;login:

THE MAGAZINE OF USENIX & SAGE

April 2001 • volume 26 • number 2

inside:

CONFERENCE REPORTS

14th Systems Administration Conference (LISA 2000)

USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

OUR THANKS TO THE SUMMARIZERS:

Lee Amatangelo
Brian Baggett
Dave Bianchi
Paul Federighi
Jim Flanagan
Steve Hanson
Dave Homoney
Tony Katz
Brendan Kelliher
Brian Kirouac
Vinod Kutty
Eric Lakin
Dave McFerren
John Ouellette
Socrates Pichardo
Nicholas Schrieber
Sam Shaffer
Josh Simon
Doug Stewart
Theo van Dinter
Craig Vershon
Steve Wormley

Photographs taken at LISA 2000 can be found at

<http://www.usenix.org/publications/library/proceedings/lisa2000/confpix/pixindex.html>

conference reports

14th Systems Administration Conference (LISA 2000)

New Orleans, Louisiana December 3–8, 2000

ANNOUNCEMENTS

Summarized by Josh Simon

The first session started with the traditional announcements from the program chairs, Phil Scarr and Rémy Evard. Phil began with the following:

- There are over 1800 attendees, making this our biggest LISA conference ever.
- There were 85 papers submitted; we accepted 36 (42%).
- The Proceedings ran 378 pages.
- Over 1,800 messages were sent to and from the program chairs to plan the conference.
- Sixty-three percent of the program committee changed jobs between LISA 1999 and LISA 2000.

Thanks go to the program committee, invited talks chairs, network track coordinators, security track coordinators, guru coordinator, WIPs coordinator, the paper readers, Ellie Young and the USENIX staff, Rob Kolstad for his work in typesetting the Proceedings, Lynda McGinley for the terminal room, and all the vendors mentioned in the conference directory.

Rémy Evard then announced the best paper awards:

System Honorable Mention—"Deployme: Tellme's Package Management and Deployment," by Kyle Oppenheim and Patrick McCormick of Tellme Networks.

Best Papers (tie)—"Peep (The Network Auralizer): Monitoring Your Network with Sound," by Michael Gilfix and Alva Couch of Tufts University; and "Tracing Anonymous Packets to Their Approximate Source," by Hal Burch at Carnegie-Mellon University and Bill Cheswick of Lumeta Corporation. Both papers had a student co-author.

Dan Geer, USENIX president, announced the proposed split of USENIX and SAGE, covered elsewhere in this issue. Barb Dijker, SAGE president, echoed Dan's announcement, reiterated the desire for member feedback, and announced the 2000 SAGE Award for outstanding achievement: Celeste Stokely, for her work in collecting and distributing systems administration information for over 10 years.

Keynote Address

THE WORLDWIDE SYNDICATE

J.D. "Illiad" Frazer, User Friendly

Summarized by Jim Flanagan

The artist behind the immensely popular Web cartoon strip "User Friendly" (*<http://www.userfriendly.com>*) related how he ventured into self-syndicating his work, and what he has learned about the traditional syndication model along the way. "User Friendly" recently celebrated its third anniversary and has experienced explosive growth in Web page impressions.

Cartoonists are curious about human nature, and the traditional syndication process causes these people no end of pain by imposing on them a fundamental separation from their audience, which is treated as a market rather than a community. Each layer between cartoonist and audience removes revenue along with artistic control over content. J.D. told how Berke Brethed stopped doing his popular comic, "Bloom County," not because he ran out of material or burned out, but because some of the content offended Donald Trump, who in turn bought the syndicate and sat on the strip.

The syndicates can only do what they do because they control access to the market. The Web, notorious for its lack of control, allows cartoonists to "syndicate" their own work, retain creative control, and be more responsive to current events, free from the cumbersome review process. Not that all cartoonists need do is sporadically draw cartoons and post them on the Web. They have to be responsible for all of those things that the syndicate provides: creative and legal support and a revenue model.

By keeping artistic control the cartoonist gains direct access to an audience – a community rather than a market – and lots of interesting feedback. But this community also needs to be nurtured, lest things get out of control. For example, an April Fool's joke in which "User Friendly" claimed to have received a cease-and-desist letter from "a large, unnamed software company" (which was reported as fact on Slashdot by co-conspirator Rob "Cmdr. Taco" Malda) resulted in several higher-ups at Microsoft asking J.D. who in Microsoft had sent the letter. Apparently, the readership had simply assumed Microsoft was the culprit and flooded the phones with complaints, threats, and pledges to attack the company's IT infrastructure on Illiad's signal.

The audience also learned a little bit about what happens behind the scenes at "User Friendly," in that there is a largish organization which provides creative assistance, fiscal management, and other overhead tasks, forever dispelling my image of J.D. penning the strips alone in his basement at night and uploading them to the Web.

Questioners from the audience wanted to know if the UF crew had any plans to help other cartoonists get off the ground with self-syndication, and if UF had any intentions of becoming a syndicate themselves. UF is waiting for the right business model before they start involving other artists, and they would never themselves become a syndicate because it makes little sense to start placing layers between the artist and the audience. There are other Web-based cartoonist collectives getting started as well. Finally, we learned that J.D. also likes the Web cartoons "Goats," by Jonathan Rosenberg (<*http://www.goats.com*>), "Sluggy," by Pete Abrams (<*http://www.sluggy.com*>), and "SinFest," by Tatsuya Isheda (<*http://www.sinfest.net*>).

Invited Talks

How Not to Get Fleeced with Employee Stock Options

Jon Rochlis, The Rochlis Group, Inc.

Summarized by Theo van Dinter

Rochlis is not a professional financial planner, but he has a strong interest in the topic and is expecting a degree in financial planning by the end of 2000. The slides from LISA, and a good amount of other option-related information, can be found at <*http://www.rochlis.com/options/*>.

This talk covered a lot of ground: What is stock? What are stock options? What are the tax rules? Is a given stock option offer any good? When should the options be exercised? Unfortunately, Rochlis only got to cover about half of the information in the slides before running out of time. There is a lot of good information, including financial planning-related links, available on the Web site listed above.

The overall conclusions from the talk were:

- Try to get incentive stock options (ISOs) whenever possible.
- Stock option value is a percentage of company market capitalization, not directly related to the number of shares.
- Don't be emotionally attached to options/shares. Do research and sell if it looks like the stock will lose value. Shares can always be bought again at a lower price.
- Be aggressive with tax deductions. Don't lie, but claim everything possible.

SAGE Update

Barb Dijker, SAGE President

Summarized by Lee Amatangelo

Dijker provided a status report on SAGE for the past year, projects in progress, and future plans.

The largest topic looming over the entire LISA 2000 Conference, starting with a statement during the Opening Session from USENIX President Dan Geer, is that talks are underway to determine if the best course of action for both USENIX and SAGE is to have SAGE become a separate entity.

Perhaps the second biggest issue discussed during this update and at other SAGE meetings during the conference was SAGE's Certification Project undertaken as a way to elevate system/network administration to the status of a profession. But getting there is going to take time. Along those lines, SAGE has already done a fair amount toward defining the various levels of system/network/database administration work.

Several student internship programs and interns present at the LISA 2000 Conferences were recognized. The SAGE board is very much in favor of supporting and promoting more internships, and the audience was encouraged to create new internship programs for system/network administrators.

There was also a call put out for mentors. SAGE encourages additional senior members of our profession to help others grow in this profession in the spirit of the whole open source mentality. Those interested in mentoring or being mentored were asked to contact SAGE.

Ideas were presented and solicited on how SAGE can market itself. Marketing will now become even more important if SAGE is to become a separate entity.

Other major highpoints of the update included the SAGE salary survey results. The results are displayed in many ways based on various criteria. Salaries can be looked at based on gender, years of experience, geographical location, educational level, certifications achieved, and number of systems (and operating systems) supported, to name the biggies.

Following the formal update presentation, the floor was opened for questions. Collective Technologies' Jeff Tyler said that he had not heard a compelling reason from either side as to whether SAGE should stay with USENIX or break off as a separate entity. One response from the board and other members of the audience was that SAGE and USENIX have two very different charters. However, regardless of whether the split happens or not, the board members of both USENIX and SAGE assured everyone that there would still be ventures and events put on jointly by USENIX and SAGE and that a great working relationship between the two groups would be maintained.

The Digital House

Lorette Cheswick

Summarized by Lee Amatangelo

Lorette Cheswick, along with her not-so-famous husband, Bill (I think he may have worked at Bell Labs once), and a little help from their two children, Kestrel and Terence, presented their "Digital House," was very interesting, entertaining, and highly educational talk.

The Cheswicks have a talking house. Whenever the doorbell rings, a message goes out over the intercom system stating, "Someone is at the door." This message is also heard by the visitor and is particularly amusing to younger children.

Other ideas soon followed. Wouldn't it be nice to know when the mail had arrived? So a wireless motion detector is positioned in the back of the curbside mailbox. Whenever the mailbox door opens and shuts, a message goes out over the intercom system, "Honey, you've got mail!" (which was quickly changed due to its political incorrectness).

The intercom system also announces daily astronomical events such as the times for sunrise, sunset, iridium flares, comets, planet risings and settings, etc. In addition, the intercom provides stock quotes throughout the day.

Here are more details from their presentation:

The Cheswick digital house includes:

- Home Ethernet and wireless
- Platform for firewall-free experiments
- Caller ID to serial port
- X10, Linux sound card to home intercom

The Cheswicks have 11 computers on their home LAN right now.

The full Powerpoint presentation is located at: <*http://www.cheswick.com/*>, then select "Powerpoint slides for Lorette's *Digital House* presentation at the New Orleans LISA 2000." Or simply select <*http://www.cheswick.com/ppt/digital.ppt*>.

Experiences with Incident Response at Ohio State University

Steve Romig, OSU-IRT

Summarized by Brendan Kelliher

Before the foundation of the Incident Response Team at Ohio State University, OSU's computer security was conducted on an ad-hoc, part-time basis. In early 1996 Steve Romig, a campus system administrator, was asked to be part of a full time, professional systems security force which would become the OSU Incident Response Team.

In the summer of 1996 the OSU-IRT responded to a local ISP's complaints about attacks emanating from campus systems. A local hacker group, the LoTecs, was illegally accessing university systems and using them as launch points to attack other sites.

Steve traced the hackers back to the campus modem pool. He had to work with the local phone company (AmeriTech), which caused him much aggravation. When tracing a call in real time, he would be passed from operator to engineer, each trying to complete a separate part of the trace.

Some of the hacker's login sessions lasted for days. One lesson he learned is that phone traces can be done "after the fact." The phone company has logs of all calls and can do traces going back for months. This freed him up to concentrate on what systems and accounts the intruders were compromising.

By watching the logins of known hacked accounts, he noticed a pattern emerge in their sign-on/sign-off times. Just after the local schools let out he would see the first logins. Then around dinnertime he would see them logout for a brief period, then sign back on shortly after. His assumption that the intruders were local high school kids would later turn out to be correct.

Steve attended hacker meetings along with a local undercover police officer. They learned the identities of several of the hacker group members, one of whom had a history of traffic violations.

In October of 1996 the hackers started to attack ".mil" sites, which got the attention of the FBI. Steve's information was used to build the government case. One interesting technique the hackers used was to personally respond to complaints from a victim they had hacked. In September of 1997, nine members of the LoTecs were served with warrants.

Here are some recommended steps you can take to guard against breaches in security:

- Determine what measures your company is willing to take against hackers.
- Create an Incident Response Team.
- Log everything!
- Make contact with your local authorities; many police departments have established “white collar” crime units which also cover computer crimes.
- Work out security procedures and emergency contacts with your telco or Internet access provider.
- Never audit your own security.
- Be patient when dealing with LEOs (law enforcement officers).
- Document all your steps when tracking an intruder; your notes will be crucial if the case ever goes to court.

INTEGRATING LDAP INTO A HETEROGENEOUS ENVIRONMENT

Leif Hedstrom, Netscape

Summarized by Jim Flanagan

Hedstrom started this talk with a brief overview of what LDAP is and is not. LDAP is a client-server protocol with its roots in the OSI directory standard X.500. LDAP is not a database but merely the protocol for transmitting and formatting directory services information on the network. An LDAP entry consists of one or more attribute-value pairs, and can have multiple values associated with a single attribute. The objectclass attribute defines what attributes an entry may legally have and provides a limited object inheritance. The standard schemas which ship with most LDAP directory servers can be extended but you should try to stick with the supplied schemas if possible.

Each entry has a globally unique identifier called the Distinguished Name, or DN, which is a path along a tree structure containing nodes like country, organization, department, and name. The DN is public information, so it should not contain privileged information such as a person’s SSN. A Relative DN (RDN) is a shorter part of the DN which is locally unique (such as within a company); an example is a userid or uid. When designing a Directory Information Tree (DIT), try to keep it as flat as is reasonable, because later changes will be easier to accommodate.

LDAP will make your life easier, but you will have to treat it like any other enterprise infrastructure element, and make it robust. LDAP can be used to integrate several information sources (HR, facilities, NIS, NT Domain, email, mailing lists), but you will need to get buy-in from all the interested parties.

One advantage to implementing LDAP is that you can assign ownership of different slices of the data to various groups, and delegate management of the data to those groups. This reduces load on the help desk. Different LDAP vendors implementations handle access control lists (ACLs) differently (as it is not part of the LDAP spec). The iPlanet LDAP server, for example, provides very powerful ACL mechanism and default behaviors (such as automatically giving the owner of a mailing list “ownership” to that entry).

LDAP can be used to replace or back end NIS for UNIX user information. Solaris 8 supports LDAP in the /etc/nsswitch file out of the box. There are scripts available to migrate your NIS databases to LDAP, and the schema is described in RFC 2307. On UNIX implementations which support Pluggable Authentication Modules (PAM), there are modules which allow direct LDAP binding over SSL (no passwords over the Net in the clear).

When you have disparate data sources, you may want to investigate one of the commercially available LDAP metadirectory solutions. These allow for bi-directional synchronization of data from multiple LDAP directories and other legacy sources (via “connectors”) automatically, and are designed to resolve namespace conflicts. Depending on the complexity of your problem, you might be able to build your own metadirectory, but most of the time you will have to buy one. There are not currently any open source metadirectory systems. Most commercial metadirectories are extensible, allowing you to write your own connectors.

Another approach is to gateway legacy data sources into LDAP. A common example of this is ypldapd, which allows you to store your NIS maps in LDAP but uses traditional tools and clients to access it. This tactic should only be used as a transitional tool rather than a solution.

The change log is used for server replication and can be a nice hook into your directory server to accomplish some metadirectory functionality. The change log is data in the LDAP directory itself, and is protected by ACLs, but in an all-or-nothing fashion, giving a possible exposure of data that is otherwise protected with more granularity in the directory. Keep the change log protected. The change log can also be used for disaster recovery if you back it up.

Exporting LDAP data can accomplish a “poor-man’s metadirectory.” You can use scripts to massage LDAP-exported data into NIS maps, DNS zone files, or whatever. It’s easy and it’s fun!

NT presents special difficulties to LDAP deployment, especially because Active Directory wants to be in control. Microsoft also made some poor decisions regarding RDN and uses a proprietary password encryption scheme, forcing the use of plaintext passwords. Having a single namespace for UNIX and NT users will help you avoid various problems.

MAPPING CORPORATE INTRANETS AND INTERNETS

Bill Cheswick, Lumeta Corporation

Summarized by Steve Hanson

Many of us often wonder what it would be like to leave our safe jobs in the corporate world to start a company. Part of Bill Cheswick's talk addressed this since he has recently spun off Lumeta from Lucent, starting a company to map corporate intranets as a service.

Most of the talk, however, was about the work Lumeta is doing. Some of this work developed out of infrastructure protection done for the government. Most intranets are completely out of control, and nobody really knows anymore what is on their networks. Some preliminary work is being done on mapping intranets and making simple visualizations of the systems. Lumeta's work was compared to some other projects in these areas, such as MIDS and CAIDA. Different visual representations of the intranets and the Internet have been done. Lumeta also did some work for the government during the Yugoslavian crisis, mapping the Internet in Yugoslavia, which gave good indications of what portions of the country had had their power knocked out by bombing.

One of the most interesting aspects of the talk was the maps themselves. Some of them are quite beautiful, and some have been purchased by museums as artworks. Of course, this information is being used primarily in the research community, and Lumeta hopes a large commercial market for their services will develop.

Maps: <*http://www.peacockmaps.com*>
Information: <*http://www.lumeta.com*>

The Design and Implementation of Highly Scalable Email Systems

Brad Knowles, Belgacom Skynet SA/NV

Summarized by Brian Baggett

Knowles, former email admin for AOL, gave a great talk on the problems faced in developing and running large-scale email systems. His talk focused less on implementation and more on fundamentals and architecture.

The approach taken by academia is often different from that of commercial institutions. Academia often reinvents the wheel and does things on a small scale, but occasionally it does something revolutionary. By contrast, the commercial world has no problem buying solutions; the time it takes to bring a product to market is crucial, and so the process tends to be more evolutionary than revolutionary. Most of their revolutions focused on scaling.

The underlying problems with all of the potential solutions are that none of them scale to handle 1 million-plus users on their own. Eliminating single points of failure or getting away from inefficient technologies like NFS has proven too difficult for many. Knowles summarized the pros and cons of POP3 and IMAP and identified the big bottleneck that hampers mail server scaling (I/O). This was a highly informative talk chock full of interesting facts and data, the results of which can be found at *http://www.usenix.org/events/lisa2000/invitedtalks/knowles.pdf*.

SANs — From A to Reality

W. Curtis Preston, Collective Technologies

Summarized by Steve Hanson

SAN systems are becoming a fact of life in most production environments, but it is not always clear if the SAN systems being purchased are serving a purpose or are just being bought because they are the new and hot technology.

Preston's presentation on SAN systems discussed the different technologies involved in SANs, and how to decide if SAN technology is a good fit for your needs.

He described the three competing distributed storage technologies – WAN, NAS, and SAN, and then went on to discuss the different hardware configurations available for SAN. The advantages of SAN (easy allocation, sharing, backup, etc.) were covered, as was its cost efficiency.

Finally, this talk discussed some of the current pitfalls of SAN (need to test thoroughly, incompatibility between different hardware), and how to decide if SAN is for you.

Why the Documentation Sucks, and What You Can Do About It

Steven Levine, SGI

Summarized by Josh Simon

Levine spoke and sang about documentation. Steven, a technical writer, talked about four major subjects: myths, difficulties, projects, and improvements.

First, Steven talked about some myths. One is that writers are editors. In reality, writers not only edit stuff others (such as developers) write, but write original content, maintain other existing documents, and produce both hardcopy and online help and Web-based documentation. They also coordinate and organize and are detectives; they have multiple information sources and work with various groups or departments. They are also responsible for documentation consistency and legal issues.

He then discussed some of the difficulties that writers face. He started this section with a song, and yes, all 300-plus audience members were singing along with the chorus. The major difficulties are lack of resources, conflicting perspectives, little (if any) usability testing of the documentation, shorter release cycles, distinctions in hardware and software, the problems of writing from experience on as-yet-nonexistent products, and having to rely on developers' time and interest to improve the documentation.

Third, he discussed some of the typical projects that writers are involved in. They are responsible for not only administrative documentation but also online documents, procedures and examples, and man pages (which may not be sexy but are certainly very useful).

Fourth, Steven discussed how to improve the documentation. The short answer is it's a two-way street. If you see something needing work in a document, let the author know. There's always some form of contact information (even if it's postal mail). Document what you want solved. Document what you did to work around a problem to help others not have to go through it themselves. Formalize your informal people networks; if you're a developer, take your writer to lunch. Produce libraries of examples, procedures, tricks you use, and so on. Collaborate within the company across department lines. Collaborate with friends and acquaintances at other companies.

REFEREED PAPERS

SESSION: DEEP THOUGHTS

Summarized by Socrates Pichardo

THEORETICAL SYSTEM ADMINISTRATION

Mark Burgess, Oslo College

Mark attempts to demonstrate that system administration can be modeled in certain instances and that this theoretical model can be built and used to further understand system administration variables and their interdependencies. This is a very important concept; once you can identify all relevant parameters and model a particular problem you can proceed to optimize these variables to achieve maximum results.

Mark started his presentation by defining a simplified system administration model in which users, OSes, resources, policies, and states are the main components.

Mark's group has concluded that system administration problems fall into these categories:

- **Random behaviors, or type I models:** Variables follow random patterns within a well-defined cause-effect relation. Things like averages, median, and statistical theory can be used to optimize these models. Most stability problems will follow this behavior.
- **Anthropological behaviors, or type II models:** Variables follow anthropological behaviors within certain boundaries. Game theory and human conduct analysis can be used to optimize these models. Most utilization problems will follow this behavior.

In other words, influences on the systems can thus be classified as either random, stochastic, or passive (type I), or as intentional, adversarial, or strategic (type II), depending on the significance of the change.

AN EXPECTANT CHAT ABOUT SCRIPT MATURITY

Alva L. Couch, Tufts University

Couch presents his solution to current scripting tools and language limitations on solving complex system management tasks. To circumvent current limitations, he is developing an "interpreter" called Babble, which will take XML-like directives and mark up tags and interpret them into desired configuration tasks.

To accomplish this, he has created his own set of directives called Stream-Structure Markup Language or SSML, based on the "Jackson System Design" from the punch card era. Jackson's Principle claimed that the way to properly design a program for processing punched card stacks is to link the structure of the program with the structure of the stack that it processes. Alva expanded this principle into: "The structure of a fully functional interactive script is exactly parallel to the branching and looping structure of the device interactions in which it must engage."

SSML and Babble introduce a new level of instrumentation that will help system administrators tackle complex administration tasks but at the expense of extremely limited functionality and low reusability. Right now, there are only limited sets of complex tasks that Babble can address with success, and SSML scripts are very dependent on a particular revision version of the devices being managed. Each device revision requires a completely independent Babble script.

AN IMPROVED APPROACH FOR GENERATING CONFIGURATION FILES FROM A DATABASE

Jon Finke, Rensselaer Polytechnic Institute

Much of Rensselaer's site configuration information is stored in a relational database. In the past lots of little custom C programs and scripts were needed to extract this information in the appropriated format for server and daemons. This approach proved to be difficult to maintain and expand.

Maintenance was cumbersome due to the variety of scripts techniques and programming styles found at the site. Expansion into new operating systems was time-consuming since all these scripts needed to be "ported" to new hardware and operating system revisions.

Now, Rensselaer is using a centralized approach to this problem. They have gathered all logic and intelligence needed for configuration file generation into their Oracle database by using PL/SQL as their scripting tool. They generate all their configuration files in the database itself, then write a short program to dump the "files" out of the database and into the file system of the target machine. This allows all of the file generation code to be stored and maintained in the central database, using a consistent set of tools. In addition, the program to copy files could be generic, and once built for a particular operating environment, could be used for any of the files they might generate for that system.

Rensselaer's solution proves that centralization, when done right, can have a significant positive impact on the overall manageability of an IT infrastructure. Their usage of commercial tools (i.e., Oracle PL/SQL) makes their tools and scripts a bit more difficult to implement. At the end of his presentation, Jon made a "call for help" for anyone interested in "porting" their scripts into any of the open source products (i.e., MySQL).

SESSION: YOU, A ROCK, AND A HARD PLACE

Summarized by Craig Vershon

FOKSTRAUT and Samba — Dealing with Authentication and Performance Issues on a Large-Scale Samba Service

Robert Beck and Steve Holstead, University of Alberta

Robert and Steve noticed a performance problem with the Samba server that was being used as a gateway to AFS. The system was getting an unanticipated new load that couldn't handle all the users' authentication.

The Samba server was running as an AFS client gateway to things like Windows clients. They found they had to run Samba with clear text passwords enabled with password crack, but found an issue with some Windows clients sending passwords in all caps. This would allow them to authenticate to Samba but not to Kerberos, which requires more varied passwords.

Once they implemented the server, they found it to be highly CPU-bound. The server was receiving repeated password failures. They found a pattern: three bogus attempts, then the real password was sent. Windows was sending the "windows" password instead.

The solution: FOKSTRAUT, patches for Samba to make a DBM password cache. First, it caches the password that failed. It stores this in a database and keeps a failure count. After three failures, it checks again and resets to zero after success. Then they cached the "corrected case" success. This was stored in a clear text database. They found this "evil" and unsecure, but a compromise had to be made somewhere.

Available at:
<ftp://sunsite.ualberta.ca/pub/local/people/beck/fokstraut>

Improving Availability in VERITAS Environments

Karl Larson, Tellme Networks; Todd Stansell, Certainty Solutions

Karl and Todd spoke of problems and solutions they found in their VERITAS implementations.

Some of the tools they created or used:

- vxstat2gnuplot: this program converts the output of vxstat into a useable format for gnuplot. I thought this was really useful to see a graphic version of the disk/volume usage.
- cricket: used for trends-based monitoring *<http://cricket.sourceforge.net>*
- save-vxlayout: this script saves VM config details; good in use for disaster recovery info
- synch: from EMC; retrieves Symmetrics internal configuration details
- emcprints: shows all back-end controllers, devices, etc.; adds this info to the vxprint output.

They found problems with millions of small files on a single file system. The metadata becomes a performance bottleneck with VxFS. It uses metadata for intent logs (journaling). Changing millions of files causes changes to much more than just the files themselves. By default all the metadata is stored at the beginning of the file system. The space reserved could be too small. They found that by using Quicklog, they could move and store the metadata on a separate device.

Running backups on millions of sequential files daily makes it hard to obtain consistent "point in time" backups. They came up with two fixes: Volume Manager snapshots and using file system snapshots.

Cool tips and tricks for VERITAS: *<http://www.vxideas.org>*.

Designing a Data Center Instrumentation System

Robert Drzyzgula, Federal Reserve Board

Drzyzgula outlined the context of his problems:

- Several conflicting production cycles
- Tight deadlines
- Enormous economic models
- Highly variable capacity requirements
- Capacity is higher priority than reliability

Drzyzgula seems to have a limited budget with which to work. They purchase many parts in bulk and assemble systems themselves. He has been working on designing and building a monitoring and controlling device, to be used on all his various systems in the data center. The goals were to be able to monitor such things as power usage and temperatures and to be able to control power cycles, console access, etc. He spoke about the various chips, sensors, and other hardware he was using to attempt to implement this. He has been able to get a small working prototype to work in a limited environment.

SESSION: USERS AND PASSWORDS AND SCRIPTS, OH MY!

Summarized by Sam Shaffer

User-Centric Account Management with Heterogeneous Password Changing

Douglas Hughes, Auburn University

Hughes details development of a Web-based tool to allow students to change their UNIX and/or NT passwords. (NT authentication is via Samba.) The "User-Centric" in the title indicates that the system was developed with inexperienced students in mind.

The paper lists five similar systems and their principal parameters. Douglas indicated that a lot of information which might have made the other systems valuable was not available to him. He is looking for someone to maintain the code, which is available at <*http://www.eng.auburn.edu/~doug/second.html*>.

Pelendur, Steward of the Sysadmin

Matt Curtin, Interhack Corp.; Sandy Farrar and Tami King, Ohio State University

This paper documents what seems to be a rather thoroughly automated account management system in use at Ohio State University.

Bottom line, this system modifies student and other accounts in response to changes in the university class database. Implementing it has greatly reduced the amount of time required to add and delete about 15,000 accounts per school term. The system is not available to the public because it isn't ready for prime time. As such, it is effectively "yet another set of design guidelines for an account automation tool" and provides some specifics of the implementation that might allow someone else to design and implement a similar system again.

Network Information Management and Distribution in a Heterogeneous and Decentralized Enterprise Environment

Alexander D. Kent and James R. Clifford, Los Alamos National Laboratory

The paper presents another unique set of circumstances which had to be incorporated into a user-friendly tool to allow people to manage their own data (such items as email aliases and email server passwords). There are several interesting components of this system, though. One is that changes to the database cause notification to an "event trigger daemon," which uses inetd to induce an update to an LDAP server.

The system source may be available. The paper indicates that U.S. encryption export issues require that the source be controlled. Make requests to the authors for information on requirements.

SESSION: THE TOOLSHED

Summarized by Jim Flanagan

xps: Dynamic Process Tree Watching Under X

Rocky Bernstein, Breakaway Solutions

In a talk as dynamic as the topic matter, Rocky described xps, which provides a view of the process table laid out as a tree rooted at the init, with colors distinguishing processes by owner or by state (running or waiting on I/O). What processes are shown can be determined by user-specified filters, and clicking on the process names can run a user-specified program such as ps to get specific information, or lsof to get the files open by that process.

The current version of xps is written using the Motif toolkit, but work is in progress to port it to GNOME. Much of the talk was spent weighing the various virtues and drawbacks of GNOME versus Motif, and how the GNOME view of the world changes how the xps problem is attacked and how one goes about optimizing for efficiency. GNOME also has tools which make life easier, such as Glade which builds dialog boxes that are much prettier than Motif's.

Rocky then demonstrated an instance of where xps might give more insight than traditional tools like ps or top. He went to a source directory with a fairly complex make procedure, and typed make. The make process showed up in the xps display, and alongside (or "under") that, you could see all the subsidiary awk commands and other commands being spawned to do the work and disappearing dynamically.

One questioner asked about the poll-based nature of xps having to read through the entire process table each refresh, and if it would be more efficient to somehow detect or trap calls to fork and vfork. Rocky said that it wasn't clear how to go about that, and that if you could, you might introduce problems with modifying the state of a program rather than simply monitoring it.

Extending UNIX System Logging with SHARP

Matthew Bing and Carl Erickson, Grand Valley State University

As grad student admins of 30 machines, the authors were being overwhelmed by having to try to winnow the interesting data from the volume of syslog data that was being generated. After giving a brief overview of how syslog works, Matt went on to present a list of areas where syslog could be improved.

- The routing of syslog messages depends only on their priority, which is the combination of facility plus a level, which is not very flexible. You can't add facilities to syslog.
- There is no standard message structure after the timestamp and pid.
- The priority information is not written to the logfile, and so is lost.
- In a centralized logging architecture, the timestamps are those of the originating system, and if the clocks on those systems are not in sync, event correlation is not feasible.
- It is not possible to detect if the logfile has been tampered with.
- UDP. Say no more.

Log-watching systems like swatch or logwatch don't completely compensate for these defects in that they don't provide realtime analysis or maintain state between runs, so they can't detect recurring problems and change their behavior (like stop paging you for the same problem).

SHARP (Syslog Heuristic Analysis & Response Program) is the author's response to these problems. Rather than replace syslogd, SHARP provides a daemon (sharpd) which also receives a copy of every syslog message. Modules compiled against the SHARP library can then register to get messages of various priorities. The messages which the modules receive are timestamped by sharpd for event correlation. The modules can then do anything with the message: put it in a file, alert a user, or bounce the message back to syslog at a different priority.

Examples of modules were:

- Mark: expects a message from each machine at a certain interval, and logs a high priority message if it doesn't get it
- UserAlert: learns about users' patterns of logins (time and location) and notices behavior changes
- ProblemAlert: after a number of repeated messages of a certain priority, they will start getting sent back through syslog at a higher priority.

While SHARP will work with syslog, the authors recommended nsyslog as a replacement to work with SHARP, as nsyslog preserves the priority of messages, uses TCP and SSL to prevent spoofing, and uses chained hashing to prevent modifications of the logfile.

Planned developments include a Perl interface for modules, access to global configuration across all modules, and making SHARP completely thread safe.

Peep (the Network Auralizer): Monitoring Your Network with Sound

Michael Gilfix, Tufts University

[Winner of the Best Student Paper Award]

The Peep tool is an experiment in leveraging innate human skills in distinguishing subtle deviations from the normal state. Peep uses a continuous, non-intrusive audio representation of the health of your network in terms of events, state, and heartbeat types of sounds. The sounds selected for use with Peep are all from nature (waterfalls, bugs, birds, etc.) since this was thought to be the least intrusive.

Peep is a departure from the current state-of-the-art monitoring tools in that, where current tools are visual, problem-centered, and provide negative reinforcement, Peep is aural, normalcy-centered, and provides positive feedback. Peep attempts to remain ambient to take advantage of unconscious processing, and the sounds are all mixed together so that combinations of sounds become significant. If your network sounds like it sounded yesterday, than everything is fine.

Peep has a Producer/Consumer architecture which supports either distributed or centralized configuration. It is UDP-based, uses auto-discovery by both clients and servers, and employs leasing to handle servers that go offline.

Michael gave a demonstration of Peep. First, he played discrete sounds such as bird chirps which corresponded to incoming and outgoing mail, bad DNS lookups, and telnet connections. Then there were some continuous sounds like running water and general forest insect noise, which represented load average and concurrent users, respectively. Then he played a sample of actual Peep output for low load, which sounded like being in the forest near a stream. After that he played a high load average sample, and while I didn't feel like it was quite time to start filling sandbags, it seemed a little less comfortable.

Someone from the audience commented that for rare events, you might forget what sound went with what event. Michael said that they were looking for a solution to that, such as a GUI quick reference utility. Another questioner asked what the overhead was on the server side. Memory is really the biggest bottleneck, as all sounds are loaded into memory. Most of the processing overhead is in the sound mixing. In trials, they were only able to drive the server load average up to 0.6.

SESSION: 1984

Summarized by Socrates Pichardo

Thresh – A Data-Directed SNMP Threshold Poller

John Sellens, Certainty Solutions

Thresh is a simple SNMP monitor tool that lives in between realtime alert monitoring systems (i.e., Big Brother) and trend analysis and history tools (i.e., Cricket). The power of this tool lies in its simplicity. Thresh is an elementary but elegant implementation of SNMP monitoring services with an emphasis on easy configuration, low system overhead, decent notification, and some basic history and logging facilities.

In spite of Thresh's low system overhead, it has some scalability issues. Its main constraints are the lack of parallelism, configuration complexity, and notification throughput.

If you need some basic SNMP monitoring without all the hassle of configuring and maintaining a feature-rich SNMP console, then Thresh could be the tool for you; otherwise stick to some of the more capable SNMP consoles available today.

eEMU: A Practical Tool and Language for System Monitoring and Event Management

Jarra Voleynik, eEMUconcept Pty Ltd.

Voleynik described eEMU as a monitoring and management event console. eEMU is a client-server system that provides for rapid development of monitoring agents. Beside its console capabilities, eEMU has a scripting language that takes advantage of heuristic algorithms implemented at the server.

One of the things that sets this tool apart from the market leaders (i.e., TNG, Patrol, OpenView, and Tivoli) is the way it handles, aggregates, and presents alarms. While other console solutions will rely on color code representations and multiple windows of information, eEMU uses textual messages for each event in a simple intuitive interface called eEMU browser. By default, the eEMU browser display only resources in "alarm state."

For its implementation, eEMUconcept Ltd. has decided to write its own eEMU agents and have them communicating with eEMU servers by using the eEMU protocol. eEMU works on the premise that all status information is handled by the eEMU server; therefore eEMU agents are simple scripts or programs that use the emsg program to send messages to the server.

One debatable design characteristic is their usage of TCP and not UDP for client-server communication. By using TCP and not transmitting "Systems OK" messages, they can avoid the common "UDP" storms generated by SNMP consoles and their polling efforts. On the other hand, we can argue that the overhead generated by TCP connections as well as the lack of "Systems OK" messages could produce some challenging programming problems for the developers and minimize their utilization gains. On their labs, they have been able to monitor 100 systems on a 33Kbps dialup line or 1,000 messages a minute on a 400MHz Pentium PC.

The power of the eEMU messaging language can be easily illustrated on the eEMU agents. eEMU agents with a few lines of code can handle complex monitoring scenarios.

Finally, eEMU has been successfully integrated with some of the major monitoring software vendors. This integration can be accomplished by using eEMU action scripts as well as other scripting hooks to their event engine.

Aberrant Behavior Detection in Time Series for Network Service Monitoring

Jake D. Brutlag, Microsoft WebTV

Realtime monitoring of service networks can generate vast amounts of time series data. Open-source tools like RRDtool and Cricket can help you with collecting, storing, and visualizing this data but, for large networks, you still need a methodology or tool to help you identify failures and/or abnormal situations.

Microsoft's WebTV division was facing this problem, and the amount of data being generated was enough to distract their network administrators from the important issues facing their networks. Their solution was to integrate a model based on exponential smoothing and Holt-Winters forecasting into the Cricket/RRDtool architecture.

Their model takes into consideration the following characteristics of time series data:

- A trend over time
- A seasonal trend of cycle
- Seasonal variability
- Gradual evolution of regularities over time

The Aberrant Behavior Detection model unpacks into three pieces, each building on its predecessor:

- An algorithm for predicting the values of a time series one time step into the future
- A measure of deviation between the predicted values and the observed values
- A mechanism to decide if and when an observed value or sequence of observed values is "too deviant" from the predicted value(s)

This model was implemented by enhancing the RRDtool with five new "consolidation functions":

- **HWPREDICT:** an array of forecast computed by the Holt-Winters algorithm, one for each Primary Data Point (PDP)
- **SEASONAL:** an array of seasonal coefficients with length equal to the seasonal period
- **DEVPREDICT:** an array of deviation predictions
- **DEVSEASONAL:** an array of seasonal deviations
- **FAILURES:** an array of Boolean indicators

On the Cricket side, Cricket 1.1 already includes a new type of monitor-threshold specific for aberrant behavior detection. Combining these two tools, they were able to monitor and alert on aberrant behavior conditions.

SESSION: THE SORCERER'S APPRENTICE

Summarized by Vinod Kutty

PIKT: Problem Informant/Killer Tool

Robert Osterlund, University of Chicago

PIKT is a system configuration management tool, addressing problems that tools such as cfengine are designed for, but in a more general purpose way. It monitors and warns of system problems, and has features that allow it to take corrective action if needed. The focus is on managing large numbers of machines rather than on individual machines.

Sysadmins typically write custom scripts to address these issues, but problems of OS diversity, code robustness and maintainability, specificity to certain tasks, scheduling scripts, error logging, script and configuration file distribution, and so on, plague this approach.

PIKT is designed to solve a lot of these problems in a fairly platform-independent (i.e., UNIX-flavored) manner. At its core is an embedded scripting language and a configuration file pre-processor that can be used with languages other than the PIKT language. It also includes a scheduling system, a distribution mechanism (like rsync/rdist), and a remote process execution facility (like rsh/ssh).

The typical deployment involves a central "master" machine which controls "slaves" (i.e., clients). Configuration files are managed on the master, then run through a tool that pre-processes and installs files, pushes changes to slaves, executes remote commands, and so on. A scheduling daemon on each client runs alarm scripts to monitor various aspects of a system (e.g., disk usage, running processes, etc.), and a flexible logging system is provided. There is some client/server security implemented using secret-key host authentication.

Some use cases not directly related to monitoring include installing and managing non-PIKT scripts and configuration files (e.g., inetd.conf), document distribution, and managing security (by complementing security tools with logfile analysis, security configuration file maintenance, and so on).

Future work includes a full security audit, a standard library of configuration files, a rewrite of the PIKT script interpreter (possibly using embedded Perl or another language), improved message routing, and graphical interfaces for the piktc and alert management components.

Relieving the Burden of System Administration Through Support Automation

Allan Miller and Alex Donnini, Hands-Free Networks

Companies increasingly have to support a growing population of users with minimal application or other technical training. This in turn increases the burden on support organizations.

An automated support system can help avoid a crisis and improve the scalability of the support organization. However, automating support is difficult and often leads to a "mountain of kludges" that do not exhibit an understanding of the issues. Automation is best suited for repetitive tasks and touches upon all aspects of a system.

Traditional user support involves some kind of problem/ticket system with a database back end that stores solutions to previously encountered problems. At each step of the support process, human intervention is necessary to clarify end user symptoms, search existing symptom + resolution databases, escalate the request, and so on. This is a labor-intensive, error-prone process and often relies on mental knowledge rather than a database.

The automated support system under discussion uses a software client, instead of the user, to detect and report problems. A database is used to track symptoms and resolutions that include executable code (called "scrips," which are collections of various modules). Thus, the software client can automatically resolve the problem if there is a match.

The expectation is that about 80% of all problems can be solved this way, with the remaining 20% involving an escalation procedure. In addition, there is a well known 80-20 rule in support circles that suggests the size of the database that can solve 80% of all problems is expected to be about 20% of the size of the universe of solutions.

Experience with the system so far has been on Windows operating systems, with a Linux port in the works. Some beta sites are using the software, and feedback indicates a remarkable similarity in problems encountered and automatically solved, despite considerable differences in the businesses.

Additional uses envisioned for the system include automated administration and maintenance functions, security patch distribution, and automated support for mobile and embedded systems.

FTP Mirror Tracker: First Steps Towards URN

Alexei Novikov and Martin Hamilton, ITEP

The FTP Mirror Tracker package attempts to decrease the load of FTP traffic on WANs while improving performance for end users. It does this by localizing FTP accessible files on mirrors and employing a transparent scheme to redirect users to the nearest mirror with the latest, most complete copy of all the files needed.

A robot gathers directory listings from FTP servers, and a summarizer component parses these and creates MD5 digests on a per-directory basis. A database back end using MySQL keeps track of FTP Mirror Tracker data and links collections to the domains being tracked. A digest exchange compresses and moves the digests into a Web-accessible area (for other trackers to access), and front-end programs provide the means for users to query trackers. An ICP (Internet Cache Protocol) server was also written as a means to allow cache querying by other Web caching systems.

This comes into play when users are redirected to the closest FTP mirror, by using Squid to redirect URLs to the ICP server component for rewriting.

Internal support for URNs (Uniform Resource Names, i.e., a persistent, location-independent naming scheme that decouples location from the name of the resource) has been added to FTP Mirror Tracker.

The system has been put into production, and preliminary results show a reasonable cache hit rate, but improvements are expected. Some of the functionality implemented on top of Squid have been folded into Squid itself as of the distribution of version 2.3 .

SESSION: FULLY AUTOMATIC

Summarized by Socrates Pichardo

Deployme: Tellme's Package Management and Deployment System

Kyle Oppenheim and Patrick McCormick, Tellme Networks

[Winner of the Best Paper Award]

Deployme is Tellme Network's solution to manage the package update life cycle across a large number of independently configured hosts. It is highly flexible and has been extended to handle many different types of packages. These packages include standard UNIX tools, local applications, Web site content, and voice site content. Deployme value can be maximized on packages that require fast, frequent deployment.

Deployme's mission is to provide a central system for tracking the entire life cycle of software packages. Its designing goals are:

- Support a wide audience
- Robustness
- Augment the development process
- Flexible destinations
- Efficient use of network bandwidth
- Quick pushes
- Seamless activation
- Rollback
- Scalability

On the other hand, Deployme designers intentionally left out several features while pursuing simplicity and shorter time to market. These features are:

- No local package management
- No dependencies
- No fine-grained operations control

Deployme is written entirely in Perl 5 and has a simple three-tier architecture.

Although Deployme is a well-implemented solution to Tellme Network's package management problems, it lacks certain features/services in certain areas. The authors mentioned some of Deployme's shortcomings:

- It doesn't have the concept of "sites" or several machines sharing a physical location.
- The lack of "transactions" at the database level makes it difficult to accurately determine the integrity of the data after a system failure.
- Deployme's lack of multicasting support negatively impacts network utilization when the same package is sent to a large number of servers.
- The system has no security features as of version 1. There is no control over who, what, or where.

Tellme Network is currently working on solutions for many of these shortcomings.

Automating Request-Based Software Distribution

Christopher Hemmerick, Indiana University

Netdist is a very complete solution for software distribution that was designed from the ground up with security, modularity, flexibility, and extensibility in mind. Netdist provides an automated mechanism for system administrators to request and receive software exports with an immediate turnaround. The system provides a simple user interface, secure authentication, and both user- and machine-based authentication. Each of these is configurable on a package-by-package basis for flexibility.

Netdist is a modular service. The user interface, authentication, and authorization are independent of the export protocol. The author is currently distributing via NFS, but adding an additional protocol is as simple as writing a script to perform the export and plugging it into Netdist.

Netdist is implemented using Perl 5 and some modules from CPAN, PGP, cron (or any other job scheduling service), and an instance of Apache with at least mod_perl and preferably a module for secure transactions. The NFS export control scripts have been written for Solaris but could be easily ported to other UNIX flavors.

The only shortcomings of this tool are its lack of installation scripts and availability. Netdist is still pre-alpha, and a lot more work is needed in order to ease installation. Also, several of the Perl modules and scripts do have host- or port-specific information coded into them. Although each of these instances is documented, the authors will attempt to extract all these values into a configuration script in the next version.

Use of Cfengine for Automated, Multi-Platform Software and Patch Distribution

David Ressman and John Valdes, University of Chicago

The author's main requirements were to create or buy an automatization package for software and patches distribution in order to improve the level of services being provided to their end users and to liberate their two SAs (authors) from these repetitive tasks. Some of the important characteristics of the solution were cost, ease of use, current development, and security.

Their solution was to "glue" with Perl some of the "best of breed" tools available for the different tasks. They took Cfengine (Configuration and System Management tool), NFS (for their file system exports), and RPM (Red Hat Package Manager) and used them as building blocks, in addition to Perl and mySQL, to create the Web interface as well as the back-end database.

The outcome for the software distribution problem is a Web-based front end where users can request which software package they want to install. Once they submit their request, the system will insert these requests into the database, create the necessary exports, and offer users the opportunity to launch the RPM module requested. On the patch distribution side, hosts will check periodically with the software depot server for new patches available for their OSes and architectures. Once clients find new patches, they will proceed to install them and to report back the exit code of such installs.

Future work on this project will involve expanding its services to include OS upgrades as well as support for other UNIX flavors.

SESSION: BUILDING BLOCKS

Summarized by Vinod Kutty

Unleashing the Power of JumpStart: A New Technique for Disaster Recovery, Cloning, or Snapshotting a Solaris System

Lee Amatangelo, Collective Technologies

A production data center requires processes to reduce downtime of critical servers as far as possible. Apart from hardware/software high-availability solutions, a disaster recovery plan is a must.

This paper describes a system that provides the following for Solaris systems:

- Bare-metal recovery
- Creation of a system snapshot on optical media
- Cloning of a system
- Rollout of multiple system clones

A typical application is one in which the root drive(s) of a server has failed or been corrupted by human error or hardware failure. Traditional backup/restore methods are not feasible when the OS cannot run.

One approach is to use bit-level imaging (or "ghosting" in the PC world), which can be quite fast, but not as flexible as something like JumpStart, which is the alternate approach that performs automated Solaris installations.

The solution – the Capture and Recovery Tool (CART) – combines both techniques. The tool evolved in an environment where security was important, and this is reflected in the requirements:

- No magnetic media allowed
- Recovery media must be bootable
- There should be minimal user interaction
- Multiple sets of removable media must be handled (e.g., if a snapshot's size requires three CDs)
- Should not involve directory services such as NIS, NIS+
- Should not depend on NFS

The implementation depends on the Solaris installboot command – which can place boot blocks on optical media – and the customizable nature of JumpStart.

A good understanding of JumpStart operation is required to understand the customizations made, but the important points are:

- Although typically associated with network installs, JumpStart is also a part of traditional installs of Solaris from CD-ROM, installing from local media rather than the network
- CART plugs into this JumpStart mechanism and replaces certain scripts so that JumpStart does not perform normal installation of packages, patches, and so on. Instead, it provides enough of a boot up process to get to the stage where a CART script can be run, after which JumpStart relinquishes control to CART

Future enhancements to CART include integration into a networked environment and implementation on other UNIX variants.

A Linux Appliance Construction Set

Michael W. Shaffer, Agilent Labs RCS

The motivation for this project started with the author's need to support remote installations of Linux servers providing file, print, and network routing services located in areas with few skilled personnel capable of disaster recovery.

One way to address the support issue and establish a fairly error-free disaster recovery process is to eliminate the traditional install of the operating system and additional software, and instead boot and run directly from removable media, using a Linux distribution configured for this purpose.

Rather than independently tuning a Linux distribution for each specific purpose – such as a print server – the scope of the project was enlarged to create a more generic framework for creating minimal Linux systems. Hence the name "Linux Appliance Construction Kit (LxA)", where the 'x' represents the function of the appliance (e.g., LPA == Linux Printing Appliance).

The design and implementation of LxA followed several principles:

- Systems are built by composition of needed pieces rather than reduction of a large set of components
- Systems run from read-only and/or removable media (although hard drives may be used for swap, /var, /tmp and other transient storage)
- Omit login and run-time configuration (except during development, where facilities such as console login and an interactive shell may be needed for debugging)
- Use modern, standard components, such as the kernel, libc, and so on

A lot of work goes into determining what is needed, testing the images, creating bootable CDs/floppies, and so on. The underlying technique for running an LxA system is to use an initrd image and run the entire system from it at boot time.

There are numerous advantages to using LxA over the more general purpose Linux installations, and these were also important goals in the design:

- Reduced complexity, which in turn enables better documentation and a more thorough understanding of the system
- Reduced security vulnerability, resulting from simplicity
- Reduced setup, maintenance, and upgrade time
- Reduced probability and impact of hardware failures

Future work will address more types of LxA appliances, enhancements to the existing LPA-CD appliance, and automated scripts for identifying components needed for new LxA systems.

More information about LxA can be found at <*http://www.equusasinus.com/lxa/*>.

Automating Dual Boot (Linux and NT) Installations

Rajeev Agrawala, Shaun Erickson, and Robert Fulmer, Lucent/Bell Labs Research

Although tools are available to automate the installation of Linux and NT, there are no good tools to automate the installation of PCs that dual boot either Linux or NT, from separate partitions on a hard drive.

This motivated support personnel at Lucent/Bell Labs Research to design a solution to this problem, as users were already starting to use dual boot installations of NT and Linux. These were inconsistently installed by different admins, time-consuming to perform, and not reproducible.

The solution employs automated installs starting with a modified "bootnet" Red Hat Kickstart floppy. The alternative is to use disk cloning, but this requires similar hardware and peripherals, cannot deal with unique NT SIDs, and involves a lot of work in updating an entire image when any piece of software is changed.

The process is designed to start with an admin booting from a floppy and selecting an install option (NT 4, Linux, or both). For dual boot installations, the first OS installed is Linux, using Red Hat's Kickstart. Automated customization is performed, the disk is repartitioned, and a DOS file system is created for the second OS install, namely NT. Note that this file system is eventually converted to NTFS.

Some files required for the automated NT installation are copied to this partition, and a reboot occurs to invoke the NT installer. After the NT install and customization steps are complete, a reboot surrenders control to the first OS – Linux – where final configuration of X and audio must be done manually, due to problems with lack of device driver support that could interrupt the automated installation.

Some difficulties were encountered with passing information about the installation type to the Linux kernel, creating two primary partitions at once, and locating LILO in /boot vs. the Master Boot Record. The design accommodates solutions and/or workarounds for these where necessary.

The system has been in use for more than six months, and future plans include support for automated X and audio configuration in Linux, and the addition of other operating systems (e.g., Win 2000).

Source code and configuration profiles are available from the authors <*dualbootinfo@research.bell-labs.com*>.

Network Track

Deploying Quality of Service Features on Your Network

Eliot Lear, Cisco Systems

Summarized by Paul Federighi

Eliot Lear's talk described what quality of service (QoS) is, why you might need it, and the methods for achieving it. The talk was mainly focused on QoS as it pertains to voice communication on an IP network, though other types of data such as video and Web traffic were also mentioned.

As Lear explained, QoS is a method of giving preferential treatment to certain types of data on the network. Applications such as interactive voice and video have special needs. Voice has certain bandwidth and latency requirements and is drop sensitive. Most packets must make it through with less than 200 ms of latency. This includes transmit time and any queuing delays. QoS is not important for non-interactive traffic or non-time-critical traffic that can be buffered.

Lear stressed the point that QoS features are needed on every piece of equipment in the communications path where packets can be queued. This includes routers, Ethernet switches, ATM switches, frame relay, etc.

There are two models for achieving QoS: integrated services (Intserv) and differentiated services (Diffserv). With Intserv, the application specifically requests (via RSVP) resources at every hop along the way. Call setup happens first, then receivers request reservation (in both directions). Some of the advantages:

- Works with both unicast and multicast traffic
- End points know early on whether there's a reservation

By contrast, Diffserv has no end-to-end signaling. Instead, it uses per packet marking rather than marking the entire stream. Traffic is marked based on a policy domain and is policed at the edges. Since there is no signaling, an error could cause an application to fail silently. One needs to pay careful attention to traffic engineering and either allow for additional bandwidth on links or constrain traffic to predictable paths. Packets are separated into different classes:

- Best effort
- Assured forwarding (AF) – all data will get there
- Expedited forwarding (EF) – preferred over other traffic

Lear explained several different buffering and queuing techniques on network equipment. The old way is to use a FIFO. When congestion occurs, the end of the transmission gets dropped. However this is unfair to lower bandwidth protocols, and dropping packets just wastes bandwidth. Methods for overcoming this include random early detection (RED), weighted red, priority queuing, and weighted fair queuing. Other methods were mentioned as well.

Management and monitoring are important with QoS. You need to know if your packets are getting through and if the latency is tolerable. Tools are just starting to become available.

Trying to achieve QoS across the Web has scalability problems. There is research beginning with "bandwidth brokers." Right now you can't get QoS across multiple providers. Instead, a good idea is to distribute data throughout the Web to reduce latency and get better bandwidth utilization.

When asked about security, Lear responded with the point that you can't do packet classification on encrypted data.

When asked about features to look for in hardware, Lear mentioned several questions to ask, including, "Are there multiple output queues?" and "Can you classify data in the output queues?" It's also important to remember that bottlenecks typically occur because of the line cards used, not the backplane of the device.

SESSION: ANALYZE THIS!

Summarized by Tony Katz

Wide Area Network Packet Capture and Analysis

Jon T. Meek, American Home Products Corp.

Jon Meek's talk covered why we need to analyze and how he went about doing it. We need to know what is happening on the wire to diagnose such problems as slow applications and network congestion. To monitor the frame networks, Jon used a small PC running Redhat Linux, which was plugged into the CSU/DSU via a serial connection to capture HDLC packets for analysis over time. The topic of time interval was interesting. You can capture packets continuously, which takes a lot of resources, or in intervals, such as 15 minutes or 10 seconds. What interval you choose can make a significant difference. Jon did all of this using C and Perl. It is a fairly inexpensive way to monitor your frame relay and get reasonably good results.

Sequencing of Configuration Operations for IP Networks

P. Krishnan, IPSoft, Inc.; T. Naik, Bell Labs; G. Ramu, CoSine Comm., Inc.; and R. Sequeira, IPSoft, Inc.

P. Krishnan addresses the problem of losing segments when updating routing configurations across a complex network if the updates do not happen fast enough. The proposed solution is sequencing. This solution will work but it assumes many things, e.g., that you are using OSPF, that routes are static routes, etc. This is achieved by indirect telnet, traceroute, and reverse traceroute. Not bad if your environment fits all the criteria.

ND: A Comprehensive Network Administration and Analysis Tool

Ellen Mitchell, Eric Nelson, and David Hess, Texas A&M University

There are lots of vendors and even more software out in the world and each one does something important, but none of them do it all. ND was designed by Texas A&M to accomplish this task. They wanted it to be powerful but low level, portable, customizable, and scalable. It was written primarily in Python, and uses SNMP, SQL tables, and a MySQL back-end database. Python was used for its modularity. With ND you can enable ports, configure new devices, do scripting, but best of all is that it has built-in documentation. This is a great feature, not so much to point a finger at someone, but to know who to talk to about why a particular change was made. Texas A&M is also looking to add event monitoring to ND. Currently, ND is not available but will probably be released to the public sometime in the future.

SESSION: GO WITH THE NETFLOW

Summarized by Tony Katz

Combining Cisco NetFlow Exports with Relational Database Technology for Usage Statistics, Intrusion Detection, and Network Forensics

Bill Nickless, John-Paul Navarro, and Linda Winkler, Argonne National Laboratory

The first part of the presentation given by Bill Nickless focused on the problem of having a high performance network with a minimal firewall.

Cisco's NetFlow provides a summary of data traffic through a router. This data must be captured and analyzed. Argonne's way to do this was through the use of database technology. Their hurdles are the amount of data coming in and the ability of the database to keep up. They went with a high-powered database running on an SGI Origin 2000. They experimented with both MySQL and Oracle 8I but ended up using a SQL back-end database. They used Perl scripts to catch the data and feed it into the database for analysis. This worked very well overall. The big issue is that not every site has an Origin to process thousands of records at a time. Scalability is determined by your database application and tuning parameters.

The OSU Flow-Tools Package and CISCO NetFlow Logs

Mark Fullmer, OARnet, and Steve Romig, Ohio State University

Steve Romig spoke about his application of NetFlow. Their interest was more of a security focus. They also had a voluminous influx of data but handled it a little differently. OSU looked at aggregation, collection, viewing, and security using a set of tools they created called Flow Tools. They reduced the data load by aggregating the data into summaries. This allows viewing at any given point. Their security features were the most interesting. There has been a lot of focus put on incident response. OSU used a variety of Flow Tools to detect "interesting network traffic," host or IP range profiling, as well as detecting network attacks, i.e., denial of service. OSU is continuing to expand their set of tools to do more in the realm of the client/server role on the end points of the wire. For a closer look at the tool set, check their Web site <*http://www.net.ohio-state.edu/software*>.

FLOWSCAN: A NETWORK TRAFFIC FLOW REPORTING AND VISUALIZATION TOOL

Dave Plonka, University of Wisconsin-Madison

Dave Plonka's presentation was saving the best for last. Not to say that the other implementations were bad, but they did not have a visualization component and Dave's did. The University of Wisconsin-Madison used an open systems software package called FlowScan. FlowScan uses a report module called CampusIO to accumulate the raw flows and push the statistics to a round-robin database. The data was then taken by FlowScan and graphed. The graph, which was color coded to traffic, was able to show both inbound and outbound traffic. The colors of the graph differentiated between HTTP, FTP, and (the ever-popular) Napster traffic. The benefits of the graph over data figures is that you can instantly see what type of traffic is using the most bandwidth at any point in time. Graphing also helps you pinpoint anomalies easily. Future expansion points for FlowScan are in the area of event notification or alerts. For more information go to <*http://net.doit.wisc.edu/~plonka/FlowScan*>.

This was a great depiction of the uses of Cisco's NetFlow. It is a definite improvement over analysis by sniffer.

BROADBAND CHANGES EVERYTHING

Summarized by Josh Simon

Brent Chapman, Great Circle Associates

Brent Chapman spoke about how broadband – which includes the variants of DSL, cable modems, and possibly even wireless – changes the way people perceive the Internet.

Broadband has two features: it's high speed and always on. DSL provides speeds on the order of 144Kbps (or more than 7Mbps). Cable modems share the same big pipe but provide similar high speeds. In comparison, even the fastest phone-modems provide no more than 53Kbps. By "always on," Brent means that there's no longer any dial-up delay and no busy signals. This makes the Internet like electricity or water: you flip a switch or turn a knob and it's just *there.* This will change how people perceive and use the Internet in the long run; rather than saying, "I'll go online later and do that," they're much more likely to hop on and off the Net for brief visits to accomplish tasks as they come up as opposed to waiting until later. (Note that most consumer electronics today – stereos, televisions, and microwaves – don't actually power themselves completely off. They remain in a reduced-power "stand-by" mode so they can appear to power up more quickly when needed.)

Broadband is also cheaper than traditional leased lines. A T1 line from a telecommunications provider (telco) used to run $1,500 a month. Comparable speeds via DSL are on the order of $300 a month.

The revolution in providing broadband leads to new capabilities, such as connecting small offices or home offices to the Internet at high speeds, as well as making telecommuting more effective for virtually everyone. It also leads to new services or more efficient older services, such as:

- Streaming audio and video
- Backing up over the network (such as @Backup)
- Software auto-updates (Apple, Symantec)
- Push services (PointCast)
- Cooperative computing (SETI@home)
- Interactive games

Unfortunately, broadband also leads to new security threats. "Always on" means "always vulnerable." You can no longer assume that you can only be hit by attacks when you're online in front of the computer when the Internet link is always up. Cable modem lines are shared within a neighborhood, so "Network Neighborhood" takes on a whole new meaning. If you have shared your disk or printer within your own home, you're also sharing them with the entire cable neighborhood. We should expect to see new hardware and software firewalls built into broadband DSL in the near future.

Broadband also allows you to save money. Many homes have more than two computers, so networking them within the home to share a single big pipe for bandwidth makes more sense to more users now. This means that you could cancel your second phone line (saving about $15/month) as well as multiple ISP accounts (saving $20/month).

What's coming in the future of broadband? Brent expects that virtual ISPs (for sales and marketing features), affinity ISPs (like credit cards), subsidization, and cross-marketing will happen in the near term. We'll also see voice-over DSL and voice-over cable (some areas already have one or both of these); the problem faced by the providers here is "five 9s reliability," or less than five minutes of downtime – scheduled or unscheduled – per calendar year. We'll see more network appliances (like WebTV and Tivo) and radio- and broadband-ready MP3 receivers. We'll also see Internet-enabled appliances, such as the refrigerator with a touch screen for restocking linked to a grocery delivery service such as Peapod or WebVan.

There are several IT management issues with broadband. First among these is security: should employees' homes be inside or outside the corporate firewall? If they're inside, who other than the employee has access to the company network? If they're outside, should the corporate Internet access be shared with the homes? If so, we need to have some kind of firewall protection (but then who maintains and monitors those firewalls?); if not, the cost to the company will skyrocket since every home user needs to have their own bandwidth. What carriers are available to the employee? Who supports and supplies the home system? How can you provide mutually secure access, such as when an employee's spouse works for the competition? Is a VPN the right solution? If so, is it PC-based (which leads to driver issues) or router-based (which doesn't address the other-people issue)? Are personal firewalls the answer? Those also lead to issues of who provides, configures, reconfigures, manages, and updates them, and ignores the multiple-connection issue.

In the question and answer section, Brent noted that distributed denial of service attacks (DDoS) will increase. Host-based security has to come back into style, since firewalls are no longer enough protection. The Cheswick/Bellovin model of a crunchy exterior and creamy interior no longer applies. Satellite broadband is unlikely because of the huge latency involved. Broadband affects the core routers. When asked what it'll take to administer the high-bandwidth providers (such as Akamai), Brent noted that there's no good answer yet but we certainly need to work on it. As an example, Akamai has 600 servers and is moving toward 600,000 servers. Broadband also leads to more peer-to-peer networking, so the traditional source-and-sink model may need to be redefined.

Security Track

Cops Are from Mars, Gurus Are from Pluto: Dealing with "the Feds" and Other Cops

Tom Perrine, Pacific Institute of Computer Security

Summarized by Dave Homoney

Tom made this a very interesting talk. His injection of real-world scenarios was very helpful. The talk was geared toward sysadmins, those of us who might have a run-in with a hacker and need to know where to turn and the protocols to use. He also talked about what to do when law enforcement (LE) comes to you.

Tom described how to tell if a call from the FBI is a hoax: if an FBI agent is calling you directly, it probably is. He stated that most federal agencies will contact you through a local law enforcement agency. He also said that contact by the FBI would be through the nearest local office.

Tom mentioned several cases in which LE screwed up, particularly the case against Steve Jackson. This case produced a lot of negative press for LE and marked the point at which LE moved from the guns blazing approach to the computer savvy LE officer approach.

Tom also talked about when not to help LE, stating that the first thing you need is a good lawyer. You don't want to do anything as "directed" by LE or you could be considered an agent of the government, causing you problems in court. Instead, you should follow the directions of your company's legal staff. And, of course, you will need to comply with all court orders such as subpoenas.

Finally, you should create a bridge between yourself and LE so that when you need them, you'll have a friendly contact. He mentioned several times that "they are just like us," adding that there are always exceptions.

Does It Take the Same Skill Set to Secure a System as to Break into It?

Panelists: Peter Shipley, Lab OneSecure Inc.; Mark Hardy, Guardent, Inc.; and Elias Levy, securityfocus.

Summarized by Dave McFerren

Some of the topics the panel discussed were:

- Should companies hire crackers to catch other crackers?
- Can you trust someone who was once a cracker?

Discussion was fairly one-sided concerning the skill set needed to break in compared to the skill set to secure. The overwhelming majority of opinion was that cracking requires only a subset of the skills needed to secure systems; there are many different ways to compromise a computer, and a cracker needs only concentrate on one particular service that the computer may deliver. Another topic tossed about was the question of whether you can hire "black hats" to do "white hat" jobs. Although there are many startup or fringe companies that tend to do this, the general consensus was that you should become a white hat by yourself and make a foray into the corporate world before earning the trust of the "suits."

The most interesting discussion arose in response to the question, "What does it take to become a security expert?" One panelist – who had previously been a black hat – insisted that you had to breathe, live, and eat security for years to become good at it. Others disagreed, believing that you can have a life with family and friends and still be able to do a good job at security. But most agreed that since security requires more than the traditional 40 hrs/wk, a person would have to be at least somewhat obsessed with the subject to be really successful.

Overall, the discussion was interesting, although I was not sure I got more than the single perspective held by the computer security profession. But it did show me the "other side" of the security issues that I deal with on a day-to-day basis.

REAL-WORLD INTRUSION DETECTION – FIRST STEPS

Mark K. Mellis, SystemExperts Corporation

Summarized by Steve Wormley

Mark Mellis covered much of what is needed to set up intrusion detection using primarily free products on a small-to medium-sized network. He noted this discussion didn't apply to larger sites, because they generally have larger problems, but the basics are the same.

He first gave an overview of why one would do intrusion detection (ID). Basically the crackers are becoming more sophisticated, the networks are becoming more complex, more protocols are flowing through the firewall, and there are more connections to business partners and other points of attack.

The assumptions for this talk were that the solution needed to be cheap, that the SA was familiar with ftp and make and normal freeware/open source setup, and that the admin was busy and ID was a part-time job.

ID systems should tell you when real threats occur but be able to log even door rattling. Important things to trigger on are config changes, auth failures, attempts to probe the site, and attempts to access services.

The things he recommended deploying included centralized syslog functions (UDP syslog or nsyslogd); a log-analysis product like log_analysis or log surfer; a tripwire like Tripwire or aide; Klaxonto monitor for connection attempts on unused ports; sscanlogd to monitor for portscans; and a product like snort, which is a lightweight network IDS.

A couple of points to remember: routers are hosts too and need to be monitored and can use syslog to do so. To capture authentication failures, brute force does work. Don't forget application exploits; scan the Web and appserver logs for anything out of place.

Finally, all exposed and all infrastructure machines should be running host-based ID, and network ID systems should be put where traffic is both concentrated and sensitive. And, of course, anything that can be done is better than nothing.

This was a good overview of the products available and things that can be used in ID. It was a good talk for anyone who needs to install this type of service on a small scale or as a precursor to learning how to do it in a large environment.

Mark K. Mellis' URL is <*http://www.systemexperts.com*>.

SESSION: SOMEONE'S KNOCKING AT THE DOOR . . .

Summarized by Eric Lakin

TRACING ANONYMOUS PACKETS TO THEIR APPROXIMATE SOURCE

Hal Burch, Carnegie Mellon University; Bill Cheswick, Lumeta Corporation

This paper was one of two papers to receive the "Best Paper" award for LISA 2000. It was based on work done a couple of years ago within Lucent's corporate network, concerned with ways to find the source of a denial of service (DoS) attack when the source of the packets is forged (the norm). Some of the assumptions made in the research which limit the usefulness of the technique against distributed denial of service (DDoS) attacks seen recently include the following: the source of the DoS packets is a single source, no modifications to the current network infrastructure can be made (router or protocol), the attack is long term, and the packet-rate is constant. Further, only DoS attacks that seek to overwhelm the victim with bogus packets are considered; attacks that attempt to cause a malfunction by specially crafted input are not.

When considering the network topology for the purposes of a DoS attack, it was convenient for the authors to describe it in terms of a tree graph. The victim's machine is at the root of the tree, with network nodes being nodes in the tree. Each path out of the victims local network subnet is a branch in the tree, with further branchings being paths out of the connected subnets, and so on. One path to a leaf node is the attacking host.

Because the source of the DoS packets is almost always spoofed, the assistance of the ISPs and network administrators outside the victim's network are usually required to locate and shut down the attacker. There is often little motivation for these people to help, and even finding the appropriate person to help may be a challenge. If the appropriate people are found and are willing to help, they can either put their routers into debug mode to determine which path the attack is taking or selectively cut off paths briefly and see if the attack slows or stops.

When the outside network managers cannot be contacted for some reason, is it possible to determine the approximate location of the attacker, without physical access to the outside network or their routers? The authors of the paper were able to come up with a way to selectively "deactivate" a line remotely by using the "chargen" service and UDP broadcast packets to selectively overwhelm, or DoS, a branch in the tree. By selectively overloading individual branches of the tree and watching the rate of incoming packets, one can determine with increasing accuracy where the attacker is located.

Because of the method used to overwhelm the branches, accuracy is limited to determining the subnet of the attacker. This, however, is enough to allow contacting of the appropriate ISP. In simulated DoS attacks within Lucent's network, the authors were able to find the attacker's subnet three out of five times, and were always able to determine the subnet within two to three hops.

The ethicality of this procedure was briefly touched upon, and it was acknowledged that this was of more academic interest – and should never be used in real situations on the Internet. Further, the program written to do the testing and analysis is not going to be released.

Analyzing Distributed Denial of Service Tools: The Shaft Case

Sven Dietrich, NASA Goddard Space Flight Center

The purpose of a denial of service (DoS) attack is to overwhelm the victim so that it is unresponsive to legitimate users, or to construct input to make the victim host act strangely or unreliably. The former is more common, general purpose, and what this paper focuses on.

The simplest form of DoS involves an attacking host sending packets directly to a victim host. As it was not always possible for a single host to saturate a victim, further refinements were made to DoS methods. Using amplifiers – hosts that amplify the amount of traffic output by an attacking host – became common with the "smurf" attack. Another possible method was to coordinate among multiple attackers to concentrate on a specific victim. However in the past, such coordination has been largely manual, such as agreeing to a time and victim through IRC.

Distributed DoS attacks are a recent "innovation" in the DoS toolkit. In the DDoS model, one or more attackers relay commands to a "handler" host, which maintains a list of "agents" – compromised machines running software to attack victim hosts. Through DDoS software, one individual can direct tens or hundreds of machines to attack targets, with the multiple sources of the attack making it highly effective and extremely difficult to stop.

The DDoS tool analyzed in-depth by the paper was called "shaft," and was the second such software available, after the original trinoo. Most analysis of shaft watched traffic between a compromised agent and a handler, as well as actual attacks which the agent participated in. Analysis revealed the attack methods the shaft tool used – a combination of TCP, UDP, and ICMP flooding – and the communication channel between the agent and handler was discovered.

SESSION: . . . DON'T LET THEM IN

Summarized by John Ouellette

YASSP! A Tool for Improving Solaris Security

Jean Chouanard, Xerox PARC

Purpose: to correct Solaris defaults:

- Permissive file modes
- Too many services running
- Inconsistent logging

Philosophy:

- One config file
- Model after Sun package – easy to uninstall
- Tolerant about what it expects
- Server or workstation based

YASSP's project:

- SECLean – core package
- Other: RCS, openssh, Tripwire, tcpd+rpcbind

Goal: to control startup of init scripts, while allowing the machine to return to its pre-YASSP state.

Needs: more English-friendly docs and testing for non-default Solaris systems.

Additional information on YASSP can be found at <*http://yassp.parc.xerox.com/*>.

NOOSE – Network Object-Oriented Security Examiner

Bruce Barnett, General Electric Corporate Research & Development

Purpose: to present tools that are lacking.

Content and function: distributed cooperative engine: Dispatcher, IW, GUI. There are presently 21,000 lines of Perl/Tk code at "research" quality level. This tool checks for patches generated from the Sun FTP site; looks for Trojan horses; parses start files; tracks variables, $PATH, etc.; examines .rhosts; understands NIS netgroups; checks NFS access to users' homes.

Problems: single-threaded, not secure.

Performance: host with 2,000 accounts took 30 minutes to check 5,000 vulnerabilities.

Future: will be TCP based, multi-threaded.

Conclusion: object orientation is key to reusable algorithms.

Subdomain: Parsimonious Server Security

Crispin Cowan, Steve Beattie, Greg Kroah-Hartman, Calton Pu, Perry Wagle, and Virgil Gligor, WireX Communications, Inc.

Problem: granting least privilege not always easy, or feasible/possible. For example, mod_perl runs at Apache level of permissions.

Solution: ACLs for programs, instead of users. Subdomain is a kernel-level enhancement to confine programs.

For instance, program foo can be restricted by a config file:

```
  foo {    /etc/readme, r
 /etc/writeme, w }
```

This gives program foo read access to /etc/readme, and write access to /etc/writeme by invoking the chhat() method (i.e., change hat).

Workshop Series

Workshop 2: Teaching System Administration

Coordinators: Curt Freeland, University of Notre Dame, and John Sechrest, PEAK, Inc.

Summarized by Socrates Pichardo

This workshop was a continuation of last year's work. The goal this time was to brainstorm ideas for class material, exam questions, concepts, etc. There was a good representation from all aspects of system administration education, with a heavy concentration of college/university educators. Here is the outline of the workshop:

Session 1
  Concepts and prerequisites

Session 2
  Concepts taught
  War stories
  Best exam question
  What I wish class looked like
  Getting students to participate in class
  Measures of success
  Questions for a prerequisite (minimum competency) exam

Session 3
  Tools we could use
  Develop examples

Session 4
  Develop exam/homework/project
  What we would like to have for a program/track/series/concentration
  Workshop review

Results of this workshop will be published under the mailing list of the working group (<*sysadm-education@maillist.peak.org*>) and will be part, along with the Sysadm Taxonomy Project, Sysadm Certification Project, and Benchmarking and Measurement Project, of SAGE's efforts to formalize the system administration profession.

Workshop 3: MetaLISA

Coordinators: Cat Okita, Global Crossing, and Tom Limoncelli, Lumeta/Lucent Technologies

Summarized by Josh Simon

Six major topics were covered:

Staying Technical

We had a good hour-plus discussion on how to stay technical and manage your boss. Some subjects that came up included having both responsibility and authority, getting someone to do the nontechnical aspects while you concentrate on the technical ones, defining and reviewing roles and responsibilities, using agendas to run and control meetings, knowing when to say "I don't know," holding regular "town hall" type meetings for the user community, and apologizing when you screw something up.

Retention, Hiring, and "Coaching Out"

The general consensus here was to involve the Legal and Human Resources departments as soon as possible and to document everything. If you need to encourage someone to leave, you can either treat it as a pure performance issue or possibly a security issue (for example, if the employee in question has root). If performance is an issue, you can use improvement plans or a probationary period. You'll definitely have to manage the morale of those who stay.

This led to a discussion on hiring. First, how do you find qualified applicants? You can look online, though for more signal and less noise you can go by word of mouth, the SAGE jobs list, and even campus recruiters. Second, how do you convince these qualified folks to join your company? Some thoughts for managers focused on looking at the long term, not the short term, since you cannot easily get rid of someone you've hired: do you have to raise or adjust the salary of the new hire? of the rest of the team? do you have to train people? Are bonuses involved?

This led to a discussion on retention. In order to keep employees on your team, we determined that managers need to have flexibility in providing raises (both in terms of frequency and amount) and reviews (more often than just annually). Providing perks, such as training, conferences, laptops, high-speed network access, soda and beer, flex time, toys to play with (both computer-related and non-), cool projects, good management, good co-workers, good environment, and respect and recognition can help you retain your best employees.

Leaving Gracefully

If you find yourself in the position of leaving a job, you should leave gracefully. Hand off your responsibilities, make sure no batch or cron jobs run from your own account, document everything (both what happens and why), and put read-me files in nonstandard directories and hosts. Be professional and do what's right for the company; don't send any hate mail. Whenever possible you should train your replacement. As a manager, you need to plan for your people leaving, be it by leaving the department, leaving the company, or being promoted out of a position. Some other questions that arose included:

- How do you decide when to quit?
- How do you handle a subordinate being promoted over you?
- How do you manage friends?

Talking (Up) to Management

When talking to your management you have to remember to tune to the audience. Talk about the technical issue in terms that your audience is familiar with. You need to focus on the business reasons and issues and not the technical jargon. You should ask your peers, or even your boss, to review any messages you're about to send. Find something in common with the manager and use that as a basis for establishing rapport in your communications.

Talking to Now-Subordinate Peers

When you're the one who's been promoted to lead your team, there are a few things you need to remember. You have to be careful with social events; as a manager you're no longer "one of the gang." There is going to be some information you cannot share with your team, and they are going to know that. You have to treat everyone the same regardless of how you may feel toward them (such as friends and non-friends in the same team). The dynamics will differ by group size; managing one or two people is different from managing 10 or 12. Finally, you have to be objective and impartial.

General Tips and Techniques

We wrapped up the day with some general tips and techniques for being good managers.

- Set boundaries for your team.
- Let your employees fail.
- Remember to test, and include testing in projects.
- Consider when to delegate or take over something.
- Review your team, your peers, and your management.
- Spend a lot of time up front on concepts and requirements of projects.
- Protect or insulate or buffer the team from more-senior management.
- Back up (support) your employees; trust your team.
- Stay out of the office to (1) delegate and (2) find out what does and doesn't work without you.
- Don't micro-manage your employees.
- Communicate up, down, and across; open communications are very important.
- Teach skills, not things or details.
- Don't lie to yourself.
- Assume the other party is trying to do what's right for the company.
- Bounce thoughts off peer-level managers.
- Find a mentor (either inside or outside your organization).
- Do one thing every day that scares you.
- Match customer expectations with reality; prioritize.
- Rotate your team through the various positions to reduce the risk of burnout.
- Don't decree decisions, unless you have no time to reach consensus, you cannot reach a consensus, or there is an obvious violation of policy.
- Kill off or postpone projects when necessary.

Workshop 5: Advanced Topics

Coordinator: Adam Moskowitz, LION Bioscience Research, Inc.

Summarized by Josh Simon

The professionalization of systems administration was one of the topics discussed. A comparison was made to doctors. We use similar skill sets – diagnosis, comparability, problem solving, and so on. But can it be said that lives are at stake when systems administrators do their job? Doctors charge by the visit or the procedure; systems administrators don't. The models are, however, converging in some ways. Many systems administrators are more concerned with architecture than are doctors. There are differences in scale: doctors are like help desks, while systems administrators tend to serve larger numbers of people. Doctors are, in fact, certified. Some systems administrators contravene organizational policies. Doctors are liable, lawyers are liable, and engineers are liable; systems administrators are rarely liable. This led to a discussion on professions: professions have standards for training and knowledge (certification); there's a fixed set of information. Sysadmins are often grassroots with self-training and apprenticeship. Certification is a required stepping stone. Maybe systems administration should be a "guild." Or maybe we should form a union.

A second area of discussion was whether or not ISPs are now perceived as commodities and whether they can be run as commodities. The consensus was that they can, but you should be sure to check out their long-term business prospects because business models change rapidly. Finding a provider for services "beyond the basics" is hard. ISP consolidation is in progress. Any new ISP will require new technology. Not only are ISPs perceived to be commodities, so are their users (who are traded). Local and national ISPs can survive; but it's tough for regional ISPs, who are neither local nor a brand name. Are there brokers for customers? There are special deals among ISPs, but no B2B site. DSL was enabled by aggregating terminations at the central office. Those who can scale will survive. You can now purchase a turn-key 10- to 50K-user ISP solution that requires very low levels of sysadmin skill. Shell accounts are a thing of the past; people are running their own servers at the end of a DSL line.

A third major area of discussion was on separating policy and implementation. One possible solution is to have an interpreted "policy language." Maybe you can use general principles and then color the bottom-level implementation to match existing policies. This is more of a mindset problem than a coding problem. Let's build policy engines, not engineer accounting (or whatever) systems. Cfengine has features that can help you implement policies. You must codify the policy in a way that's measurable so you know if you're "on policy" or not (and then you can get back on policy if you get far enough "off track"). We're already adapting host-based tools that query directories. Maybe we can graft policy engines onto directory responses.

A fourth area of discussion was on how new technologies in the last few years seem to be languages. This is true because languages can express extensible ideas; they build from primitives and move to greater complexity. Some people say "use a database for policy," but databases too often require predefinitions. Languages, on the other hand, are infinitely extensible. We think this is the solution for policy expression. A well-crafted language could potentially address this problem, but we don't know of one right now. We think languages can express these specifications at the proper level. The real problem is the ability to describe when a particular operation is authorized. We need to agitate for richness of expression in commercial tools. Windows has a lot of configurable options under the hood that were difficult to access via the desktop or command line, even though an API was available. Declarative languages like Prolog might be able to help here. Exceptions are surely the difficult and important part of this problem.

We wrapped up by looking at our 1998 and 1999 predictions to see if we were late or still wrong. We still have more misses than hits. In 1999, 9 of our 19 predictions came true (or mostly true), for a 47% success rate. More of our 1998 predictions came true in 2000, but we're still looking at about a 50% success rate.

Our predictions for 2001 are:

- Peer to peer (including systems administration) will grow, then shrink. (85%)
- DSL-based ISPs will grow in popularity, then die as the tech-savvy turn their DSL to a friends-and-family ISP. (65%)
- Alternate dialtone-to-home competition will break loose (50% of the market) this year – telephone companies will have to change their business plans. (100%)
- The number of purported PDA- and managed home-appliance systems will double in the next 12 months. (75%)
- Some time this Christmas season things will go well with e-commerce. (100%)
- There'll be an e-commerce disaster some time in the next year, though. Or, a Fortune 100 company will have an above-the-fold e-disaster. (40% think it'll show up in print)
- We will have at least one Microsoft-facilitated security bug à la Melissa or ILOVEYOU in the next year. (100%)
- There'll be at least one major Silicon Valley power outage that provides above-the-fold problems for at least one company. (85%)
- Many dot-coms with otherwise profitable, viable business models will fail because their names aren't AOL or Yahoo (investor confidence will drop further). (70%)
- This is NOT the year that Silicon Valley loses its shine and people start a mass exodus. (75%)
- 802.11b will become standard on all business desktops and laptops in the next year. (80% on desktops, 100% on laptops)
- 802.11b public Internet access will be available in the top 25 US airports by December 2001. (100%)
- A huge mobile phone will be dug up on the surface of the moon.
- You will not see networks on airplanes in 2001 (not counting dial-out via airphone). (100%)
- Businesses will find their storage (at least) doubling this year, with the concomitant backup problems. (75%)
- Linux will splinter (speciate). (10%)
- The price of 17-inch flat panels will come down to $700. (100%)
- 200-dpi-resolution displays will appear on desktops. (10%)
- Gigabit Ethernet hubs will dramatically decline in price in 2001. (35%)
- Serialization of Object Application Protocol (SOAP) – RPC over HTTP; will ascend to wide acceptance. (15%)
- Official or commercial music delivery services will fail. (85%)

Finally, we listed some cool tools we're using:

- VMware
- Rethinking the world in terms of PHP, MySQL, and HTTP
- Wireless everything (802.11b)
- Some of the new load-management tools (batch queuing tools) for clusters
- 65-pound brute-breaker demolition hammer with its own cart
- PL/SQL
- XML and JavaScript
- Wiki (a very simple browser-based editing environment)
- VNC (virtual network consoles for NT)
- Baytech power strips with Ethernet access to remotely reboot via power outlet (vector for the Microsoft security outage we're predicting)
- Blackberry (two-way) pager
- Unison (file synchronizer tool between desktop and laptop, two-way comparison, etc.)
- Python
- Palm/Handspring
- ssh in IOS
- Netflow (Cisco-created protocol gaining acceptance) tools that've come out in the past year; you can now do accounting without trashing performance
- tangram (<*http://www.tangram.org*>); a Perl module that makes variables to persistent storage (SQL database)
- Herman-Miller Aeron chair; it's really worth it if you sit on your ass all day
- Authenticated Web environment that keeps the authentication token on all the time
- Win32's NetCaptor; tabbed Web-browsing Web interface
- Win32's PowerMarks; bookmark manager that has keyword-based searches
- blog (short for weblog) that logs where you go and lets you store a bunch of stuff in a column as if you were writing a letter or column like slashdot; similar to userland (see <*http://www.userland.com*> for details). Can be published as RSS feeds to JavaScript news-ticker applications.
- Newsbytes-style column

Workshop 8: System Administration Process Improvement, Benchmarking, and Metrics

Coordinators: Carolyn Hennings, Megapipe, Tom Limoncelli, Lumeta/Lucent Technologies, and Alva L. Couch, Tufts University

Summarized by Nicholas Schrieber

Caveat: this report is not intended to be minutes of the meeting.

Some workshops are odd, especially the first one on a topic, with no clear agenda or mission. The mission options at the start range from disbanding at the end of the day, saying, "OK, we've exhausted that topic," all the way to planting the seed of what eventually becomes a 500-member consortium with an executive director.

The eight of us who attended spent much of the day exploring an appropriate mission for the group. We examined and compared possible forms that a useful SA process improvement program or document could take, and we tried to determine appropriate roles for the committee within SAGE, and vice-versa. The initial mission statement we decided on for our day's work was: "Develop a framework and methodology for measuring and improving organizational maturity with respect to SA practices." At the end of the day, it was very clear that we had not actually even begun a framework and methodology, but we had reached some decisions on the shape it should take, as well as the role it should play vis-à-vis SAGE and several other SAGE-recognized programs.

Among the various models we had to consider were the System Administration Maturity Model (1993), by Carol Kubicki, the SEI Software Capability Maturity Model upon which it was based, some high-level process measurement techniques a committee member presented, as well as the Systems Administration Body of Knowledge, for which Geoff Halprin is preparing a Guide document. We also needed to consider the SAGE taxonomy project and appropriate complementarity with it.

We need to be able to ensure reproducible results, not just repeatability of process. This distinction represents an easy shortcoming many such models could harbor. Those of us who had read Carol's 1993 SAMM were in consensus that it was a good solid attempt at what was needed. There were two problems we saw, one being the state of the systems administration industry at that time, and the other is that, even now, the model as she presented it is probably not clear enough to make its widespread adoption likely. We seemed to agree that the state of the industry has advanced, and even the codification of the in-progress SA BOK suggests we're now ready to apply Carol's SAMM principles. But they need to be clearer, less academic, and probably should incorporate some changes that additional years and a larger committee can bring to the project.

The final outcome of the day was multi-part:

- We would work toward the development of a new SAMM document that is essentially reflective of the 1993 document, but greatly improved.
- We would ask SAGE to officially sponsor the project, which would in itself facilitate the status of the document toward becoming a standard. We consider that this sponsorship is already basically a fact, but requires clarification. Perhaps the best term is "SAGE-recognized" rather than sponsored.
- Further, we would ask SAGE for funding that might be necessary to provide staffing for further development of the new SAMM document.
- Carolyn Hennings would write a formal proposal, probably for presentation at the February SAGE board meeting.

Other loose notes from the day include:

Part of this job is the creation of a metrics language. This obviously dovetails very significantly with the taxonomy project. We need to take a step back and internalize existing taxonomy.

Alva Couch presented some interesting ideas on metric (measurement) systems. He pointed out that the biggest problem is too many variables.

As a high-level summary, he made reference to the Boehm approach from SE and proposed a modal approach. A major aim would be to relate life cycle cost over ideal cost (complexity of task). His conclusions:

- Robustness: ability to survive changes = cost of replacement/complexity of assigned tasks
- Efficiency: relative cost of life cycle to ideal

Over the long run the group would have to deal with complex couplings of major issues. For example, coupling with the SA BOK project is on the one hand obvious and perhaps inevitable. But it could raise false presumptions about their compatibility or the "ownership" of the SA guide project. Geoff Halprin is currently the author and owner, although he expects to allow free use of it, as long as his authorship is respected. In the Project Management field, the PMI owns the guide to the PM BOK, and as such it does not act as a private person or corporation, but more as a standards group.

Workshop 9: Taxonomy

Coordinator: Rob Kolstad

Summarized by Dave Bianchi

The goal of the workshop was to create a framework for a taxonomy of systems management to enumerate all the tasks that a systems manager might perform.

The eventual goal of the taxonomy project is to provide a list of tasks that a system manager performs along with a short description of each task. Rob would like to have a fairly complete list of tasks within three months.

The first half of the day was a brainstorming session, naming tasks and trying to group them in some meaningful way. The group came up with a grid system to categorize common activities around a task. Common activities included policy, standards, security, budgeting, and legal issues.

The second half of the day was spent focusing on just one task: "Printers." Rob proposed that a Web site be created to allow a group of people to work on the rest of the tasks, and he volunteered to create the Web pages. The group volunteered to help maintain the Web pages over the next three months.

See: *<http://ace.delos.com/taxongate>*.

BoFs

FreeBSD

Summarized by Eric Lakin

The FreeBSD BoF was chaired by two FreeBSD developers, one of whom is a core member (David Greenman). Questions were mostly directed at the two developers, with only minor help from the audience. For some questions, this was appropriate – such as the current "state" of FreeBSD and Core, and the progress of BSDi integration into FreeBSD.

"Core" is a group of developers that oversees the direction of FreeBSD development. Until recently, a person joined the core by developing and proving their ability, and then being asked to join. In the past year, there was a first-ever election of the core (by the active developers, I believe).

Expectations for the integration of BSDi/FreeBSD following the merger of Walnut Creek CDROM and BSDi have failed to materialize. A rewrite of the SMP sections of the kernel are currently in progress, with design influences from the BSD/OS code, but no actual code mingling has occurred yet.

Of particular interest was the discussion of a logging or journaled file system in FreeBSD; on-hand in the audience was Kirk McKusick, the architect of the BSD Fast File System. He gave an overview of SoftUpdates, an FFS addition currently available in FreeBSD that increases performance and addresses some of the issues that relate specifically to journaling file systems. Theoretically, at least, fscking a file system shouldn't be necessary when softupdates are enabled, but fscking is currently still done "just in case" until the code and theory have proven themselves.

The remaining discussions mostly related to individual problems with FreeBSD, most of which boiled down to "more information needed."

Getting Plugged Into Sendmail

Mike Smith, ActiveState

Summarized by Theo van Dinter

This BoF was actually titled "libmilter: Getting Plugged Into Sendmail." Libmilter is included in Sendmail 8.10 and above, and gives access to the Sendmail Mail Filter (milter) API. Milter gives hooks into every stage of an SMTP transaction, and lets you perform custom actions based on any part of said transaction, including the content of the message. While some of this is possible without milter (for example, procmail lets you filter after message acceptance), milter gives you more functionality before the message has been accepted by the mail server.

Some things you can do with milter:

- **Mail archiving:** selectively archive messages based on specific criteria
- **Spam control:** deny mail delivery based on your programmed specifications
- **Content rewriting:** add new headers or change existing content
- **Virus scanning:** verify that incoming/outgoing attachments are clean

All of the examples shown during the BoF were written using ActiveState's PerlMX product, which allows these filters to be written in Perl instead of the usual C. It is a commercial product but is available for free to individuals and educational sites. For more information, see *<http://www.activestate.com/PerlMX/>* and libmilter/README from the Sendmail distribution.

BoF: NetBackup

Curtis Preston, Collective Technologies

Summarized by Steve Hanson

Due to some scheduling confusion, Preston was late coming to this BoF, but in normal USENIX fashion the attendees (the room was very full) proceeded to run the BoF themselves, and a lively discussion of various NetBackup technical and support issues began. This mostly revolved around the typical vendor complaints and a few specific issues with backup scheduling and system security in NetBackup.

After Curtis arrived, the discussion quickly became more of a tutorial, which was quite interesting. Most of the information was on bpgp, an undocumented command in NetBackup which is used internally to copy files between systems. Although one could consider bpgp to be a security hole (at least on NetBackup systems which are not using the built-in authenticated communication methods), it is a very useful tool for copying configuration files and other information between systems which are NetBackup clients or servers. Some typical uses include dissemination of exclude files and any other sort of configuration information. In all, this was a good and worthwhile session, though it could have been much more effective if it had been scheduled for longer than an hour.

Postmaster

Strata Rose Chalup, VirtualNet Consulting

Summarized by Theo van Dinter

This BoF was intended to get people together to talk about SAGE's upcoming "Role of Postmaster" booklet. The booklet is meant as a "best-practices" guide for system administrators in charge of mail systems, but will not be limited to any particular mail transport agent (MTA). The booklet is being co-authored by Strata Rose Chalup and Brian Kirouac. There were far too many suggestions and planned topics to be listed here. This booklet might actually have to be split into multiple booklets to cover all of the different parts of the postmaster role. If you're interested in becoming involved with the booklet (either by making submissions/suggestions or just seeing what everyone else is sending in), please contact Strata via email: <*strata@virtual.net*>.

Sendmail – Open Source / Commercial

Summarized by Brian Kirouac

Open Source 8.12:

- new IO library
- new memory management to limit forks
- no longer need to get sfio for security add-ons

Open Source 9.x:

- global optimization
- some things from qmail and postfix
- not a monolithic program, but will not go as far as postfix ("postfix has too many small processes")
- most processes will be threaded
- some platforms will be dropped
- new configuration files
- ambitious goal for performance: machines two or three generations out will be able to handle 100 million messages a day, and this will work on clusters

SMI Advance Messaging Server:

- release 2.5weeks ago

SMI:

- Eric is being cautious
- he hasn't killed the new CEO yet
- still in the process of going through ripples of new CEO
- "still going wonderfully"

The last CERT advisory for Sendmail was in January of 1997.

LISA 2000 Solaris Docs

Summarized by Steve Hanson

The Solaris Docs BoF was held by several of the Sun documentation developers. The primary purpose of the BoF was to solicit input from Solaris Documentation users and to answer questions they posed.

Several issues were raised during the BoF, including:

- Jumpstart documentation is out of date. Attempts are being made to improve it for Solaris 9.
- There were several complaints about the navigation on sun.com in general, and on docs.sun.com in particular.
- Several users wanted documentation for End-of-Life'd products kept online so that users of older hardware can still obtain information on their products.
- One of the primary improvements made in the last several documentation releases was to try to include more task-specific information. This was prompted by user requests.

There should be several upcoming improvements in the BigAdmin site at Sun, including tying BigAdmin into the other documentation sites and providing more information on upcoming Solaris releases. BigAdmin is a useful Solaris administration resource and is located at <*http://www.sun.com/bigadmin*>.

Email comments on documentation can be sent via the comment alias on docs.sun.com, or by sending directly to the documentation developers:

<*cindy.swearingen@sun.com*>
<*julie.nelson@sun.com*>
<*kathy.slattery@sun.com*>

UNIX on Handhelds

Summarized by Steve Wormley

Steve Wormley led a BoF on "UNIX on Handhelds and Wearables" Wednesday evening. It turned out to be mostly on wearables by a guy from MIT (next time, just handhelds will be covered).

It was an interesting group with lots of questions and details of what is being done in the field. The head-mounted displays are getting smaller, the networking (802.11 and CDPD) is getting more widespread, and the devices are getting more powerful.

All in all, it was interesting but a bit short on handhelds, and they definitely still aren't for everyone, and not even for most of us yet.