

Visualization in Detection of Intrusions and Misuse in Large Scale Networks

Robert F. Erbacher

*Department of Computer Science, LI 67A
University at Albany - SUNY, Albany, NY 12222
erbacher@cs.albany.edu*

Deborah Frincke

*Department of Computer Science
University of Idaho, Moscow, ID 83844-1010
frincke@cs.uidaho.edu*

Abstract

The Internet is quickly becoming entrenched in the communication and commercial sectors of everyday life. With this movement away from traditional fixed infrastructure we are also moving away from the traditional securities placed within fixed infrastructure. This has led to increasing numbers of attacks designed to infiltrate or disrupt the activities being performed by companies and individuals on the Internet. We are exploring the applicability of visualization techniques in conjunction with a well-known intrusion detection system (Hummer) for the detection and analysis of misuse of computer systems connected to the Internet. The visualization techniques will allow users to identify the behavior of users connecting to the system and identify those whose intentions are unwelcome.

1. Introduction

This paper examines visualization techniques in the context of protecting large-scale networks. Examination of both commercial and research efforts to identify security violations consistently results in the observation that considerable quantities of data are generated—usually considered to be far too much to be evaluated effectively using current techniques [1, 2]. Some of this is due to the way that data-gathering choices are made [1]. It is our belief that refinements in the data gathering decision making process will not suffice: as networks grow larger, the amount of misuse-relevant data will also grow. Hence, better methods for analyzing the data are needed, rather than continued reliance on primarily textual techniques.

The use of text is limiting since reading textual information is inherently a perceptually serial process. Interpretation of graphical images, on the other hand, is perceptually a parallel process [3, 4]. Forcing the user to use textual information therefore slows the analysis process substantially in comparison to the use of graphical imagery. An additional advantage of imagery is that more “concepts” can be presented in a single image.

Thus, rather than observing individual reports or report summaries, it is possible to observe a single image that embodies the same information. This will reduce the amount of mental context switching required by users, making system assessment both easier and more efficient.

Large-scale system security is a critical national problem. A 1996 CSI-FBI survey found that \$4.5 billion was lost to business due to compromises in information security, and that the majority of businesses experienced some form of intrusion during the year. Systems of all kinds are vulnerable, as evinced by the 1998 attacks on Pentagon computers and the 2000 attacks on e-commerce. It is clear that even if systems can be made more secure, attack and internal misuse of technologies will evolve, making some form of intrusion/misuse management a necessity for critical systems.

This research focuses on the development of techniques and tools for the visualization of network data with the goal of aiding users in identifying intrusions at a point early enough in the intrusion attempt that the user has not caused damage to the system. This rapid analysis and continuous monitoring are not possible with textual log information due to the time required to perform the analysis. This project aims to improve the state of the art in this area. Our goal in the long term is to provide an extent of automatic detection of attempted intrusions as well as techniques such that our target audience, e.g., security experts, will be able to quickly identify any new types of attacks that may occur. In terms of visualization, our goal is to ultimately provide general capabilities for the representation of network information that can automatically be provided for any set of networking characteristics that might be required or useful.

2. Previous work

2.1. Intrusion detection systems

Little previous work has been done towards the use of visual analysis as an aid to intrusion detection. For instance, many have proposed use of a simple “odometer-like” or metered scale to indicate the

estimated level of attack a system is enduring. This is embodied in the Hummer “perceived level of threat” [5] indicator. Earlier systems, such as DIDS [6, 7], provided graphical representations in the form of color to indicate when a system had experienced a sequence of suspicious events. While useful, these approaches do not provide adequate detail to do more than observe that attacks are in progress and do little to aid diagnosis. Dr. Frincke has performed preliminary investigations towards identifying likely models for depicting system state [8]. This expands on lessons learned from that system and its prototype.

2.2. Visualization systems

In contrast to intrusion detection, quite a bit of visualization research has been applied to network accesses. The principal body of work related to network intrusion is from the information exploration shoot-out, organized by Georges G. Grinstein and supported by NIST [9]. In this project, researchers were given access to a dataset consisting of network intrusions. The idea was to identify which researcher’s techniques were effective at identifying the intrusions. The driving philosophy was that little work has been done to compare visualization techniques in a formal setting. Perceptual studies have been done to identify characteristics of the human visual system [3, 4] that should be used as a basis for the development of visualization techniques but little has been done to actually compare and contrast visualization techniques. There is no body of literature that identifies what visualization techniques definitively work better on a given data set.

Most previous work involving visualization related to networks has emphasized graphics that depict network performance and bandwidth usage [10, 11, 12], even down to the router level [12]. The techniques developed for these purposes do not provide sufficient detail for our needs. Other work has been geared towards visualizing systems for program analysis and program development [13]. These environments typically deal with small numbers of processors that are working on a single task and thus have a common grounding.

This research into network usage has not been applied to network intrusions. It does, however, provide a starting point. Becker et al. [14] discuss the SeeNet environment that provides linkmaps for visually representing the amount of data being sent between two network nodes. It can identify when a node is overloaded, shows the network’s behavior, how data moves from one location to another and its volume. This is important when a crisis occurs and usage increases dramatically, e.g., after a California earthquake. Understanding the consequences of events, such that

telephone companies can be prepared for changing demands, is imperative.

Livelihood [15] is an environment for visualizing and measuring the web. By probing web accesses they gather statistics as to the number of hits web sites are receiving. This information is presented in statistical form as charts and graphs. An extension to the environment provides a more graphical representation. In this advanced form, approximate location of network nodes is represented, showing geographical association between web sites. The visual representation of each node is then presented in such a way as to reveal the activity of the site. Each node can represent several parameters simultaneously.

3. Environment description

Our experimental process for this project was to develop a complete visual interface between an intrusion detection system (IDS) and our target user base (system or security administrators, investigators, students, etc.). The IDS we have selected is the Hummer system. This IDS is a highly configurable collaborative system, arranged both hierarchically and in a distributed way, permitting data to be gathered from widely distributed sites. This makes it particularly appropriate for use in investigating large-scale intrusions. Hummer’s design allows filtering of data sharing at any level (between hosts, networks, or enclaves of networks) and it is thus useful for adjusting the data-gathering activities with regard to the visualization. Hummer currently has a sophisticated graphical interface (although it does not provide visualization). Finally, Hummer is designed deliberately to make it easy to interface with additional IDS and additional interfaces.

Our initial approach was to perform post-mortem analysis of network and system activity collected by the Hummer system. We are extending this to include real-time capabilities. Real-time analysis is necessary to analyze the current state of the network and interrupt potential attacks. Post-mortem analysis is essential for inspection of network activity of past unmonitored or lightly scrutinized periods, e.g., overnight and weekend activity. For very large-scale networks, data transport makes it likely that attacks may progress significantly before information about them can reach an analysis site.

We have identified patterns visible in our current environment that are indicative of potential intrusions. As we develop a more complete environment we must identify characteristic patterns and the activities they are associated with. This will provide a basis for observing the network. We must also examine many actual intrusions and ensure they can be detected and what characteristics correspond with them. Ultimately, we

need to provide users with detailed descriptions of how to use the environment and examples of what to look for when analyzing network behavior.

3.1. The hummer intrusion detection system

The HMMR protocol has been designed to address the requirements needed to permit network sites to share security relevant data while still retaining local control of data gathering activities (figure 1), deciding locally how much trust to place in data received from outside sites, and determining how much data to share with outside sites. This data is stored in a textual format, figure 2.

Hummingbird is formulated as a distributed system for managing misuse data. Hummingbird employs a set of Hummer agents, each assigned to a single host or host set. Each Hummer interacts with others in the system through manager, subordinate, and peer relationships. Managers may transmit commands to subordinates; such commands include requirements to gather/stop gathering data, to forward/stop forwarding data, and the like. Peers may send requests for data forwarding/gathering/receiving to peers; the peer decides whether to honor such requests. Subordinates may send requests to their managers as well. Kerberos is used for authentication and encryption of communication between Hummers and for authentication between Hummers and their database. Hummer agents share data based on their own locally controlled policies regarding trust, information flow, and cooperation. Simple filtering is also performed by each host.

3.2. The visual intrusion detection system

The visualization environment, figure 3, is based on OpenGL and is written in C++. The environment parses the hummer database, generating an event list, which is then evaluated sequentially, updating the visual display

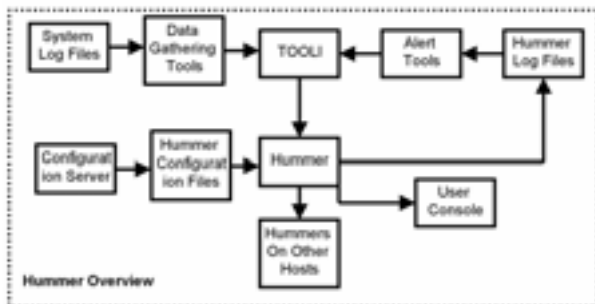


Figure 1. Overview of the basic Hummer data flow and architecture. The path taken by data during collection by the Hummer intrusion detection system.

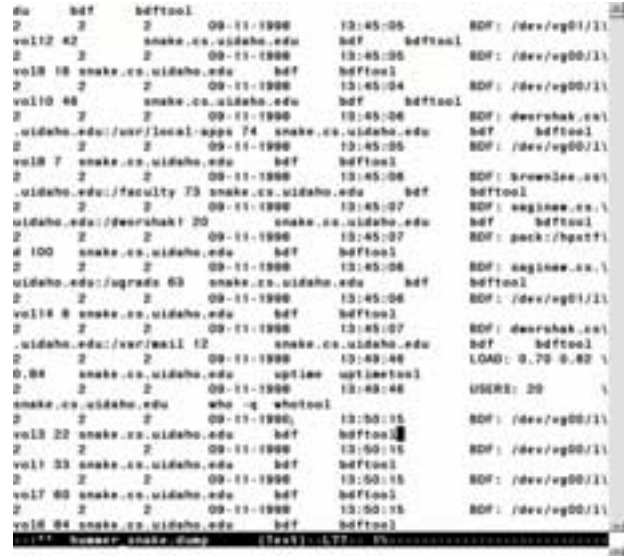


Figure 2. Raw Hummer Data. The unmodified data collected by the Hummer intrusion detection system is shown here.

in correspondence with the events. This design allows for additional database paradigms, e.g., tcpdump, to be added to the environment quickly with only needing to replace the initial data parsing routines.

The system we have implemented currently supports some of the basic information provided in the Hummer database. Our environment creates a visual representation of the nodes in the database and shows data accesses between them, figure 4.

Circles on the screen represent nodes or hosts. The nodes are positioned on the screen in five rings, with the node under analysis located in the center of the screen. The ring for a node is chosen based on the difference between its IP Address and that of the center node. If only the right most number differs it is placed in the first ring and so on. If no valid IP Address could be found for the node it is placed in the fifth ring and is colored red, figure 5. The position of a node is recorded and node

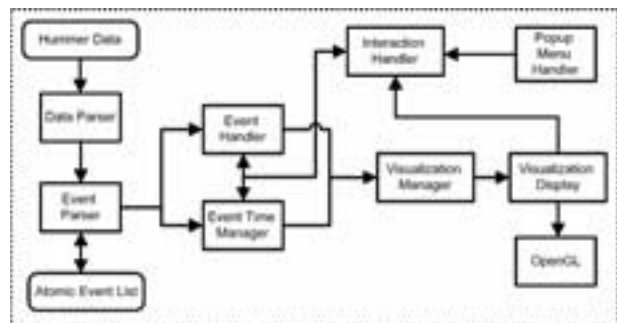


Figure 3. Visualization architectural overview. The path taken by data during analysis by the visualization system for the Hummer log files.



Figure 4. Moderate volume activity analysis in the morning with time based representation of known nodes.

positions are not re-used. This ensures that nodes or hosts always appear in the same position without confusion.

Nodes under investigation have additional information provided, if available. The number of users is represented by spokes extruding from the center of the circle. The system load is represented by the thickness of the circle. The intensity of the node is representative of the duration

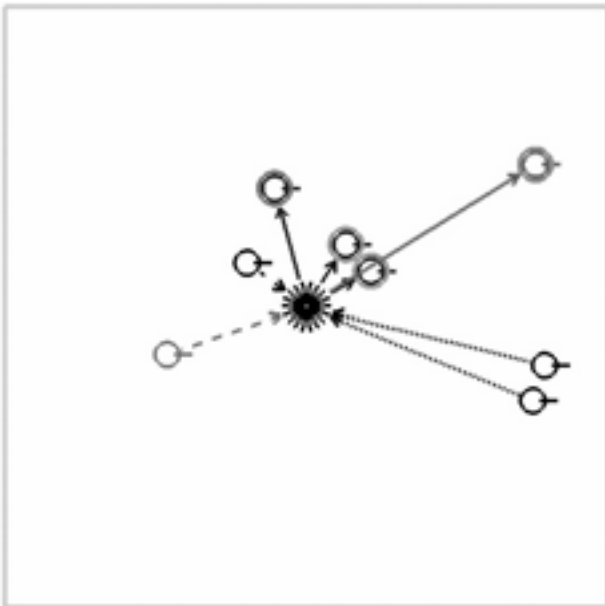


Figure 5. High volume analysis of activity in the afternoon with known and unknown nodes.

since the node was last accessed. After the access to/from the node terminates the node is allowed to fade off, providing a history function. After the node completely fades it is removed. If the node is accessed again before being removed its degradation is reinitialized and it begins to fade again from the beginning.

A solid directed line represents accesses to a remote filesystem, e.g., through nfs mount, from the accessing system to the accessed system. The density of accesses is represented by the intensity of an outer circle around the node. Directed lines are also used to represent ftp accesses from the initiator to the target system. Privileged ftp's are represented using long dashes while anonymous ftp's are represented by dotted lines. The nodes and lines remain active until the ftp session is terminated at which point the node and lines are allowed to degrade (fade).

Since the environment can be used to examine log files, i.e., logs created over a weekend or even just overnight, we provide a visual representation of the time of day. This is done through the intensity of a border around the screen. White represents noon and black represents midnight.

Finally, an important issue is the ability for investigators to continue use of the environment while performing other tasks. This requires that the user be able to shrink the visualization window and still observe sufficient details to identify behavioral patterns. From this visual representation we can discern basic behavioral patterns. Patterns can already be seen that would raise a question in the mind of a system administrator or security personnel. These patterns are constituted by several characteristics that when taken together raise issues when independently they would not be questioned, for example:

1. Lengthy anonymous ftp access after midnight from a site whose IP address cannot be determined.
2. Numerous ftp accesses from closely associated but differing IP addresses occurring in rapid succession.

It is important to note that these behavioral issues are observable as the system executes and the changes in state are animated naturally over time. The behavior of the individuals observed in these animations can be interpreted and characteristics of the individuals can be determined. Static images and text will not fully exhibit these qualities. We must also note that the system itself is creating data that must be accounted for. Figure 6 shows the system in the middle of the night with a single user. The user is accessing two remote nodes. A second user is accessing the system through anonymous ftp. The user on the system itself is the user running the monitoring system, Hummer, and the accesses the system is making to two local nodes is the Hummer system collecting and

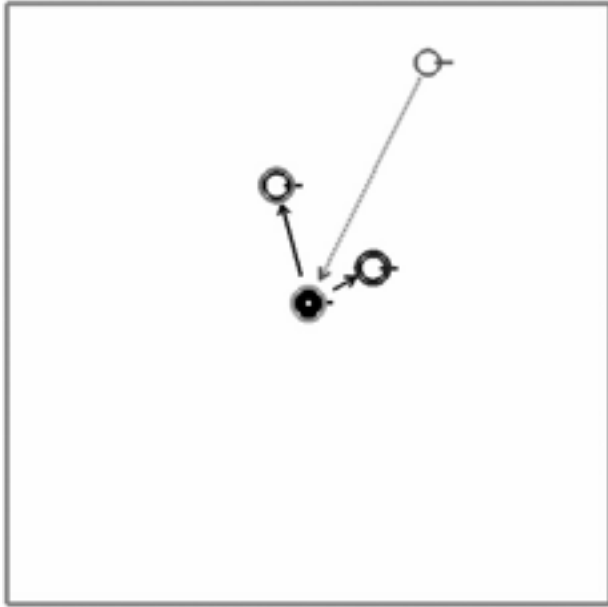


Figure 6: Analysis of low volume activity in the middle of the night.

storing its own data that it has generated. Consequently, we have a probe effect in which by monitoring the environment we are impacting the environment. This must be realized when analyzing the behavior of the system.

3.3. Results and analysis

The environment can be used to identify potential misuse of a system by observing the behavior of the accesses to the system. Any single event on the system is likely not indicative of an intrusion or misuse, it is multiple events when taken together that indicate a potential problem. System administrators or other responsible personnel must be aware of the types of scenarios that could be a lead in to intrusion.

Accessing a computer system from a remote node in and of itself is not an indication of a problem. However, if a remote node begins attempting to access many computer systems in rapid succession then this could be an indication of a problem, particularly if the hostname doing the accesses is not resolvable through name table lookup.

The impact of the probe effect mentioned earlier could be significant on several fronts. First, what impact will our data collection rate have? If we collect too much data we will slow down the network, and will in effect be creating our own intrusion. If we collect too little data we may miss information vital to the identification of an intrusion attempt. These issues feed back on themselves since if we collect too much data the time it will take to

retrieve the data will increase, again slowing our response. How much data should be collected and at what rate? Second, if users know we are monitoring the system will they change their behavior to hide from our monitoring methods? Hackers are known for their ability to conceal their tracks. Will such advanced monitoring systems keep them away? Our visualization methodology is geared towards adaptability, since we are observing unusual behavior rather than specific triggers will we be able to identify hackers who use unexpected methods in an attempt to mask their true intentions? These are questions that will only be answered after the system has been extensively deployed and hackers are aware of the environment, its capabilities, and the need to defeat the system. Only visualization provides the adaptive real-time analysis capabilities needed for such a pervasive and kinetic environment.

What form should our collected data be in? We can compress the data to speed transmission, though this will slow analysis. We can encrypt the data to protect against prying eyes, but again this slows analysis. How confidential is the data being transmitted and how important is network bandwidth compared to processing power? Our goal must be to provide the data needed for analysis without crippling our own network environment.

4. Conclusions

The research we have described in this paper is geared towards analyzing accesses to computer systems through the data collected by the Hummer IDS and visualization techniques being developed specifically for this purpose. We feel this research will greatly improve the ability for our user base to identify intrusions or misuse and trace it to its origin such that it can be combated and eliminated. As the recent web attacks against many popular sites have shown the ability to attack computer systems is NOT a well-guarded secret. Many system administrators and hackers use the same tools to identify weaknesses in system integrity. With the explosive growth in the computer industry these attacks are only going to become more proliferate. The capabilities we are developing are sorely needed and will provide great benefit to the interruption of computer misuse.

5. Future work

Since our environment will be used for both real-time and post-mortem analysis we must provide the ability to analyze both simultaneously. While continuing to examine the network in real-time mode the user may wish to backtrack over a period of time to re-examine a

period of activity or review logs from the previous night. We will need to provide multiple independent time controls to allow the user to specify exact time frames to be reviewed.

The environment must also act as a direct interface for the administrator to the Hummer system and potentially to the network itself. When visualizing the network in real-time it will be imperative to incorporate interactive computational steering techniques [16] to allow the administrator to modify the data that is being collected on individual systems.

We must visually represent the complex interaction among the systems under investigation with only limited display resources. Techniques must be provided to display the entire local network, subsets of the local network, and individual nodes. Magic lens filters [17] provide the ability to focus on selective areas but have not been applied to three-dimensional applications. They also distort the visual representation, which may be too confusing for our user base. Using multiple views or windows may prove beneficial as well [18].

Additional issues are common to these types of environments that we must resolve, for example:

1. How can we visually represent large numbers of nodes?
2. The current prototype works well for a light load, small numbers of links at a time. How can we visually represent very high loads without occluding or hiding important details?
3. Is the current metaphor for visual representation appropriate for our user base or should we choose an alternative visual metaphor?

References

- [1] Picciotto, J., "The Design of an Effective Auditing System," Technical report, 1987.
- [2] D. Zerkle et al. "A Data-Mining Analysis of RTID Alarms", *Recent Advances in Intrusion Detection*, Sept 1999.
- [3] Markus Gross, *Visual Computing, The Integration of Computer Graphics, Visual Perception and Imaging*, Springer-Verlag, 1994.
- [4] William R. Hendee and Peter N.T. Wells, *The Perception of Visual information*, Springer-Verlag, 1994.
- [5] 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.
- [6] Snapp, S. et al., "DIDS (Distributed Intrusion Detection System) Motivation, Architecture and An Early Prototype," *National Information Systems Security Conference*, 1991.
- [7] C. Ko and D. Frincke and T. Goan et al., "Analysis of an Algorithm for Distributed Recognition and Accountability," *ACM conference on Computer and Communication Security*, V1(1), 1993.
- [8] Vert, G., J. McConnell, and D. Frincke. "Towards a Mathematical Model for Intrusion," *21st National Information Systems Security Conference*, pp. 329-337, October 1998.
- [9] Georges Grinstein, "Workshop on Information Exploration Shootout Project and Benchmark Data Sets: Evaluating How Visualization does in Analyzing Real-World Data Analysis Problems," *Proceedings of the IEEE Visualization '97 Conference*, IEEE Computer Society Press, Phoenix, AZ, 1997, pp. 511-513.
- [10] Richard Becker, Stephen Eick, and Allan Wilks. Graphical methods to analyze network data. In *IEEE International Conference on Communications ICC '93 Proceedings*, pages 946-95, Geneva, Switzerland, May 1993.
- [11] Taosong He and Stephen G. Eick. Constructing Interactive Visual Network Interfaces. *Bell Labs Technical Journal*, 3(2)47-57, April-June 1998.
- [12] Kenneth Cox, Stephen Eick, and Taosong He. 3D geographic network displays. *ACM Sigmod Record*, 25(4):50.
- [13] Robert F. Erbacher, "Visual Assistance for Concurrent Processing," University of Massachusetts at Lowell Doctoral Dissertation (1998 CS-3), Lowell, MA 01854.
- [14] 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, 1999, pp. 215-227.
- [15] 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, 1999, pp. 469-492.
- [16] Robert van Liere, Jurriaan Mulder and Jack van Wijk., "Computational Steering," *Future Generation Computer Systems*, Vol. 12, no. 5, 1997.
- [17] M. Sarkar and M. Brown, "Graphical Fisheye Views," *Communications of the ACM*, Vol. 37, no. 12, pp. 73-84, 1994.
- [18] Jonathon C. Roberts, "On Encouraging Multiple Views for Visualization," *Proceedings of the International Conference on Information Visualization*, pages 8-14, London, England, July 29-31, 1998.