Intrusion and Misuse Detection in Large-Scale Systems

Robert F. Erbacher
University at Albany-SUNY

Kenneth L. Walker
New York State Unified Court System

Deborah A. Frincke
University at Idaho

Attacks and misuses of computer systems are major concerns in today's network-based world. We present information visualization techniques based on a glyph metaphor for visually representing textual log information.

Intrusions and misuses of computer systems are becoming a major concern. Our nation's infrastructure is heavily network based in all industries. However, the nation's network infrastructure can't deal with attacks on a local or global scale, leaving network and computer security up to an organization's individual efforts. With the growing concern with regard to cyberterrorism there's a need for new tools and techniques to monitor networks and systems for intrusions and misuse. The goal must be to identify an attack before an organization incurs damage, loses information (theft or otherwise), or has its integrity impugned. With today's network-based economic resources, a successful attack will negatively impact consumer confidence and decrease consumers' willingness to make electronic purchases.

Clearly, even if we can make systems more secure, attacks and internal misuse of technologies will evolve, making some form of intrusion and misuse management a necessity for all systems. We need tools to help detect and eradicate attacks. A root of this problem is that the Internet was initially designed under the guise of open communication. Security has merely been retrofitted on top of the existing infrastructure and not integrated tightly with the design.

Researchers have based most previous work on visualizing network data on measuring performance or bandwidth characteristics. (See the "Previous Work" sidebar for more detailed information.) Little prior work has dealt with visualizing network intrusion data, particularly real-time network intrusion data, as is our ultimate goal. We're attempting to visualize the actions of an enterprising hacker actively seeking to counter the attempts we make to identify that hacker's actions.

The capabilities of computer and network security aren't yet in place to let system administrators efficiently detect and counteract intrusions and misuses of the systems and networks under their control. Only through innovative, new technologies can we hope to counteract the growing threat from hackers. Perusal of textual log files is totally inadequate. Of the techniques available, visualization appears well placed to take on the brunt of this task. As a result, we've developed information visualization techniques based on a glyph metaphor for visually representing the textual log information collected from systems.

Scale of the problem

Examining the computing environment at the University at Albany-SUNY provides some insight into the scale of this problem. The university's primary Unix server provides email and computing resources for the entire university. As many as 200 users access the system simultaneously. During a one-week period, the server handles more than 25,000 successful connections from over 2,500 different hosts and an undeterminable number of unsuccessful connections. The number of different hosts connecting to the system generates a significant issue of scale for any visualization technique. In addition, the university maintains many other Unix systems that it must constantly monitor.

Many individual departments within the university maintain their own networks of computing systems that must also be monitored. This is important because interactions between the separate networks of computing systems are critical to identifying how the systems are being used and for identifying distributed and complex attacks. Monitoring the primary Unix server and a dozen other systems generates approximately 500,000 log messages each week. We expect the scale of the university's problem to be comparable to most organizations.

Currently, system administrators are required to analyze log files to identify an attack. These log files can incorporate hundreds if not thousands of messages per day. The amount of data available results in system administrators not fully collecting or monitoring all available information for all systems under their con-

0272-1716/02/$17.00 © 2002 IEEE

Previous Work

Researchers have based most previous work on visualizing network data on measuring performance or bandwidth characteristics, but little work has dealt with visualizing real-time network intrusion data. This dynamism of the computer and network security problem alone requires novel solutions.

Intrusion detection systems

Aside from our ongoing work in this area,[1] researchers have applied little prior activity to using visual analysis as an aid to intrusion detection. For instance, many researchers have proposed using 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[2] indicator. Earlier systems, such as the Distributed Intrusion Detection System (DIDS),[3] provided color graphical representations to indicate when a system had experienced a sequence of suspicious events. While useful, these approaches don't provide adequate detail to do more than observe that attacks are in progress and do little to aid diagnosis. Frincke[4] has performed preliminary investigations toward identifying likely models for depicting system state.

Visualization systems

In contrast to intrusion detection, researchers have applied quite a bit of visualization research 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 the US National Institute of Standards and Technology.[5] In this project, researchers were given access to a data set consisting of network intrusions. The idea was to identify which researcher's techniques were effective at identifying the intrusions. The driving philosophy was that researchers had done little work to compare visualization techniques in a formal setting. They have done perceptual studies to identify characteristics of the human visual system that should be used as a basis for developing visualization techniques but have done little to actually compare and contrast visualization techniques. No body of literature exists 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[6,7] even down to the router,[6] individual packets, and individual email messages.[8] These techniques don't provide sufficient detail or handle sufficient numbers of nodes and attributes in combination for our needs. The work by Eick et al.[8] strictly deals with email and subsequently resolves many fewer nodes and attributes than is needed for intrusion detection. Other work has been geared toward visualization systems for program analysis and development. These environments typically deal with small numbers of processors that work on a single task and thus have a common grounding. Researchers haven't applied this research into network usage to network intrusions.

Becker et al.[9] discuss the SeeNet environment that provides linkmaps to visually represent the amount of data being sent between two network nodes. It can identify when a node is overloaded, the network's behavior, and how data moves between locations and its volume. This is critical during a crisis and usage increases—for example, after a California earthquake. Understanding the consequences of events, so that, for example, telephone companies can be prepared for changing demands, is imperative.

Livelink[10] is an environment for visualizing and measuring the Web. By probing Web accesses, Livelink gathers statistics on the number of hits Web sites receive. The environment visually presents this statistical information as charts and graphs. An extension to the environment provides a more advanced graphical representation, which represents the approximate location of network nodes, showing geographical association between Web sites. Each node's visual representation is then presented to reveal the site's activity. Each node can represent several parameters simultaneously.

Netmap[11] is a generic visualization tool for representing relationships within a data set. The environment is principally geared toward showing known relationships of static data sets. Netmap isn't geared toward exploratory data analysis in which the relationships are unknown and must be identified, or for temporally changing data sets. In contrast, it's the unknown nature of intrusion data that's the driving force behind the visualization techniques we're developing.

References

1. R.F. Erbacher and D. Frincke, "Visualization in Detection of Intrusions and Misuse in Large-Scale Networks," *Proc. Int'l Conf. Information Visualization 2000*, IEEE CS Press, Los Alamitos, Calif., 2000, pp. 294-299.
2. D. Polla et al., "A Framework for Cooperative Intrusion Detection," *Proc. 21st Nat'l Information Systems Security Conf.*, Nat'l Inst. of Standards and Technology, Washington, D.C., 1998, pp. 361-373.
3. S. Snapp et al., "DIDS (Distributed Intrusion Detection System) Motivation, Architecture and An Early Prototype," *Proc. Nat'l Information Systems Security Conf.*, Nat'l Inst. of Standards and Technology, Washington, D.C., 1991, pp. 167-176.
4. G. Vert, J. McConnell, and D. Frincke. "Towards a Mathematical Model for Intrusion," *Proc. 21st Nat'l Information Systems Security Conf.*, Nat'l Inst. of Standards and Technology, Washington, D.C., 1998, pp. 329-337.
5. G. Grinstein, "Workshop on Information Exploration Shootout Project and Benchmark Data Sets: Evaluating How Visualization Does in Analyzing Real-World Data Analysis Problems," *Proc. IEEE Visualization 97 Conf.*, IEEE CS Press, Los Alamitos, Calif., 1997, pp. 511-513.
6. K. Cox, S. Eick, and T. He, "3D Geographic Network Displays," *ACM Sigmod Record*, vol. 25, no. 4, Dec. 1996, p. 50.
7. E.E. Koutsofios et al., "Visualizing Large-Scale Telecommunication Networks and Services," *Proc. IEEE Visualization 97 Conf.*, IEEE CS Press, Los Alamitos, Calif., 1997, pp. 457-461.
8. S.G. Eick and G.J. Wills, "Navigating Large Networks with Hierarchies," *Visualization 93 Conf. Proc.*, IEEE CS Press, Los Alamitos, Calif., 1993, pp. 204-210.
9. R. Becker, S. Eick, and A. Wilks, "Visualizing Network Data," *Readings in Information Visualization: Using Vision To Think*, S. Card, J.D. Mackinlay, and B. Shneiderman, eds., Morgan Kaufman, San Francisco, 1999, pp. 215-227.
10. T. Bray, "Measuring the Web," *Readings in Information Visualization: Using Vision To Think*, S. Card, J.D. Mackinlay, and B. Shneiderman, eds., Morgan Kaufman, San Francisco, 1999, pp. 469-492.
11. C. Davidson, "What Your Database Hides Away," *New Scientist*, no. 1855, 9 Jan. 1993, pp. 28-31.

trol, rather focusing on primary systems and servers and examining minimal information from the remaining systems. In fact, system administrators generally don't collect or analyze any data related to Microsoft- or Apple-based operating systems, even though it's feasible for these systems to be the target of break-ins, subversions, and misuses. They only intermittently analyze network traffic itself. This leads to situations where misuses or intrusions might not be detected for some time—for example, the misuse of the CIA computers for Internet Relay Chat (IRC).

Log-file analysis is becoming the greatest time consumer for system administrators. Identifying actual intrusions and misuses requires that they must know the user's intentions while examining the user's activity because the same activity can have a more disconcerting meaning under different auspices. Manually reviewing these log files is currently unfeasible and results in missed attacks and false alarms. This situation is only likely to get worse, and with the globalization of e-commerce and interest in Internet voting, the potential for serious damage increases. Ultimately, the goal must be to identify an attempted break-in or attack before the attack is successful to allow a response to initiate before an organization incurs damage.

Current log-file analysis only reveals that an attack has occurred in the past. At this point, we might not be able to determine if the attack was successful because hackers generally subvert the log reporting facilities as one of their first actions. This then requires extensive analysis to determine each system's integrity. It's imperative that we reduce the number of false alarms and increase the number of real attacks detected.

Figure 1 shows an example of a log file for five lightly loaded workstations collected in one hour. This log information was collected during winter recess within the Computer Science Department at the University at Albany-SUNY. Because students weren't present, the amount of information is small, approximately 40 messages. During the semester, when the systems are more active, there will generally be approximately 150 messages per hour. The systems only collected a limited amount of the available information. The department's machines are lightly loaded compared to the university's principal server that has a hundred times the usage. If you can imagine attempting to parse through such textual information on a repeated basis to derive greater understanding and relationships, you'll see the true futility of the task when given large complete databases.

Attacks on a system can range from an attempt to gain entry and subvert it or merely to access a system remotely, either to access data or disable it. In the latter case, denial-of-service attacks are the greatest concern. In this scenario, attackers send requests to the server with the intent of bogging down the system by forcing it to analyze and respond to requests that the client never completes. These incomplete transactions stay active in the system, consuming resources. At a massive scale, the system will effectively become useless, preventing the connection requests of legitimate users from succeeding. By identifying this type of activity, a system administrator can take action to ensure such an attack doesn't disrupt legitimate system activity.

With large environments, it's difficult for a system administrator to keep track of the configuration of all systems in the environment. This is complicated by the growing population of somewhat-savvy users who install software or change a system's configuration without understanding the impact of their modifications on security and reliability. It's rarely through primary systems, over which the system administrator keeps a careful eye, that an attack first occurs. With tens or hundreds of systems under an administrator's auspices, analyzing the log files for all the machines and interrelating their usage is a daunting task. Such complete analysis, while necessary for a truly secure environment, is rarely performed by an organization and never on a continuous basis. We're empowering system administrators with the ability to perform such a feat.

Data collection

We collected the majority of our data from the university systems using the Hummer intrusion detection system.[1] This data consists of the information available through normal log files with additional statistics and other available system information. This prevents the need for specialized kernel modifications or performance draining data collection tools. In addition, we merged the log files for all systems under consideration into a single Postgres database, providing for easy querying. The data can then be analyzed in a postmortem or real-time format, depending on the current needs of the analysis process.

The total volume of data collected through log files varies significantly from one system to another, depending on usage from 30,000 records a week for a lightly used workstation to 200,000 records a week for critical servers. The total number of events an administrator must analyze and interpret in relation to one another is huge. We collected approximately 500,000 records over the course of a week from the university's principal server and a dozen other workstations.

Data collection and filtering techniques have greatly aided the analyses. However, examining commercial and research efforts to identify security violations consistently generate considerable quantities of data—usually far too much to be evaluated effectively using current techniques.[2] Some of this is due to the way that data-gathering choices are made by administrators.[2] However, refinements in the data-gathering decision-making process won't suffice. As networks grow larger, the amount of relevant data that's misused will also grow. Hence, we need better methods for analyzing the data rather than continuing to rely on primarily textual techniques.

The visualization environment

The problem with analyzing log files is that reading textual information is inherently a perceptually serial process. Interpreting graphical images, on the other hand, is perceptually a parallel process. Forcing the user to use textual information, therefore, slows the analysis process substantially in comparison to using graphics. An additional advantage of imagery is that we can present more concepts in a single image than a comparable volume of

```
Jan 9 12:15:12 visualizer-s.cs.albany.edu xinetd[899]: START: pop3 pid=28097 from=169.226.2.54
Jan 9 12:15:12 visualizer-s.cs.albany.edu xinetd[28097]: USERID: pop3 WIN32 : Administrator
Jan 9 12:16:31 broomstick.cs.albany.edu in.telnetd[16593]: connect from root@cs.albany.edu
Jan 9 12:16:31 cs.albany.edu in.telnetd[16593]: connect from root@cs.albany.edu
Jan 9 12:20:24 broomstick.cs.albany.edu sshd[238]: log: Generating new 768 bit RSA key.
Jan 9 12:20:33 broomstick.cs.albany.edu sshd[238]: log: RSA key generation complete.
Jan 9 12:22:29 visualizer-s.cs.albany.edu CROND[28100]: (root) CMD ( /sbin/rmmod -as)
Jan 9 12:25:31 broomstick.cs.albany.edu in.telnetd[16628]: connect from cdial20.infoblvd.net
Jan 9 12:25:31 cs.albany.edu in.telnetd[16628]: connect from cdial20.infoblvd.net
Jan 9 12:26:02 cs.albany.edu named[25266]: dangling CNAME pointer (google.lb.google.com)
Jan 9 12:29:45 broomstick.cs.albany.edu in.telnetd[16654]: connect from Workstation72.ctg.albany.edu
Jan 9 12:29:45 cs.albany.edu in.telnetd[16654]: connect from Workstation72.ctg.albany.edu
Jan 9 12:29:51 von.cs.albany.edu in.rlogind[5625]: connect from pb@broomstick.cs.albany.edu
Jan 9 12:30:13 visualizer-s.cs.albany.edu xinetd[899]: START: pop3 pid=28101 from=169.226.2.54
Jan 9 12:30:13 visualizer-s.cs.albany.edu xinetd[28101]: USERID: pop3 WIN32 : Administrator
Jan 9 12:31:30 cs.albany.edu named[25266]: Cleaned cache of 799 RRs
Jan 9 12:31:30 cs.albany.edu named[25266]: USAGE 979061490 977153081 CPU=447.12u/258.82s CHILDCPU=0u/0s
Jan 9 12:31:30 cs.albany.edu named[25266]: NSTATS 979061490 977153081 Unknown=6 A=393521 NS=3 CNAME=98 SOA=9575
PTR=73966 MX=15120 TXT=10 AAAA=42 AXFR=32 ANY=12019
Jan 9 12:31:30 cs.albany.edu named[25266]: XSTATS 979061490 977153081 RR=198301 RNXD=66697 RFwdR=150932
RDupR=302 RFail=619 RFErr=0 RErr=17 RAXFR=32 RLame=16943 ROpts=0 SSysQ=23483 SAns=373313 SFwdQ=131146
SDupQ=30183 SErr=0 RQ=504450 RIQ=0 RFwdQ=131146 RDupQ=2489 RTCP=1069 SFwdR=150932 SFail=3460 SFErr=0
SNaAns=68541 SNXD=241409
Jan 9 12:32:28 visualizer-s.cs.albany.edu CROND[28103]: (root) CMD ( /sbin/rmmod -as)
Jan 9 12:34:07 karp.cs.albany.edu in.telnetd[27063]: connect from nas-70-57.albany.navipath.net
Jan 9 12:34:17 cs.albany.edu named[25266]: dangling CNAME pointer (gd25.doubleclick.net)
Jan 9 12:42:29 visualizer-s.cs.albany.edu CROND[28105]: (root) CMD ( /sbin/rmmod -as)
Jan 9 12:45:12 visualizer-s.cs.albany.edu xinetd[899]: START: pop3 pid=28106 from=169.226.2.54
Jan 9 12:45:12 visualizer-s.cs.albany.edu xinetd[28106]: USERID: pop3 WIN32 : Administrator
Jan 9 12:51:36 karp.cs.albany.edu in.telnetd[27115]: connect from [B49DGoPz/VbUaLLZxAn44kzfOa4FzOiQ]@hopper.sdsc.edu
Jan 9 12:52:29 visualizer-s.cs.albany.edu CROND[28108]: (root) CMD ( /sbin/rmmod -as)
Jan 9 12:52:33 karp.cs.albany.edu in.telnetd[27137]: connect from 169.226.14.70
Jan 9 13:00:12 visualizer-s.cs.albany.edu xinetd[899]: START: pop3 pid=28109 from=169.226.2.54
Jan 9 13:00:12 visualizer-s.cs.albany.edu xinetd[28109]: USERID: pop3 WIN32 : Administrator
Jan 9 13:02:29 visualizer-s.cs.albany.edu CROND[28111]: (root) CMD ( /sbin/rmmod -as)
Jan 9 13:03:29 visualizer-s.cs.albany.edu CROND[28113]: (root) CMD (run-parts /etc/cron.hourly)
Jan 9 13:06:44 cs.albany.edu in.telnetd[8550]: twist vaughn.arch.rpi.edu to /bin/echo “Service not permitted”
Jan 9 13:08:30 broomstick.cs.albany.edu in.telnetd[16702]: connect from cm-24-29-78-15.nycap.rr.com
Jan 9 13:08:30 cs.albany.edu in.telnetd[16702]: connect from cm-24-29-78-15.nycap.rr.com
Jan 9 13:11:43 karp.cs.albany.edu in.telnetd[27175]: connect from grande.cs.albany.edu
Jan 9 13:12:29 visualizer-s.cs.albany.edu CROND[28115]: (root) CMD ( /sbin/rmmod -as)
Jan 9 13:12:33 cs.albany.edu printer: offline or intervention needed
Jan 9 13:12:33 cs.albany.edu printer: error cleared Jan 9 13:14:55 cs.albany.edu named[25266]: dangling CNAME pointer
(md1.doubleclick.net)
```

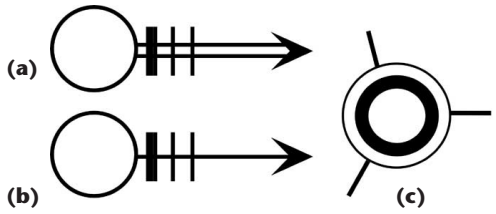
1 Example of a log file for five machines over the course of an hour in a lightly loaded environment.

text. Thus, a single image can embody the same information as an entire report or report summary. This reduces the amount of mental context switching required by users, making system assessment more efficient.

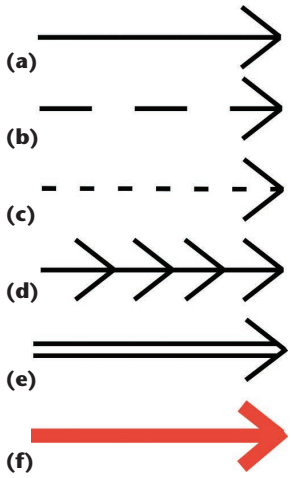
The visual intrusion detection system

Our visualization environment creates a visual representation of the systems in the database as glyphs (see Figure 2). These glyphs incorporate visual attributes representative of parameters in the database, including the number of users, system load, status, and unusual or unexpected activity. It's important to create the visual attributes in conjunction with the database parameters such that the correlation is appropriate and the relationship is comprehensible to the analyst. The glyphs and visualization match an administrator's expected view of the network, and the visual attributes are easily interpretable for their actual meaning.

We represent initial connection requests as parallel lines. This occurs when the client first initiates a connection request with the server but before the client has successfully passed the authentication challenge. This information is available for FTP, telnet, and rlogin connections. On Sun systems this information is collectable by running inetd with the ë-t option. Figure 2a shows the parallel lines as they are represented visually. Upon a successful connection, we remove and replace the parallel lines with a single line representative of the type of connection (see Figure 2b). All lines are representative



2 Basic glyph organization. (a) The initial inetd connection to the system. (b) The resulting connection after authentication. (a) and (b) also represent the number of users with connections from the given remote host and the number of connections by said users through the use of the cross hatches. The monitored system, (c) showing number of users and load.



3 Line appearances and their relationships. (a) Telnet and rlogin connections as solid lines, (b) privileged FTPs as long dashed lines, (c) anonymous FTPs as short dashed lines, (d) Network file system (NFS) accesses as solid lines with many arrows, (e) initial inetd port connection, and (f) port scan.

of the direction of the connection. Figure 3 shows additional information encoded into the line style.

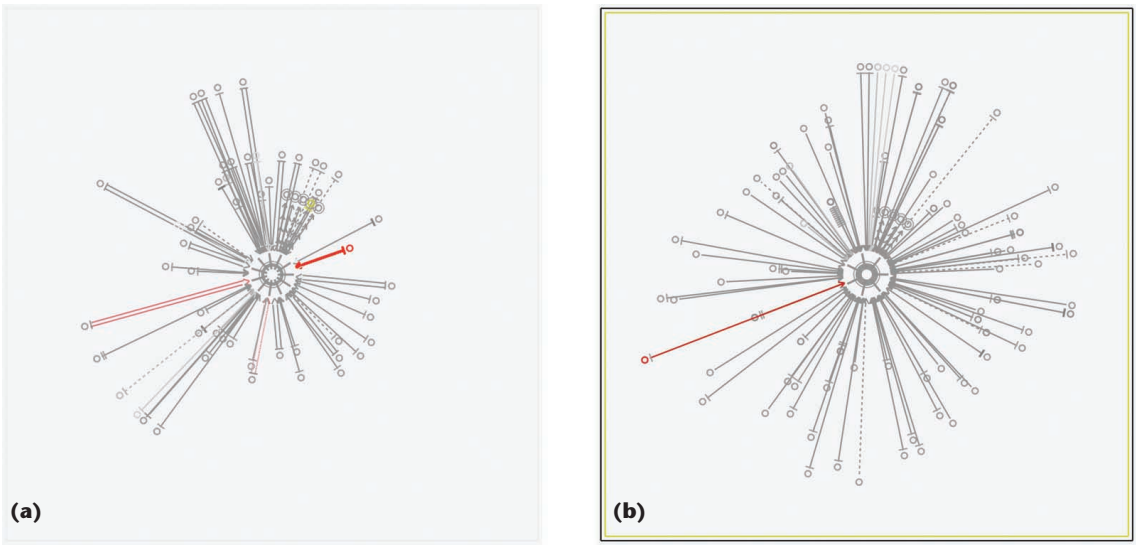
The nodes and lines remain active until the user terminates the session, at which point the node and lines degrade (fade). This degradation provides temporal relationships and assists in making events more persistent. Because many of the events, such as port scans, are instantaneous, it's necessary to ensure the events exist on screen for a sufficient duration so administrators can perceive them.

We represent the number of user connections from one system to another as hashes along the directed line. A single hash mark represents each user. Thicker hashes represent users with multiple connections. With this information, administrators can measure activity between systems and monitor behavior patterns.

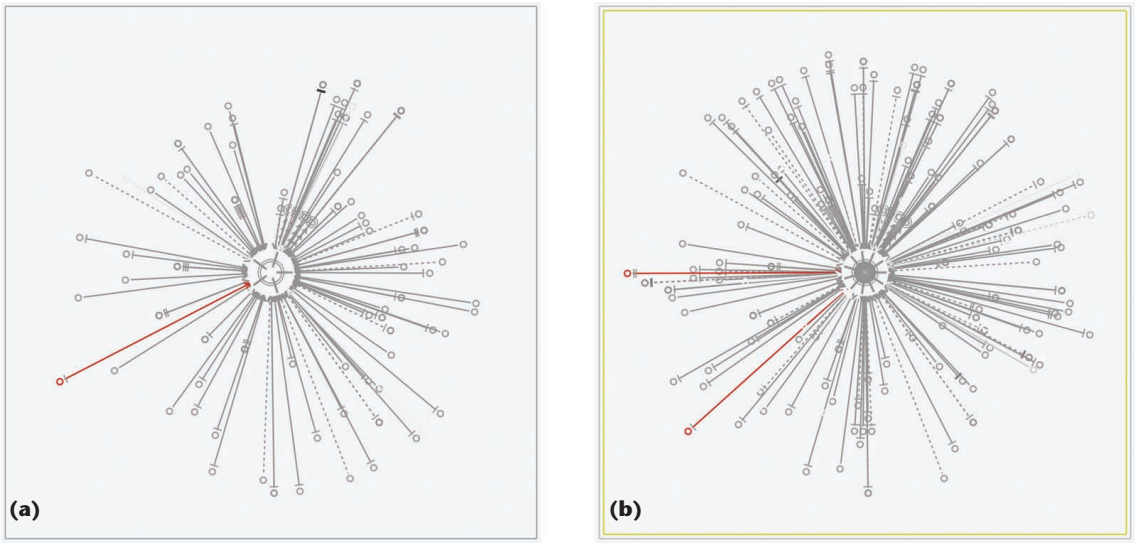
We use red to highlight unusual or unexpected activity. Depending on the type of event, the system will either highlight an individual node, the directed line between two nodes, or both. Yellow highlights questionable activity that isn't as critical in importance. Nodes themselves highlight red when a user executes a su or sudo command. (These commands provide privileged users with access to protected systems resources.) Nodes turn yellow when NSF mounts to that system don't respond. We use red links when timeouts expire and the user hasn't successfully passed the authentication challenge. Red nodes and links are used when port-sentry—an intrusion detection tool designed to capture attempts to access inappropriate system resources—identifies an attack or when the hosts address can't be resolved using a domain name server (DNS) lookup. In the last example, we made the port-sentry attacks stronger and brighter because of their greater importance.

We provide additional information for nodes under investigation (see Figure 2c), if it's available. Spokes extending from the perimeter of the circle represent the number of users. Each spoke represents 10 users. The number of users simultaneously on the system makes representing them individually ineffective. The inner circle's thickness represents system load. The node's intensity represents the time elapsed since the node was last accessed. After all accesses to or from the node terminate, the node fades off, providing a history function and assisting in the representation and comprehension of temporal information. After the node completely fades, it's removed. If the node is accessed again before being removed, we render it at full intensity, and fading begins anew.

Figure 4a shows an example with many of these features. This snapshot is from early in the morning so many nodes are attempting initial connections. To assist with determining the time of day, the full display



4 Basic visual representation of network and system activity. (a) Two connections that failed to authenticate, a port-sentry identified attack, a lost NFS mount, several initiated inetd connections, ftp and telnet connections during the morning. (b) A higher level of activity but with fewer anomalies during the late evening. Notice the large number of users connecting from one particular system to the main server in the CS department. We can identify an anomaly at the top where many remote systems are connecting to the server in sequence for short periods.



5 **Basic visual representation of network and system activity. (a) A large volume of activity shortly after midnight with few anomalies. Notice the change in the border color from Figure 4b. (b) Notice how high the traffic load can become on the system. We can still clearly identify the individual activity.**

includes a border around the screen. The intensity (gray level) of this border is white at noon and black at midnight. An additional yellow border indicates p.m. This assists in determining if the intensity is increasing or decreasing at any particular point in time.

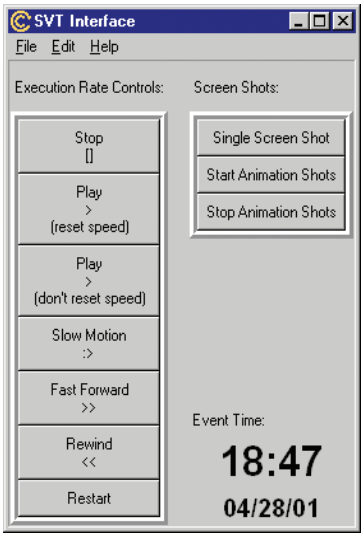
The example in Figure 4a has a light border without a yellow border so it was clearly approaching noon at the point this image was taken. The larger circles are the principal systems being monitored. The smaller circles are remote connections, including both local and remote connections as well as Unix- and non-Unix-based systems. Figure 4a also shows two connection attempts that failed to successfully pass the authentication challenge before timing out, an NSF mount that doesn't seem to be responding to queries, and a port-sentry attack. Note that the port-sentry attack appears to be a local system and thus should be investigated. We incorporated node locality into the node position.

The nodes are positioned on the screen in five rings. The ring for a node is chosen based on the difference between its IP address and that of the monitored system. If only the right-most number differs, then the node is on the local subnetwork and is placed in the first ring, closest to the monitored system, and so on. This is representative of the user's network locality relative to the local network. We can easily identify users on the same subnet as the monitored system. This assists in identifying the activity on the system because local users' activity will be clearly different from nonlocal users, and it shows multiple and indirect accesses that otherwise wouldn't be visible. Because most attacks originate from within the organization, this is critical, particularly because internal personnel will often begin with additional knowledge and information to assist them in making a successful attack or disruption of the organization's systems. If the IP address for the node couldn't be resolved—that is, the system isn't known by the DNS for some reason—it's placed in the fifth ring and colored red. This occurs frequently when attackers or other users spoof their IP address to avoid detection or identification. We record a node's position and don't reuse node positions. This ensures that nodes or hosts always appear in the same position. This is critical for identifying temporal relationships or anomalies related to a single host's network activity.

We place the primary rings we just described sufficiently far apart to let each expand when the first pass becomes filled. In this way, each ring will slowly expand to accommodate additional nodes. With this technique, the environment can handle several thousand node positions without overlap. This lets us uniquely represent a substantial number of different connecting systems. A future task will require the incorporation of replacement strategies to let us represent unlimited numbers of nodes over time through the reuse of node positions. Given that there are generally only 150 nodes used at any given time, the current technique provides sufficient space to prevent substantial occlusion from nodes and directed lines overlapping.

Figure 4b shows an example in the p.m., actually close to midnight, when there are many more connections. This example also shows a single system with many individuals connected to the monitored system, fading nodes, and a system whose IP address can't be resolved. Note that as nodes are positioned in rings and a ring is completely filled, the nodes are placed in a new ring immediately outside of the just filled ring. This becomes critical when there are large numbers of hosts connecting. The fundamental rings are spaced enough to allow for this growth.

Figures 5a and 5b show additional examples. Figure 5a shows a lightly loaded environment early in the morning and several facts about the environment. First, the environment generally reaches a steady state when many connections are maintained continuously. This can make analysis of connections during the night easier to analyze because most of the activity is actually static. This type of



6 User interface for the visualization environment.

phenomenon would be incredibly difficult to identify and correlate if an administrator was only analyzing textual log files. Identifying the steady state requires viewing the animated visualization, showing temporal information. This leads to another question: Are all these users enacting some kind of mechanism to ensure the security of their systems during the middle of the night? Maintaining indefinite connections of this form is a bad habit. As long as the workstations are locked or some other security provisions are provided, there shouldn't be a problem, but if others have access to the system, unauthorized individuals can use it inappropriately.

The final example, Figure 5b, shows the environment in the afternoon at its highest point of use. Many connections are coming and going in rapid successions, and the server is heavily loaded. This example clearly shows how much activity there is on the system and the volume of activity that needs to be analyzed to identify inappropriate uses or intrusions. Represented textually, a full analysis of such activity is impossible.

The user interface

Our user interface provides basic capabilities for controlling the environment. In addition to a typical menu, the user interface contains convenient buttons for taking snapshots and animations of the environment (see Figure 6). This is critical if the system administrator needs to capture information relevant to an attack for future analysis.

The exact time of day is also provided in the user interface so that the users can focus on a single display and not miss any critical events. It's often necessary to know the exact time of day, particularly when action must be taken and details of an attack forwarded to the offending site. The remainder of the interface is occupied by VCR-like controls that let an administrator control the simulation's execution rate, speeding it up, slowing it down, pausing it, and rewinding it as necessary.

System implementation

We implemented this environment in C++ using OpenGL and Tcl/Tk. The GUI runs in a separate execution thread, ensuring effective response to user interaction. Communication between the two threads is performed through semaphores and shared memory. Because the node locations depend on the system's IP address, IP address and hostname pairs are cached and stored in an external file for quick recovery. This occurs for both successfully identified addresses and hosts, and hosts that fail to resolve. This greatly improves performance for known hosts because the time delay incurred for many name resolution calls is substantial.

On a 1.7-GHz Pentium 4 with a FireGL 2 card, the system generates approximately 100 frames per second. Each frame consists of approximately 150 active nodes on the system. This is more than sufficient to adequately display all incoming events, even on slower systems because they arrive without any delay. Therefore, the implementation's focus is on providing the system administrator with an effective experience rather than speed and efficiency. If the environment runs too quickly or displays an event for too short a time, the system administrator might completely miss an event or not fully absorb the event's impact and its temporal relationship to other events. On the other hand, when attempting to review longer periods for identification of temporal activity, the system's performance is sufficient for reviewing vast periods.

Attack analysis

Analyzing the collected information and determining if an attack or misuse is occurring requires that an administrator analyze the individual's intent or behavior. Currently, when suspicious behavior is noted, the individual's activities are examined, most often after the fact. With visualization, we can examine an individual's activity as it's occurring and determine immediately, before substantial harm has been done, that the individual's activities are unacceptable. Even suspicious activity can be difficult to detect with standard log-file-based approaches that require the system administrator to peruse textual information.

We can incorporate network traffic data into the display, letting administrators quickly examine the data for particular types of traffic, such as illegal systems on the network, improper application usage, or connections from unknown systems or users. For example, the personnel at the CIA who ran an illegal chat room could have been detected through the analysis of network traffic information that would have identified the characteristic IRC packets on the network, a clear indication of misuse. Because the information isn't being read textually but visually through a graphical display, the gigabytes of information related to network traffic and user space applications can quickly be analyzed at intervals.

An individual's single actions alone don't provide much context or basis for their motivation in their activity. Certain activities clearly indicate illegal system usage, however, and these actions are most often identified in users who are inexperienced in subverting a system. We can easily identify these novice users with conventional techniques. Therefore, our principal con-

cern is with experienced hackers who will attempt to hide their tracks or camouflage their actions. In these situations, the users' overall actions when taken together will clearly indicate their overall motive. In this fashion, we're providing system administrators with tools that let them visually examine the activity on the computer systems as well as network usage in a merged environment.

Behavior identification

In everyday life, we must ascertain individuals' intent and motivation. In a computing environment, the level of information we use socially isn't available. We must collect the information that's available and provide it in a form that lets us examine the user's activities. We can derive individuals' behavior from the activities they perform, when they perform these activities, the order they perform them in, and how the presence of others affects their activities. At issue is the need to collect much information that currently remains unused due to the clutter in log files.

For example, if we consider the example in Figure 5a, we can see that numerous users are accessing the system. Most of these accesses are static, carryovers from the daytime and resulting from individuals who didn't log off. There's one node of interest: a user's connection that's red because a reverse hostname lookup failed for that system. Taking individual actions alone aren't enough to comprehend the meaning of this activity. If, however, we consider that the user is performing a telnet operation in the middle of the night from a hostname that we can't look up, then the situation begins to appear objectionable. The user's saving grace is that the connection appears static, showing up in Figures 4b and 5b. Thus, all the characteristics taken together tell a story about users and their activity. An administrator can then use this to derive the meaning of an activity and determine if it should be considered objectionable and what level of action should be taken.

This ability to take multiple characteristics, through multiparametric visualization techniques, and integrate them to find a greater understanding is the key to analyzing network and computer usage for intrusions and misuses. This becomes even more important when analyzing heavily used systems and examining multiple systems, particularly to decipher a user's actions across multiple systems.

Figure 4b shows a second example. The six nodes at the top are connections that were made in rapid succession from different IP addresses. Notice that these nodes aren't within the university's local network. Is this an indication of an attack? Had they been local to the university's network they would have been deemed to be students logging on immediately after class. Once removed from the university, seeing such sequences in rapid succession should raise concern. Other types of activity in conjunction could have made the scenario more or less objectionable, particularly if they had been port scans. However, visually, this display clearly shows the actions of users on the system and how connections are being made. These types of activities are warning signs of possible intrusions or misuses. The administrator's knowledge of local behavior is imperative to making sense of the data.

Although it's becoming critical for organizations to maintain 24/7 network monitoring, particularly when the organization's livelihood depends on the availability of the organization's computer systems, this is often unfeasible.

It's important to note that these behavioral issues are observable as the system executes and the changes in state are animated. An administrator can interpret the behavior of the individuals observed in these animations and determine the individuals' characteristics. Static images and text won't exhibit these qualities as clearly.

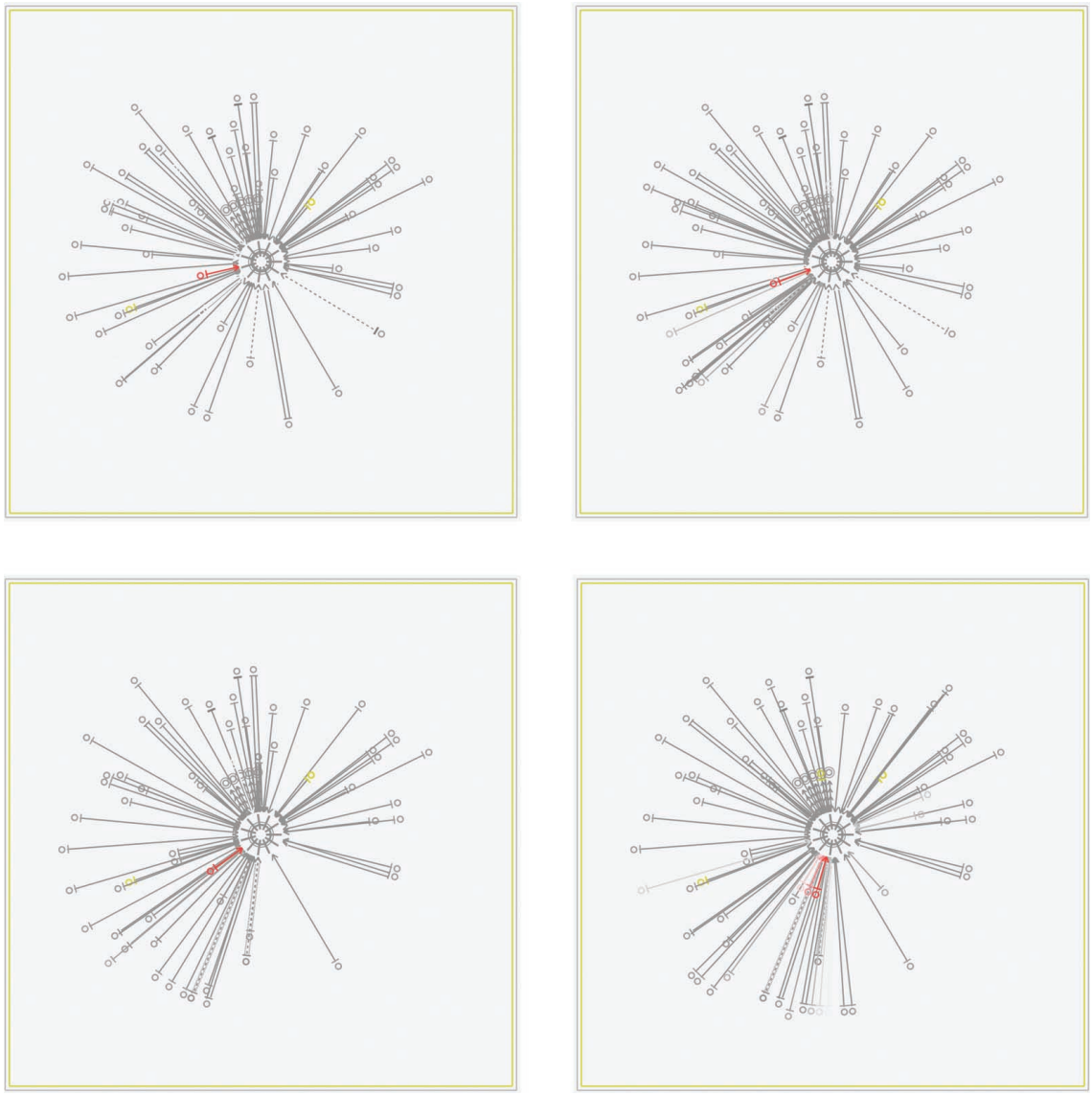
Attack examples

We designed this environment for system administrators. Because network monitoring is generally only one task of any system administrator, we designed the visualization to be effective as a small window on the administrator's screen. Thus, we designed the environment to be an additional tool at the system administrator's disposal. Although it's becoming critical for organizations to maintain 24/7 network monitoring, particularly when the organization's livelihood depends on the availability of the organization's computer systems, this is often unfeasible. Therefore, the VCR-like controls provide the administrator the ability to review periods of time during which the network wasn't monitored, such as overnight and on weekends, and to review in more detail periods of questionable activity.

The most important aspect of the environment for identifying an attack or intrusion before damage is incurred is the identification of temporally related events (see Figures 7 and 8, next page).

The sequence of actions in Figure 7 shows many failed connections for a sequence of local hosts. Because the connecting IP addresses appears to change according to a preordained sequence, this is clearly an attack and not merely a student failing to successfully connect. Because the connections are occurring from different hosts makes this difficult to identify using normal log-based analysis. The short difference in time between the attempts also adds to the analysis.

Figure 8 shows an example of a port-scan-based attack. The offender has scanned the system from off site and is now attempting to connect to the system through a variety of protocols. There are several failed inetd connections immediately after the port scan,



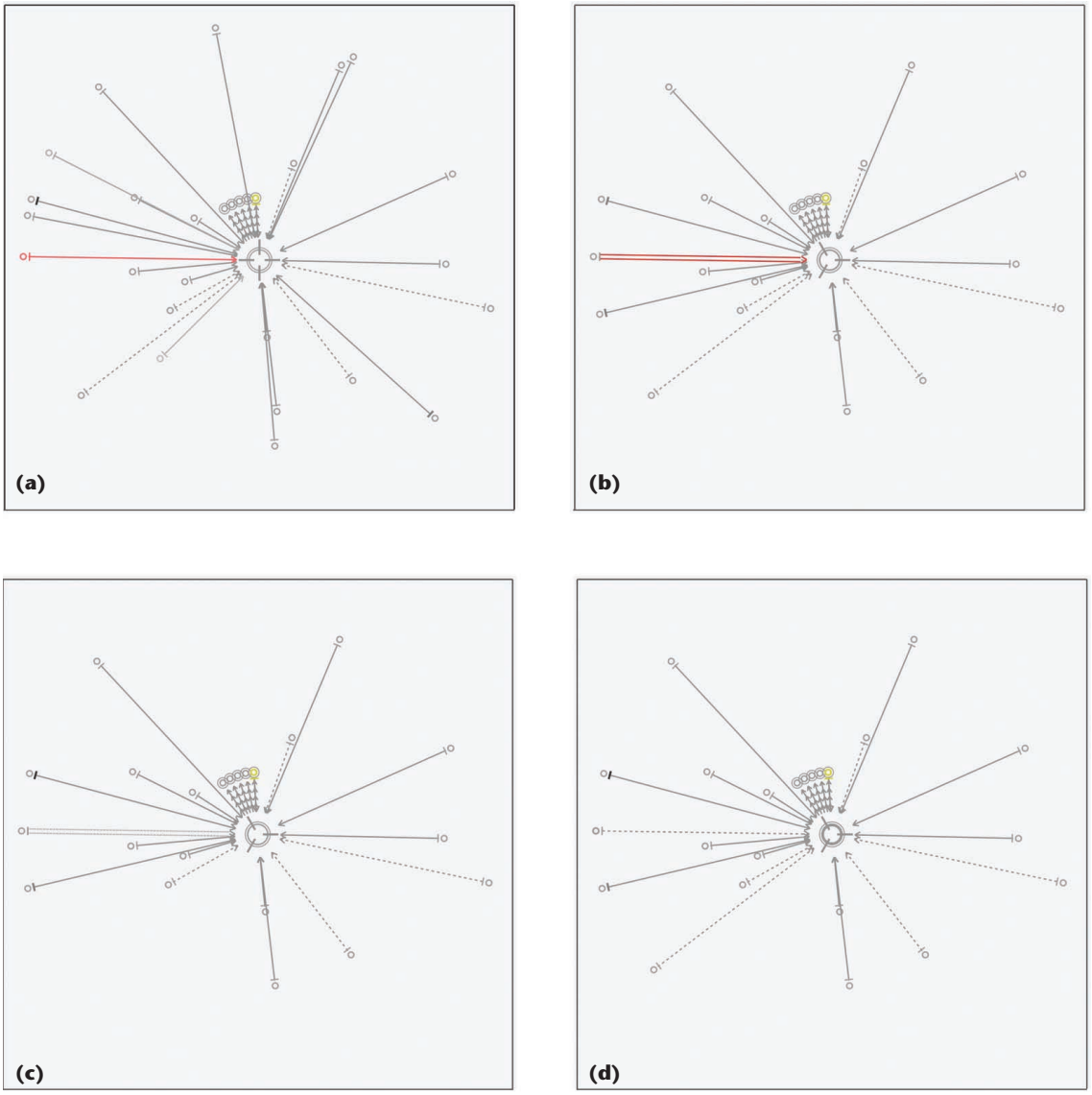
7 Image sequence showing login-based attack as a user attempts different user name and password pairs. The sequence shows multiple port-sentry attacks (in bright red) from different hosts in rapid succession.

which indicates the offender is attempting to connect with several protocols and user name and password pairs. This is common if the offender believes there may be a weak account on the system. Many operating systems, particularly older versions, were shipped with poorly protected standard accounts. These accounts provide a common backdoor to many systems and are still attacked today. More specifically, the attack begins with a port scan (see Figure 8a) and proceeds to an attempted telnet connection that fails (see Figure 8b). The offender then attempts an FTP connection in Figure 8c, which succeeds in Figure 8d, likely because of a built-in FTP account. The fact that these events occur from the same host in the middle of the night adds to the information presented to the system administrator. While such individual events aren't uncommon and normally aren't indicative of an attack, the sequence of events together raises serious concerns.

Conclusions and future work

By providing sufficient attribute mappings within a visualization, we can represent substantial characteristics as to the overall behavior of users within the environment. By analyzing user behavior as a whole, we can gain insight into the user's intent and ultimate goals. By focusing analysis on the user's behavior, we reduce the number of false alarms and increase the reliability of the systems administrator's analysis. Ultimately, incorporating visualization tools should greatly improve the detection of intrusions and misuses before organizations incur system damage.

Ultimately, visualization tools will become much more than just an early warning system for system administrators, rather they'll become a filtering device letting a system administrator filter out unwanted details and identify real activity of concern. The ability for visualization to harness the power of the human visu-



8 **Image sequence showing port-scan-based attack. The attacking system is on the left side of each image on the horizontal plane to the monitored system. This sequence shows an organized attack. (a) An initial port-sentry scan by a remote host. (b) The same host failing to successfully pass the authentication challenge posed by the server for the telnet protocol. (c) The system is now attempting to connect through FTP. (d) The successful connection by the same host through FTP, likely because of a built-in account with weak protections.**

al system and intuition of the administrator lets visualization provide a tool capable of assisting in detection and identification of intrusions and misuse that would not otherwise be detectable. This is particularly important as hackers adapt and evolve their techniques to counter new detection techniques.

The ultimate impact of this research will be the empowerment of system administrators, fewer successful intrusions and misuses of computer systems and networks, and reduced time requirements on the part of the system administrator to do the analysis.

The biggest missing feature of the environment right now is the ability to pick nodes and connections for additional information. This interactivity will enhance the administrator's ability to track activity over time and identify culprits for appropriate action. We must explore the data collection issues on additional systems. Every operating system has different information available and stores this information in different places on the system. We must expand our data collection capabilities to allow for the collection of data on all systems easily; Microsoft Windows systems will be particularly challenging because the log information provided by such systems don't conform to the style or content as Unix-based systems do. In fact, much of the data and the tools we rely on are simply unavailable in a recognizable form, if at all. Finally, the environment currently operates in a postmortem fashion. Extensions to fully support real-time analysis are critical. ■

2002 CG&A Editorial Calendar

January/February: Information Visualization

Computer-based information visualization has emerged as a distinct field centered around helping people explore or explain data by designing software that exploits the properties of the human visual system. New methodologies and techniques are critical for helping people keep pace with the torrents of data.

March/April: Image-Based Modeling, Rendering, and Lighting

Despite its recent arrival on the scene, the field of image-based modeling and rendering has already established itself as an important tool for a wide range of computer graphics applications. Image-based techniques use real-world digital photographs to synthesize novel imagery, letting us creatively explore and reinterpret realistic geometry, surface properties, and illumination. It has already experienced great successes ranging from selling real estate on the Web to amazing visual effects in film.

May/June: Graphics in Advanced Computer-Aided Design

The use of computers in the design and manufacturing processes has come a long way from the first CAD systems in the automobile and aerospace industries, with the huge mainframes and enormously expensive displays. Current CAD systems exploit innovative uses of the new technologies that help to move ideas from concept to model to prototype to product.

July/August: Virtual Worlds, Real Sounds

We only need to close our eyes for a moment to experience the amazing variety of information that our ears provide, often more quickly and richly than any other sense. Using real sounds in virtual worlds involves parametric computation; synthesis; and rendering sound for VR, entertainment, and user interfaces.

September/October: Computer Graphics Art History and Archaeology

Archaeologists can use computer graphics techniques to reconstruct and visualize archaeological data of a site that might otherwise be difficult to appreciate, with applications in analysis, teaching, and preservation. Similarly, art historians use computer graphics to analyze, study, and preserve great works of art, which may be too fragile or too valuable to touch or move.

November/December: Tracking

High-resolution tracking of user position and orientation (head, hand, feet, and so on) is increasingly a critical issue for virtual reality, augmented reality, modeling and simulation, and animation. Current tracking hardware is based on a variety of sensors including magnetic, optical, inertial, acoustic, and mechanical (as well as hybrid combinations).

IEEE Computer Graphics and Applications

References

1. D. Polla et al., "A Framework for Cooperative Intrusion Detection," *Proc. 21st Nat'l Information Systems Security Conf.*, Nat'l Inst. of Standards and Technology, Washington, D.C., 1998, pp. 361-373.
2. D. Zerkle et al. "A Data-Mining Analysis of RTID Alarms," *Recent Advances in Intrusion Detection*, Elsevier, Netherlands, 1999.

Robert F. Erbacher *is an assistant professor in the Department of Computer Science at the University at Albany-SUNY. His research interests include computer graphics, information and scientific visualization, virtual reality, interactive computational steering, and concurrent processing. He has a BS in computer science from the University of Lowell and an MS and ScD in computer science from the University of Massachusetts-Lowell. He is an associate editor for the* Journal of Electronic Imaging*, chairs the SPIE Conference on Visual Data Exploration and Analysis, is on the IEEE Visualization conference committee, and is on the International Association of Science and Technology for Development technical committee on computer vision.*

Kenneth L. Walker *is a senior computer applications programmer at the New York State Unified Court System, Division of Technology. He has a BS in computer science and applied mathematics and an MS in computer science from the University at Albany-SUNY.*

Deborah A. Frincke *is a faculty member at the University of Idaho and codirector and cofounder of the Center for Secure and Dependable Software. She is also cofounder of TriGeo Network Security. Her primary research interests involve intrusion detection and enterprise-wide system defense. Her recent work is on aspects of collaboration between remote sites and investigatory techniques for cyber attacks that cross enterprise boundaries. She has a PhD in computer science from the University of California, Davis. She is a member of the* Journal of Computer Security *editorial board and was program chair for the First Workshop on Intrusion Detection for the Second International Workshop on Recent Advances in Intrusion Detection.*

Readers may contact Robert F. Erbacher at the Univ. at Albany – SUNY, Dept. of Computer Science, LI 67A, 1400 Washington Ave., Albany, NY 12222, email erbacher@cs.albany.edu.