical view not available in the animated view. In combination, the two techniques provide a complete analysis capability. The integration of the two techniques requires effective interactive capabilities in order to correlate between the displays. Additionally, we are continuing to analyze and refine our table of values.

| SysLog Identifier | Numerical Value |
|-------------------|-----------------|
| ALERT             | 9               |
| ANONYMOUS         | 5               |
| INETDFTP          | 2               |
| INETDTELNET       | 2               |
| LOGININCORRECT    | 8               |
| PORTMAP           | 7               |
| PORTSENTRY        | 10              |
| PRIVILEGED        | 2               |
| SUDO              | 10              |
| TELNET            | 2               |

**Figure 4:** Sample of syslog identifier to numerical value mapping. One is the lowest value or of least concern. Ten is highest. This is only a subset of the full table.

Each event is represented by a corresponding spike in the histogram. For items such as telnet, separate disconnections messages and consequently spikes are shown. If there is no activity for a period of five minutes then the activity level is reset to zero.



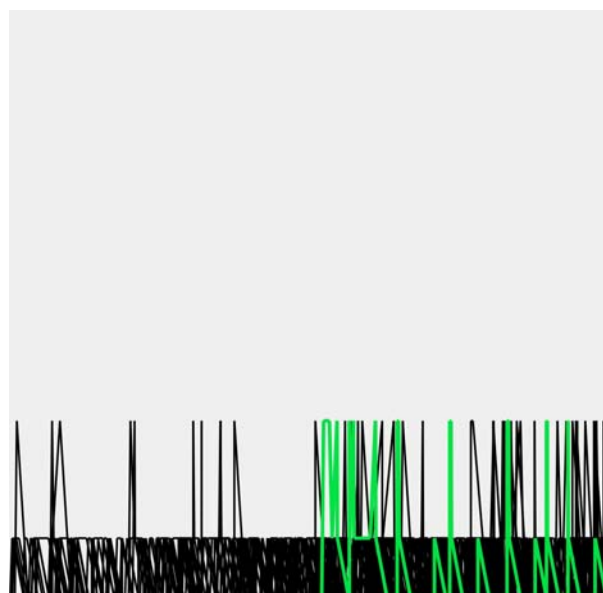
**Figure 5:** Regular repeated host activity selected. User interface for the histogram display is also shown. The push button on the bottom left switches between host/user mode, displaying the current mode. List box to the right shows the currently selected hostname. The list box also allows selection between multiple hosts when needed.

An example of the actual representation of the host-based display is shown in figure 5. A host has already been selected. The hostname is shown in the drop down list on the right side of the interface bar. A

host is selected by clicking on the appropriate line. Should multiple lines be selected, quite an easy occurrence, then the drop down list which shows the selected hostname will contain a list of all associated hostnames, allowing the analyst to quickly switch between hosts. The display will update correspondingly, highlighting the activity of the newly selected host.

An example of the user-based display, is shown in figure 6. As can be expected, known users normally operate within expected boundaries. It is the unknown users that cause most of the problems. From the forensic analysis point of view however, this display is still critical after an intrusion to follow the activity of the user.

In the example in figure 6 a user has been selected which shows pretty much normal activity. This consists of essentially random activity and just as importantly random start times. In other words it is not continuous. The activity consists mainly of telnets, disconnections, and mails.



**Figure 6:** User-based histogram example. The activity signature of a selected user is shown.

## 6 Interaction Techniques

As has been discussed, the interaction capabilities are as important to our needs as the visualization techniques themselves. The interaction techniques fall into four categories, including:

- Capabilities to enhance the review and analysis of data
- Probing capabilities for review of data specifics
- Selection and highlighting facilities to identify and track activity

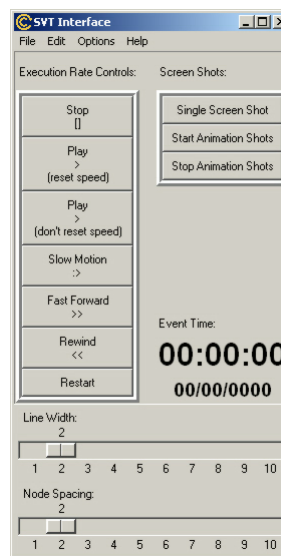
- Multiple-linked views to correlate identified activity between displays (visualizations).

Additional capabilities are provided within the user interface window, including:

- Screen capture facilities
- Date and time of the currently executing event
- Controls for the thickness of lines and spacing of nodes. Allowing density to be increased or decreased.

### 6.1 Activity Review and Analysis

As our principal activity is the analysis of data available for forensic analysis, we must provide effective capabilities examining desired periods of history. There are two main mechanisms provided. First, are VCR like controls, Figure 7, which allow the analyst to control the execution rate, pause the event execution, or rewind the event execution.



**Figure 7:** Example of the user interface window and its associated capabilities.

The second mechanism is provided in the histogram window. As discussed, the horizontal axis represents time. By right clicking at any point in time within this display a menu is popped up providing the analyst with the option “Jump to”. This will allow the analyst to quickly jump to a particular point in time. Since events must still be evaluated by the environment to determine the current state at the destination time the jumping cannot be instantaneous.

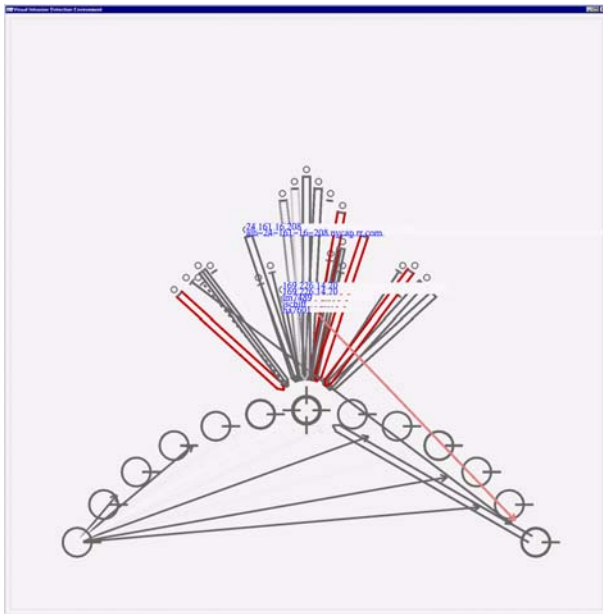
With the combination of these two capabilities, once an initial event or period of time has been identified that requires additional attention the analyst can quickly jump to that time period and review it repeatedly gaining greater insight into the activity. This process can then be repeated .

## 6.2 Probing

The visualization environment creates abstract representations of the data within the system log files. During the analysis process it is generally necessary to retrieve specifics to identify for instance exactly what host has been performing the identified questionable activity. By clicking on a host with the environment paused the following information will be displayed on top of the host:

- Host IP Address
- Hostname (if available)
- Username list. All users will be listed one after another.

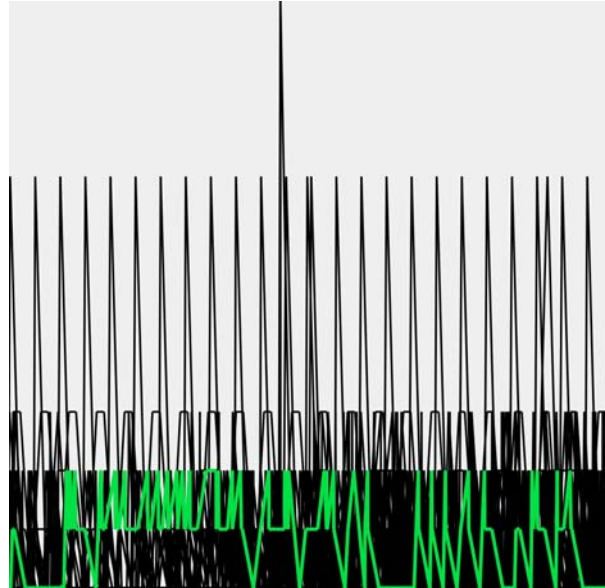
Multiple hosts may be selected simultaneously.



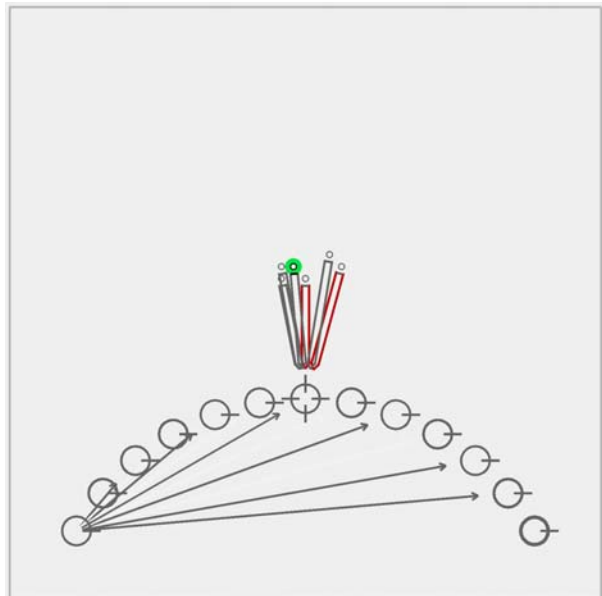
**Figure 8:** More complete example incorporating multiple monitored nodes, multiple connections to local nodes, local inetd, etc. Probing is modified to show hostname as well as IP address and all connected users.

## 6.3 Selection and Highlighting

In order to allow the analyst to track a host's or user's activity more clearly once it has been identified as having been involved in suspicious activity, we allow the host to be selected. This is done merely by clicking on the host. Once a host is selected it will remain selected so that the analyst can review activity over various periods of time while maintaining the selection.



**Figure 9:** Host-based histogram example. The activity signature of a selected node is shown.



**Figure 10:** Node selection example. When host is selected in either visualization display, it is highlighted in both.

To improve the effectiveness of this techniques we also incorporate the concept of multiple linked views [19]. With this mechanism, interactions in one display impact the other. For example, when we select a node in the animated display, the associated activity histogram is highlighted in the histogram display. Conversely, when the analyst clicks in the histogram display, the animated

display will immediately jump to the corresponding point in time. If a particular host can be identified as having been selected within the histogram display then we will highlight that node in the animated display and its associated activity display.

Figure 9 shows an example of a selected host's activity within the histogram based visualization. The corresponding selection in the animated display is shown in figure 10.

## 7 Analysis Methodology

Our goal with the environment is to improve the effectiveness and timeliness of the analyst. The animated representation can run through a weeks worth of messages from a networked infrastructure of 12 machines, including a server, in under half an hour. The test file consisted of about 200,000 messages. Clearly full analysis will take more time, though far less total time than that of analysis of the textual log files directly.

The forensic process with the environment is essentially typical of any forensic analysis process. The goal is to provide tools which makes the process more efficient and hopefully allow anomalies to be identified that otherwise would not have. With forensic analysis the goal is to identify how and when an intrusion occurred. The analyst can start with anomalies, quickly identify what information is available in correlation with the anomaly, and from there identify the activities that actually led to successful intrusions. As discussed in our prior work, much of this task relates to behavior analyses [11], typical of forensics, as we must identify why an individual may have been performing a set of tasks to determine if a substantive thread is being followed or not.

While this paper discusses the application of our environment to forensic analysis, the environment is also applicable to online monitoring. This is particularly true with respect to the animated environment. One goal with the development of this visualization display is that the principal characteristics be visible even when using only a very small window. The goal being to allow a system administrator to maintain the visualization active upon their desktop along with their other typical tools. As soon as an anomaly is identified a more detailed extensive analysis can be performed.

## 8 Analysis Examples

Most types of attacks and their associated anomalies are identifiable within the visualization environment through clear visual representations. The interactions and additional visualization capabilities allow these anomalies to be fully explored within the visualization

environment, eliminating most of the need for the textual log files.

### 8.1 Multi-Local Accesses

Figure 3 shows an example image in which a single local workstation is maintaining a large number of connections to other local workstations. Since each workstation is essentially identical this is unusual activity as a single workstation is generally sufficient. While individuals will often connect from a workstation to the server, connections among workstations is unusual and often undesirable as it may impact the individuals using the other workstations.

The animated display shows that all of these connections are not persistent. In fact, figure 8, which shows another time point of this data set, shows that several of the connections have been disconnected. Disconnected connections are faded out to aid with temporal persistence and to handle instantaneous events, as with radar screens. The histogram display is much more limited in this scenario as it will show the many connections and disconnections but is limited in the meaning it provides to them.

At this point, in order to identify the meaning of the identified activity we can look into prior activity to determine if there was prior unusual activity leading to this individuals initial connection. We can also examine the user's activity on the destination machines to gain insight into the user's ultimate focus. The meaning in this scenario will relate to who the user is and what activity they were performing on the other local workstations.

### 8.2 Local-System Connect Through

Figure 3 shows another example of an anomaly, namely that of a local workstation which has been connected through. A user has evidently connected initially to one local workstation and then quickly connected to the local server. Why not connect to the server directly?

This scenario actually matches the behavior of typical hackers who, after breaking into one system on the network, will immediately attempt to see what other systems they either have access to or can break into with this system as a launching pad. Is this scenario a true attack or just unusual behavior by an individual?

An important issue to discuss in this anomaly is the clarity of the visual presentation. First, a direct connection to a local workstation creates a deviant line and line intersections which attract the user's focus. Second, the characteristic 'V' shape is clearly defined and stands out. The perceptual qualities of the node layout algorithm help ensure these types of anomalies are quickly discerned by the analyst. Notice the

differences between this image and that shown in figure 8. While, we have a similar ‘V’ shape figure 8 clearly shows a different scenario. In this scenario, portsentry has identified a remote host as having probed the system. Meanwhile, a user on that local system is connecting to the monitored server.

### 8.3 Temporally Related Events

In [9] we show an example attack in which we can identify four unique stages of the attacker progresses:

1. An initial port probe of the target system
2. Inetd connections to the systems without authentication
3. Inetd connections with successful authentication
4. FTP access to the target host

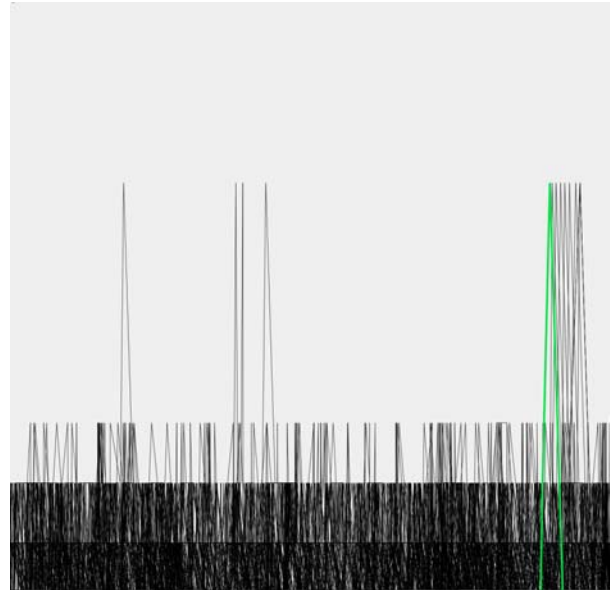
This attack can be identified in the animated environment through the sequence of activities occurring sequentially. The selection and highlighting features essentially make this type of activity much easier to follow as it is no longer possible to lose track of a system the analyst desires to monitor.

Additionally, this type of attack can be identified within the histogram-based environment as a graph with decreasing levels. In fact, any graph starting at level 10 and ending with level 2, indicating a successful connection, should cause concern.

### 8.4 Mult-Access Correlation

The example in figure 12 shows an interesting set of peaks, each of the peaks, equal to the highlighted link, is set to level 7. These happen to be repeated login failure messages which occur when a user fails to enter a correct password 5 times. In an academic environment, this is not uncommon, which would likely explain the isolated peaks to the left side of the visualization. The anomaly on the right side of the screen, however, is far more insidious. The high density of messages indicates either a concerted effort at intrusion or some other event which happened to cause synchronized failures. The next step is of course to identify whether there is any reason for such a synchronized attack. Since this did not occur near the beginning of the semester it is not a group of new students having difficulty with their password. Looking at the hostnames of the individual machines which originated the connection attempts indicates they did come from a public access lab. They also came from different machines. Additionally, we know that the access came in a sequence, not simultaneously, and within a fairly short amount of time, each message being logged a few minutes after one another. If we try to think like a forensic analyst and analyze the behavior of the activity we can pick out an additional clue: the first several messages occurred from

the same machine the remainder occurred from different machines. This would appear to be a student that forgot their password, attempted multiple times on one machine, and finally attempted several other machines to determine if the results were different.



**Figure 12:** Portmap attack example. Multiple isolated events are shown on the left side of the histogram. Coordinated attack is shown on the right side of the histogram

All of these details are available within the visualization environment. Identifying the unusual sequence of activity would have been very difficult with only the textual information. Clicking on each peak would quickly show the hostname of each offending machine and also verify there are no overlapping events. Finally, the temporal closeness of the events can be identified in two different ways. first through the animated display, which would also have highlighted the anomalous activity. Second, since the histogram for this database happens to only be representing a five hour period of time, the close spacing clearly indicates very short period of time delay.

## 9 Conclusions

We are developing techniques and capabilities within an integrated environment that can assist in the analysis and investigation of intrusions within a distributed computing environment. All available data is applied to greatly improve the analysis ability, aiding the analyst in examining and analyzing the data. Our techniques focus on the application of perceptually-based visualization

techniques in conjunction with interactive capabilities to deploy exploratory data analysis capabilities.

These capabilities will improve an analyst's efficiency and effectiveness, greatly reducing the impact of false positives and false negatives. Given the size of the log files which are the principal method of such analysis ISP's are only retaining these files for relatively short periods of time, often ranging only 30 to 90 days. This is a very short period of time for law enforcement, especially given the number of cases in which they are involved.

We also provided detailed examples as to how the environment can be used in different situations to aid the analyst's task. These examples provide only a sampling of how the environment can be used

## 10 Future Work

The techniques we are developing and integrating within our environment already provide extensive capabilities. We must continue to expand the visualization and interaction techniques. Improving on the histogram-based display is one of our most critical tasks. Additionally, we must look at integrating network-based data as this will provide a substantial amount of additional data not otherwise available, including handshake data. The network data in and of itself would not eliminate the need for host-based data. For example, the multi-access correlation example would not have been identifiable with only network-based data.

Additionally, while our visualization environment will run on any environment we have not incorporated Microsoft Windows specific events or event formats. As Microsoft Windows platforms make up the majority of the systems deployed, this is a substantial limitation.

## 11 References

1. Richard Becker, Stephen Eick, and Allan Wilks. "Graphical methods to analyze network data," In *IEEE International Conference on Communications ICC '93 Proceedings*, Geneva, Switzerland, pp. 946-95, May 1993.
2. Richard Becker, Stephen Eick, and Allan Wilks, "Visualizing Network Data," *Readings in Information Visualization: Using Vision To Think*, Stuard Card, Jock D. Mackinlay, and Ben Shneiderman, editors, Morgan Kaufman Publishers, pp. 215-227, 1999.
3. Berton, J.A., Jr., "Strategies for scientific visualization: analysis and comparison of current techniques," *Proceedings of Extracting Meaning*

4. Tim Bray, "Measuring the Web," *Readings in Information Visualization: Using Vision To Think*, Stuard Card, Jock D. Mackinlay, and Ben Shneiderman, editors, Morgan Kaufman Publishers, pp. 469-492, 1999.
5. Kenneth Cox, Stephen Eick, and Taosong He, "3D geographic network displays," *ACM Sigmod Record*, Vol. 25, No. 4, pp. 50, December 1996.
6. C. Davidson, "What Your Database Hides Away," *New Scientist*, Jan. 9, 1993, pp. 28-31.
7. Stephen G. Eick and Graham J. Wills, "Navigating Large Networks with Hierarchies," In *Visualization '93 Conference Proceedings*, San Jose, California, pp. 204-210, October 1993.
8. Robert F. Erbacher and Bill Augustine, "Intrusion Detection Data: Collection and Analysis," *Proceedings of the 2002 International Conference on Security and Management (SAM '02)*, Las Vegas, NV, June 2002, pp. 3-9.
9. Robert F. Erbacher, Kenneth L. Walker, and Deborah A. Frincke, "Intrusion and Misuse Detection in Large-Scale Systems," *Computer Graphics and Applications*, Vol. 22, No. 1, January/February 2002, pp. 38-48.
10. Robert F. Erbacher, "A Component-Based Event-Driven Interactive Visualization Software Architecture," *Proceedings of the 2002 International Symposium on Information Systems and Engineering (ISE'2002)*, San Diego, CA, July 2002, pp. 237-243.
11. Robert F. Erbacher, "Visual Behavior Characterization for Intrusion Detection in Large Scale Systems," *Proceedings of the IASTED International Conference On Visualization, Imaging, and Image Processing*, Marbella, Spain, September 3 - 5, 2001, pp. 54-59.
12. Deborah Estrin, Mark Handley, John Heidemann, Steven McCanne, Ya Xu, and Haobo Yu, "Network Visualization with Nam, the VINT Network Animator," *IEEE Computer*, Vol. 33, No. 11, pp. 63-68, November 2000.
13. Grinstein, G.G.; Levkowitz, H.; Pickett, R.M.; Smith, S. (1993) "Visualization alternatives: non-pixel based images," *Proc. of IS&T 46th Annual Conf.*, pp. 132-133.

14. Markus Gross, *Visual Computing, The Integration of Computer Graphics, Visual Perception and Imaging*, Springer-Verlag, 1994.
15. Taosong He and Stephen G. Eick, "Constructing Interactive Visual Network Interfaces," *Bell Labs Technical Journal*, Vol. 3, No. 2, pp. 47-57, April-June 1998.
16. Eleftherios E. Koutsofios, Stephen C. North, Russel Truscott, and Daniel A. Keim, "Visualizing Large-Scale Telecommunication Networks and Services," *Proceedings of the IEEE Visualization '97 Conference*, IEEE Computer Society Press, San Francisco, CA, pp. 457-461, 1999.
17. Bruce Momjian, *PostgreSQL: Introduction and Concepts*, Addison-Wesley, 2000.
18. *NIST/SEMATECH e-Handbook of Statistical Methods*, <http://www.itl.nist.gov/div898/handbook/>, September 2002.
19. Polla, D., J. McConnell, T. Johnson, J. Marconi, D. Tobin, and D. Frincke, "A FrameWork for Cooperative Intrusion Detection," *21st National Information Systems Security Conference*, pp. 361-373, October 1998.
19. Jonathan C. Roberts, Multiple-View and Multiform Visualization," *Proceedings of Visual Data Exploration and Analysis VII*, January 2000, pp. 176-185.
20. Joel Scambray, Stuart McClure, and George Kurtz, *Hacking Exposed: Network Security Secrets and Solutions*, 2nd Edition, Osborne/McGraw Hill, 2000.
21. John W. Tukey, *Exploratory Data Analysis*, Addison-Wesley, 1977.
22. <http://www.snort.org>