;login:

THE MAGAZINE OF USENIX & SAGE

February 2002 • Volume 27 • Number 1

inside:

CONFERENCE REPORTS

LISA '01: The 15th Systems Administration Conference

USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

conference reports

This issue's reports focus on on the 15th System Administration Conference (LISA 2001) held in San Diego, California, December 2-7, 2001.

Our Thanks to the Summarizers:

Mark Burgess
Marguerite Curtis
Yolanda Flores-Salgado
Liliana Hernandez
Doug Hughes
Steven Levine
Mark Logan
Armando Rojas-Morin
Joel Sadler
Michael Sconzo
Josh Simon, Coordinator
Tim Smith
Crystal Stockton
Jeff Tyler
Jin-ping Wan
Jason Wertz
Garry Zecheiss

15th Systems Administration Conference (LISA 2001)

San Diego, California

December 2-7, 2001

KEYNOTE ADDRESS

Slime versus Silicon

Greg Bear

Summarized by Steven Levine

The keynote address at LISA 2001 was given by much-awarded science fiction author Greg Bear. Mr. Bear spoke about new paradigms in our understanding of biological systems that encourage us to view cells, particularly bacterial cells, as nodes in a network, independent supercomputers that cooperate, communicate and use transfer media to alter themselves and each other. Who is particularly suited to understand and talk about this paradigm? System administrators.

Well, system administrators and science fiction readers, who are also characteristically open to listening to visionary worldviews underscored with conspiracy theories. Mr. Bear noted right up front that the LISA crowd was indistinguishable from the crowd at the largest science fiction conventions. He later extended the comparison by noting that science fiction fans are like children in that they are "eternally curious and not interested in fashion." Sci-fi fans, like sysadmins, are below the radar level of most of society. Mr. Bear's understanding of system administrators, and particularly the self-image of system administrators, was stunning.

The real LISA, says Mr. Bear, exists in the acreage surrounding the Town and Country Resort Hotel. LISA is the Laterally Integrated Stochastic Anticipator, the bacterial computer network in the soil system. A bacterial network is a slime machine, a bacterial supercomputer: a cell has three billion base pairs, each a computational unit. Bacterial cells communicate and cooperate.

Greg Bear

If a cell is a supercomputer and a node in a network, who administers the cells and the network? Who is nature's sysadmin? Not DNA, administering from the top down, as we have been taught. Cells can absorb information and change proteins by means of viruses and retroviruses. The viruses are the methods of communication between the nodes in a network. They seem to be necessary, which is why they are so tough to get rid of.

In a biological system, everything is a kludge. Change, however, is a necessity. Mr. Bear spoke of Barbara McClintock's work with jumping genes, DNA that changes systematically. A genome is like an ecosystem, and a gene must cooperate with hundreds and thousands of other genes; we know this from the evidence of embryological cells not cooperating, and the resulting problems.

What we are looking at is evidence of "social biology" (social biology – not sociobiology). Biology is social from the genome on up. We are now finding the very language to describe how systems work. And who speaks this language already? System administrators.

The old paradigms of biology – randomness, DNA writing to RNA in a process that never reverses – are "dead wrong." Yet the paradigms are still being taught. We need, instead, to look at the way we put systems together to gain some understanding of the problems of biology.

So, Mr. Bear says, go forth and study biology. You will learn how to administer systems; slime has been doing it for billions of years. All biological systems are networks of users, and users all have different priorities. We must learn to be open, to think like children, in order to deal with networks of users.

REFEREED PAPERS

STIRRING THE MATRIX: ORGANIZATIONAL SYSTEM ADMINISTRATION

Summarized by Tim Smith

Defining the Role of Service Manager: Sanity through Organizational Evolution

Mark Roth, University of Illinois at Urbana-Champaign

The presentation began with Mr. Roth defining a service as a collection of tools that allow users to do their jobs. He then reviewed the evolution of services over the past decade. Services in the early nineties were homegrown tools where no distinction was really made between the system and the services provided. In the late nineties, client-server applications, where there was some distinction between the system and the service provided, took the place of in-house software. Services today are taking the form of black boxes where the service software is distinct from the system it runs on, and users are not aware of the type of system used to run the service. The thesis of the paper is that in this new environment the role of service manager should be entirely separate from that of system administrator.

As presented, the role of service manager includes several components: initial planning, production deployment, and ongoing maintenance of a service. Thereís too much work required by the components to be handled by either system administrators or developers. System administrators are too busy to begin with, and their core competency is system management not maintaining services. In addition, system administrators need to achieve economies of scale in their work, and this is not possible in service management. Service management should not be handled by developers due to their incompatible time requirements and core competency in programming.

In Mr. Roth's approach the service manager focuses on the users of the service and is responsible for ensuring that the services needed by users are available. This means giving requirements to developers for in-house software and delivering system requirements to system administrators so the hardware service will be available when needed. The advantages of this approach as seen at UIUC include improved communication with the service manager as the only communication channel between system administrators, developers, and users; increased staff retention; easier budgeting; and achievement of some economies of scale. Mr. Roth pointed out that the approach is not fully in place at UIUC but that its advantages were already being seen.

Additional information can be found on Mr. Roth's Web page at *http://www.uiuc.edu/ph/www/roth/*.

New Technologies for Small and Medium Businesses (SMB)

Degan Diklic, Venkatesh Velayutham, Steve Welch, and Roger Williams, IBM Almaden Research Center

Mr. Diklic's presentation addressed the remote outsourcing of services for multiple branch offices and small businesses. In this presentation a small business is defined as having fewer than 500 machines, and a medium-sized business is a business with fewer than 5,000 machines. Servicing branch offices and small businesses is not an attractive venture for service companies trying to make a profit, because the clients are on the other side of a firewall, dedicated lines to bypass the firewall are expensive, a full-time administrator for a small site is also expensive, and there is no generic service infrastructure in place across sites. One idea is to remotely manage small sites using VPN and remote management tools. Current solutions that implement this idea, such as OpenView or Cobalt Blue, are expensive.

Mr. Diklic's solution avoids the expense of the available solutions while still allowing sites to be serviced effectively. The solution places a communication server outside the firewalls of the remote site and the service provider. These communication servers are owned by a trusted company, IBM in this case, and are used to make a connection between usher servers on the local LANs. The usher server is used to resolve network locations of the remote machines so they can be administered as if they were part of the service company's network.

This architecture has been used in several projects in Mr. Diklic's research group. The first project was a disk expansion project that allowed additional disks to be used as an extension of an existing disk in the remote site. This allows remote sites to share disk storage and makes remote data storage possible. The second project is a backup utility for remote sites. The backup of the remote machines is performed over the network at night when the bandwidth is not being utilized. Data restoration of user data is performed by an application CD that connects to the remote backup facilities and allows the user to access the data. Authentication is also handled by the CD since it contains the customer's username and password in a secure form.

TECHNOLOGIES INDISTINGUISHABLE FROM MAGIC: ANALYTICAL SYSTEM ADMINISTRATION

Summarized by Marguerite Curtis

A PROBABILISTIC APPROACH TO ESTIMATING COMPUTER SYSTEM RELIABILITY

Robert Apthorpe, Excite@Home, Inc.

Apthorpe began by expounding on how the tutorial on probabilistic risk assessment is a technique for finding vulnerabilities. He then addressed the reasons why he wrote it, talking about the problems that system administrators face. For example, they generally have little background on systems engineering and therefore have little context for understanding formal risk assessment, or they don't know how to detect problems, or they simply make a bad decision, like putting a primary and secondary server on the same switch. After acknowledging the problems, he spoke on why risk assessment is so relevant. His list contained about five points. For example, analysis is cheaper than firefighting and a good design defends against known problems.

In the second half of his talk Robert gave us the overview of his method, which consists of eight steps: define your problems; define your system; build a logic model of system failure; decompose system information of most basic actions and events; find the minimal sequence of events that lead to failure; estimate probability of event sequence from observed or estimated data; generate measures of component importance; and sanity check the model and the results. He then showed us an event tree analysis and a sample event tree. The next topic was how to use the results and what the weaknesses of the system are. He addressed these points and why they exist. Concluding, he spoke of his future hopes and plans, other possible research topics, and other applications, such as security, capacity analysis, or insurance and risk management.This paper won the Best Theory Paper award this year.

SCHEDULING PARTIALLY ORDERED EVENTS IN A RANDOMIZED FRAMEWORK: EMPIRICAL RESULTS AND IMPLICATIONS FOR AUTOMATIC CONFIGURATION MANAGEMENT

Frode Sandnes, Oslo University College

Sandnes began by explaining his idea: the schedule would automatically maintain a system state to benefit all users and would be achieved by tools such as cfengine. It can be viewed as a mix of dynamic and static schedules where the dynamic tasks are triggered by actions and the static tasks have precedence. One of his main points is that the user can allocate any task to a particular schedule. He moved on to discuss randomized strategies and algorithms, as compared to the scheduled algorithm. His objectives consisted of finding out how the randomized schedule affects the efficiency, ability to intervene, and ability to identify the config model. He compared a deterministic management framework with a random one to determine efficiency. In the second half of his talk, he addressed malicious intervention and how it relates to his work and randomization. If the abuser wants to uncover the model, assuming the abuser can observe, randomized scheduling can make it more difficult. His idea is that the abuser will be unable to observe a sequence of events if there is only a random sequence to look at. Sandnes concluded that randomized scheduling lead to consistent performance, reduces the predictability of management abilities, and hides strategies, as well as noting that the framework is easy to implement.

THE MAELSTROM: NETWORK SERVICE DEBUGGING VIA "INEFFECTIVE PROCEDURES"

Alva Couch and Noah Daniels, Tufts University

Couch began by stating his target problem, which is to automate network troubleshooting. His dream was to create a quicker response to network response. He then began showing how it all is formed, starting with pre-declaring precedences, which must be done every time a script is added. This is a pain, so he moved on to discovering order between scripts without declaring, claiming that they will fail robustly when called at the wrong time, tell you when they fail, and won't undo each others actions. Moving further into discovering order in ineffective procedures, he talked about how it is easier to check whether a condition is present than to execute it. Trading extra executions for lack of precedence tables is cheaper and less work. The efficiency depends on the initial ordering.In the second half of his talk, Alva explained what is necessary for the commands and what he learned from it all. Each command requires awareness as to whether or not it failed (which is the easy part), must be homogeneous (which is the hard part), and must be convergent. There are no preconditions to engineering maelstroms, and they are safe to run in any sequence. He learned that causality is not a myth and cannot determine what will happen. You can determine what repaired a specific problem, not what caused it. It is not causal, but operational. He concluded with tasks he is working on and will be working on, such as a troubleshooting script.

MONTE LISA OVERDRIVE: EMPIRICAL SYSTEM ADMINISTRATION

Summarized by Joel Sadler

PERFORMANCE EVALUATION OF LINUX VIRTUAL SERVER

Patrick O'Rourke and Mike Keefe, Mission Critical Linux, Inc.

O'Rourke presented a performance comparison showing the relative merits of Linux Virtual Server (LVS) over hardware-based load balancing alternatives. Patrick began by explaining what LVS was and how it could be used to improve a Web site's performance. Their testing showed that not only is LVS quite capable of competing with hardware LB devices, it can be dramatically less expensive per request/second.

Measuring Real-World Data Availability

Larry Lancaster and Alan Rowe, Network Appliance, Inc.

If there is a holy grail in sysadmin today, it's the much-coveted five 9s (99.999%) of reliability. This somewhat eye-opening presentation showed a view of reality specifically with regard to NetApp filers. Using data gathered from customers via ONTap's Autosupport feature, Lancaster showed how they had categorized the failure data and then laid out the conclusions the data had shown. Quite surprisingly, their data showed that fewer "Operator" type errors occurred than power failures, even with clustered systems.

Simulation of User-Driven Computer Behavior

Harek Haugerud and Sigmund Straumsnes, Oslo University College

Haugerud talked about the challenges inherent in building a model to simulate user behavior on a given multi-user computer system. His presentation showed their design principles and explained some of the initial goals they had in setting out. He then presented a fairly technical breakdown of their testing methodology. In doing so, Harek showed how they had tested the simulation with a known user-action data set obtained from a third party. Their results were impressive; while they admit that this particular simulation is in its infancy, its usefulness is easily visible.

SEEING HOW THE LAN LIES: NETWORK MONITORING

Summarized by Liliana Hernandez

Specific Simple Network Management Tools

Jürgen Schönwälder, Technical University of Braunschweig

Schönwälder described the design and implementation of an SNMP management tool called scli, which provides an efficient-to-use command-line interface to display, modify, and monitor data retrieved from SNMP agents.The SNMP management tools available today fall into one of the following five categories.

- Generic low-level SNMP tools
- Generic low-level SNMP APIs
- Generic MIB browsers
- Generic monitoring tools
- Generic management platforms

But the author still often feels uncomfortable when trying to use them; for example, the generic tools often do not understand the relationships between MIB objects.The software design addresses five key requirements: extensibility, robustness, maintainability, efficiency, and portability.The package uses the glib library to archive portability and to reuse generic data structures such as list and dynamic strings. The SNMP engine gsnmp has been derived from the gxsnmp package and was subsequently modified to fix bugs and to improve stability. The SNMP engine itself uses glib. The interpreter core and some command implementations also use the libxml2 library to create and manipulate XML documents.The SNMP engine does not yet support SNMPv3 security. The code generator can be improved in many ways. The biggest limitation right now is the restriction that stubs can only operate on table rows or groups of scalars.

Gossips – System and Service Monitor

Victor Götsch, Albert Wuersch, Tobias Oetiker, Swiss Federal Institute of Technology

Gossips is a modular client-server-based system monitor. Gossips not only reports problems but suggests solutions to the problems by consulting a knowledge base. The monitor software is written in object-oriented Perl.The goal in this project was to address some of the problems found with existing solutions like SNMP, Big Brother, Swatch, Spong, and PIKT.

- Big Brother is good in design, scalability, and messaging. It is okay in configuration and is extensible.
- Swatch is very extensible. It is okay in configuration, design, scalability, and messaging, but it is not modular.
- Spong is good in design, scalability, and messaging. It is okay in configuration, extensibility, and modularity.
- PIKT is good in configuration, design, scalability, extensibility, and messaging, but it is missing modularity.
- gossips is good in configuration, design, scalability, extensibility, modularity, and messaging.

The distributed architecture of gossips builds a scalable monitoring system. Through its flexible and central configuration environment, together with its command-line module, gossips is easily maintainable.The object-oriented design of gossips builds a flexible and well-defined framework for developing new monitoring tasks. The concept of separating data acquisition and data analysis makes defined monitoring tasks reusable and provides the possibility to build combined tests. The knowledge base allows one to archive solutions to known problems in one place and to integrate the knowledge of the system manager. By including cfengine, gossips could be extended into an automated repair tool.

The CoralReef Software Suite as a Tool for System and Network Administrators

David Moore, Ken Keys, Ryan Koga, Edouard Lagache, kc claffy, CAIDA

CoralReef is a package of device drivers, libraries, classes, and applications and provides a suite of tools to aid network administrators in monitoring and diagnosing changes in network behavior. CoralReef offers a unified platform to a wide range of capture devices and a collection of tools that can be applied at multiple network levels. Its components provide measurements on a wide range of real-world network traffic flow applications, including validation and monitoring of hardware performance for saturation and diagnosis of network-flow constraints. CoralReef can be used to produce stand-alone results or data for analysis by other programs. CoralReef reporting applications can output in text formats that can be easily manipulated with common UNIX data-reduction utilities, providing enormous flexibility for customization in an operational setting. CoralReef provides a balanced collection of features for network administrators seeking to monitor their network and diagnose trouble spots. By covering the range from raw packet capture to real-time HTML report generation, CoralReef provides a viable toolkit for a wide variety of network administration needs.

LEVEL 1 DIAGNOSTICS: SHORT TOPICS ON HOST MANAGEMENT

Summarized by Jeff Tyler

Global Impact Analysis of Dynamic Library Dependencies

Alva Couch and Yizhan Sun, Tufts University

SoWhat is a tool for analyzing and tracing library dependencies in a large distributed environment. ldd can tell you what libraries any given program will load, but how do you determine the total set of programs in a large environment that might require a given library? The simple answer is that you don't, and thus one can never delete a library in a complex environment without a significant chance that some program somewhere in the environment will then break. Therein lies the path to library rot. SoWhat attempts to address this problem by analyzing and cataloging all library dependencies in just such a large environment. SoWhat currently runs on Solaris 7/8, with a Linux version promised. It's written in Perl and requires MySQL. It is freely available at *http://www.eecs.tufts.edu/~couch/sowhat*.

Deriving Tools to Administer Domain and Type Enforcement

Phil Kearns and Serge Hallyn, College of William and Mary

In this context, Domain and Type Enforcement (DTE) means a mechanism to provide fine-grained mandatory access control beyond the level provided by a conventional UNIX kernel. DTE systems normally use a rather densely populated text file as a control and policy establishment tool, and simple typos in these files can have a catastrophic effect on the surety of the system. Phil and Serge have addressed this issue by producing two tools to aid in administration of DTE configuration files and to provide a graphic view of system objects that any controlled program might interact with. The tools are called DTEedit and DTEview and are available at *http://www.cs.wm.edu/~hallyn/dte*.

Solaris Bare-Metal Recovery from a Specialized CD and Your Enterprise Backup System

Lee Amatangelo, Collective Technologies, and Curtis Preston, The Storage Group

Building on the success of their popular CART tool (first presented at LISA 2000), Lee and Curtis have constructed BART, the Solaris Bare-Metal Recovery Tool. CART was a system-specific tool, but BART is a networked version that can deal with multiple machines using an enterprise backup system and a single CD. It currently works on Solaris only due to Jumpstart dependencies and will operate with both Legato and Veratis NetBackup, although there are some Veratis issues.

Accessing Files on Unmounted Filesystems

Willem A. (Vlakkies) Schreuder, University of Colorado

Now *this* is a very useful utility. It is used for recovering files from unmounted disks and general bunged-up disk spelunking. If you've ever spent any time in fsdb you'll appreciate what went into the construction of this tool. It works like cat and has both stand-alone (ruf) and callable library (libruf) versions; it can automatically determine the location of alternate superblocks and perform other useful disk tricks. It currently works on *BSD, Linux, Sun OS/Solaris, and HP-UX. It is available under the BSD license at *http://www.netperls.com/ruf*.

TO YOUR SCATTERED PCS GO! DISTRIBUTED CONFIGURATION MANAGEMENT

Summarized by Tim Smith

Automating Infrastructure Composition for Internet Services

Todd Poynor, HP Labs

The automatically configured data center is an environment where the efficient redeployment of resources in the data center is required in order to meet changing demand. In this environment federations of resources from autonomous compute systems work together to provide a service. Such an environment does not exist today, but the framework presented by Poynor is a result of research and industry activity.

Poynor's talk presented a framework for composing Internet services from component services. In this framework, system administrators issue instructions to the computing resources on what services to deploy. The services that can be deployed are grouped into contexts that allow services to cooperate to achieve a larger goal and automatically discover new members of the context. Information is also provided to the framework about deployment changes in the context that allow services to adjust and reconfigure relationships so the proper services are still provided.

Changes to the service deployment must be specified by an administrator or automated process. The affected resources are notified of the change, which allows them to start and stop component services and possibly reboot machines into new environments. The services add and drop relationships based on the current environment. Once the instructions have been provided, Internet services are stopped and started without administrator intervention. Mr. Poynor gave an example of what would happen when adding a new machine into a Web server farm.

The framework will require a protocol suitable for all hardware and software. The current prototype used by Mr. Poynor's group is an extension of the IETF Service Location Protocol. The framework, implemented with the protocol, extends UNIX system startup scripts or the Windows Services applet to allow the machine to be dynamically configured.

The PowerPoint presentation of the talk can be found at *http://www.hpl.hp.com/personal/Todd_Poynor/*.

TemplateTree II: The Post-Installation Setup Tool

Tobias Oetiker, Swiss Federal Institute of Technology

TemplateTree II addresses the problem of adding new machines into an environment. Each new machine needs an operating system and software packages installed and any site-specific configuration changes. All of the modifications can be made to a base operating system install using cfengine.

Oetiker's presentation covered how TemplateTree II can be used to generate the cfengine.conf files necessary to apply the modifications to a base system and POD-style documentation of the components that make up the modifications. TemplateTree II uses tools to set up subsystems including network configuration, the AFS client, and SSH configuration. Metadata are added to each subsystem description, which also includes the component configuration files, so TemplateTree II knows what each subsystem does.

Configuration of a base system using TemplateTree II involves specifying which subsystems to apply to the system. The metadata of each subsystem are used to create the cfengine.conf file. Once cfengine and the generated configuration file are installed on the base system, cfengine will finish installation and configuration of the subsystems, and the system will be ready for use in the system administrator's environment.

More information on TemplateTree II can be found at *http://isg.ee.ethz.ch/tools/*.

The Arusha Project: A Framework for Collaborative UNIX System Administration

Matt Holgate, Glasgow University, and Will Partain, Arusha Project

The Arusha Project allows system administrators at modest-sized sites to collaborate with one another on a large scale using the Internet. The presentation focused on ARK, an XML-based configuration language that can be used to describe system administration objects. An object is anything an administrator interacts with, including software packages, systems, and teams of administrators.

Partain's presentation showed how a software package can be described by different administrators using the configuration language. Each administrator began with different description fields, which include the package name, any administrator comments, the options used to build the package, and many other fields. Partain's presentation showed how isolated system administrators can collaborate with a few other system administrators via the Internet to exchange their package descriptions. Each administrator can take the descriptions from others and plug useful fields into his or her own description. As the updated descriptions of the package are shared, the package description at each site is improved.

Partain's examples showed how packages can be parameterized and inherited by other packages. He also showed how macros are created in the configuration language and clean up the parameterization of an object.

The Arusha Project home page can be found at *http://ark.sourceforge.net/*.

HUMAN INTERFACE: TIMELY SOLUTIONS

Summarized by Jeff Tyler

Lexis: An Exam Invigilation System

Mike Wyer and Susan Eisenbach, Imperial College

This paper won the Best Applied Paper award.

Wyer and Eisenbach faced the problem of converting (temporarily) a large number of Linux workstations to a highly secured configuration to allow students to take programming examinations while still maintaining network connectivity to a central server to collect test data. After the exams are over, the workstations have to be reverted to a more normal state. This transition has to be done repeatedly over the course of a semester.

They solved this problem with a combination of local and remote lockdown tools and a secured client-server configuration built around SSH and ipchains. They took advantage of tricks like mounting the root file system without suid bits active and heavy use of custom run levels. To activate a Lexis client one simply changes to run-level 4 and the rest is automatic. The Lexis server, on the other hand, is a dedicated box with more traditional system security and maintains "Lexis state" at all times.

Mike provided a lot of detail about development and testing of the system, building confidence with students and staff, and discussed how they overcame problems such as scaling and system reliability. The Lexis system is in use at Imperial College and may be obtained under GPL at *http://www.doc.ia.ac.uk/~mw/lexis/*.

Javamlm, A Customizable Mailing-List Manager

Ellen Spertus, Mills College; Robin Jeffries, Sun Microsystems

The authors attempted to tackle a problem with which all of us are familiar: the fact that a successful mailing list soon generates volume levels that overwhelm some users, who then drift away. They studied and rejected some traditional approaches such as static sublists and user filtering and implemented a dynamic sublist approach (threads). This approach presumes nothing on the part of the mail client (e.g., no filtering) and allows the user access via a Web interface to adjust subscriptions and preferences. Javamlm works with qmail to do the heavy lifting (e.g., thread distribution) behind the scenes.

This effort was strictly a prototype and the authors intend to fold their work into mailman, the GNU mailing list manager.

GEORDI: A Handheld Tool for Remote System Administration

Stephen J. Okay, Road Knight Labs, and Gale E. Pedowitz, Protura, Inc.

This was an interesting talk and a fascinating paper. The authors detail their efforts to build a useful (and relatively secure) sysadmin remote access tool on a Palm Pilot. Quoting directly from the paper, "At base, GEORDI is a forms based UI wrapper for an RSA/DSA ssh connection to a remote host running sudo." GEORDI also understands about 60 UNIX commands and can recover state from previous sessions (scripts and commands built with its command-builder tool).

I suffer from the same bias that I suspect most of us do – if I can't get to a shell, then I tend to suspect the utility of the tool. The authors, sysadmins themselves, have gone a long way toward addressing this concern and producing a useful tool, given the inherent limitations of the platform. If you need to perform highly mobile systems administration, then GEORDI may very well be useful to you.

GEORDI is available under GPL at *http://www.GEORDI.org*.

ADAPTING THE COLLECTIVE: SHORT TOPICS ON CONFIGURATION MANAGEMENT

Summarized by Tim Smith

Pelican DHCP Auto-Registration S ystem: Distributed Registration and Centralized Management

Robin Garner, Tufts University

Garner presented the DHCP registration system implementation used at Tufts University. The university has a class-B address space and about 9,500 systems. Six DHCP servers are used to service these machines. After experimenting with DHCP registration systems from 1997 to 1999, Garner was involved in the development of Pelican. Pelican was developed to address scaling issues not handled by the other registration systems.

Before registration the host is given an IP address from an untrusted portion of the address space. Pelican works by deriving a hosts MAC address when they register with the Web utility. The MAC is put into the dhcp.conf file, and the DHCP service is restarted every fifteen minutes to pull in the new MACs. When the new machine is restarted after DHCP has its MAC address, it is able to obtain an address in the trusted address space and see the entire network and Internet. Pelican also has functions to add and purge leases from the database, and to purge old machine registrations from the database. Garner concluded with performance results of Pelican.

A Management System for Network-Shareable Locally Installed Software: Merging RPM and the Depot Scheme under Solaris

R. P. Channing Rodgers and Ziying Sherwin, US National Library of Medicine

Network-shareable software has several problems. Packages are not independent of one another, and there is a need to allow host-specific "custom" packages. While locally installed software allows for host-specific packages, it is a large burden to maintain.

After covering previous work in the area, Rodgers noted that depot was selected for the project because it is a simple format, RPM was selected because its format is open, and each format contains information that complements the other.

Rodgers' presentation concluded with future work for the project. The RPM and depot functionalities should be coupled. In order for the combined format to work well, the RPM database needs to be modified to allow for dependency checks across the network. The documentation in the two formats should also be merged. Other RPM enhancements that would require modifying the RPM code would also be useful.

File Distribution Efficiencies: cfengine Versus rsync

Andrew Mayhew, Logictier, Inc.

The native file transfer protocol in early versions of cfengine did not work. To overcome this problem Mayhew put rsync in place to transfer files between systems. When the cfengine file transfer was fixed, the possibility of comparing it against rsync motivated Mayhew to compare the two.

The experimental setup used two machines on the same segment and compared the performance of unencrypted cfengine file transfers, encrypted cfengine file transfers, and rsync. Files transferred in the experiments varied in size from 128 kilobytes to 2 megabytes.

In the experiments rsync performed better on large file transfers, while cfengine was better transferring smaller files.

CfAdmin: A User Interface for cfengine

Charles Beadnall, W. R. Hambrecht, and Andrew Mayhew, Logictier, Inc.

Mayhew presented CfAdmin, a user interface for cfengine designed to allow facilities staff, release engineers, system administrators, and network operators to preview and edit information regarding systems and the software for them. Each of the groups needed to use cfengine for their work, so a common interface was created.

CfAdmin uses cvs for version control, cfengine for host management, and netcool for network monitoring. Apache with secureid is used for the interface. The interface allows the different groups to perform host entry and software location entry (e.g., binary paths, etc.). The system administrator then uses the interface to configure a system before deployment using software information from the release engineer. Facilities are able to use the interface to install the machine in the proper location.

A cfengine.conf file is generated by CfAdmin and then pushed out to all hosts based on the information entered. The automatic generation and distribution of the configuration file eliminates human error while updating the configuration file and makes cfengine easier to use.

WORK-IN-PROGRESS REPORTS

Summarized by Jeff Tyler

Email Redo Logs for User-Initiated Restores

Rich Graves, Brandeis University

Email redo is more concept than code at the moment but in use nonetheless. In a nutshell, create two mail spools and declare one read only (at the user level). Allow the users to recover individual pieces of mail from the r/o spool after they zap them in error in the traditional r/w spool. Supports UW-IMAP, currently running on Linux. Roll the spools on a systematic basis and expire the r/o spool at a reasonable point. A very clever idea that only requires simple changes to the local mailer to write both spools on incoming mail, some pointers to allow users to recover their own files, an expire mechanism for the r/o spool, and a LOT of disk space.

Bringing Undo to System Administration: A New Paradigm for Recovery

Aron Brown, University of California, Berkeley

Very much at the concept level at the moment, undo would provide the ability to "recover" at will to almost any point in time. Requires a LOT of prior planning. The concept is based on the system-recover three Rs: rewind, repair, replay. It sounded very transactional but aimed at base OS, not databases. It will be interesting to see where this one goes.

Operational Failures in Large-Scale Internet Services

David L. Oppenheimer, University of California at Berkeley

This case study – the first in what is hoped will be a series of case studies involving large-scale Internet-based enterprises – involved an Internet-based network storage company. Among the findings of interest was the fact that most failures were due to human error, which accounted for 33% of the events studied. The next largest cause of failure was listed as software failure and this was charged against the fact that software used was mostly home grown and being operated without rigorous change control. Oppenheimer is actively seeking companies or organizations willing to participate in his study and guarantees that no one will ever learn your name if you sign up.

Verdad

Jeff Kellem and Jeff Allen, tellme.com

Verdad is a central configuration store. It understands inheritance, versioning, and is based upon MySQL and Perl. It controls software, DNS and DHCP data, and user ACLs. It has a r/w Web interface. Sounds useful.

Establishing an Associate of Applied Science in Computer Security Degree

Will Morse, North Harris Montgomery Community College

Morse is working on establishing this degree program in the Texas community college system. He has a core curriculum, two OS-based courses, and a "lore" segment. He's looking for ideas and wants to draw upon the experiences of others in this area. His goal is to produce an entry-level security person who can meet 80% to 90% of the security requirements that a small business might have.

Installing Linux in under Three Minutes

Paul Boven

The keywords for this talk are PXC, BIOS re-direction, serial console, and remote power control. Then stir in DHCP, TFTP, and NFS. Ever dig into the Sun OS or Dec remote workstation build procedure? Then you understand 90% of this WIP. The other 10% is mostly in getting PC hardware smart enough to boot off the wire. Paul seems to have a very nice scalable version of this ever-popular hack. It used to take us longer than three minutes to do this in Athena, but disks and the wire were both slower then. Boven can be contacted at *p.boven@sara.nl.*

The Condor Cluster Tool

Erik Paulson

It's a bird, it's a plane, it's a big cycle stealer! Currently at 1000 CPUs and growing daily. It's been adapted to interactive use and taught some manners (from the standpoint of the people whose cycles you are stealing). Currently uses only advisory locks and has some issues with reservation timeouts (they don't). In the words of it's author, "It's not too secure . . .." V2.0 is under development and will address security with Kerberos, have resource reservation timeouts, and stronger-than-advisory locking. Don't let those spare cycles go to waste, folks.

A Portable Linux Cluster

Mitch Williams, Sandia National Labs

This was an extremely neat hardware hack. Visualize this: a 4-banger Linux cluster in a tiny (5.3" x 5.3" x 13") custom rack. Weighing 15 pounds, it has its own little packing case that fits in the overhead rack on a plane. Built around the PC104 card buss system. Each CPU has 128 Megs of memory, and the whole thing is driven by a 50W power supply. They built it in a month from scratch for about $5,000. Seems like Sandia needed a PORTABLE teaching and demo system that could do some serious parallel processing. You have to see this thing to appreciate it – it's a jewel-like creation. I asked Mitch what he'd change if he had to do it again and he said, "I might make the power supply a tad larger, like 75W or maybe even 100." Mitch asked that, in addition to his coworkers at Sandia, I give credit to the Parus and Advanced Digital Logic corporations for all their help. This was the winning WIP and a very well deserved win in my opinion.

INVITED TALKS

CNN.com: Facing a World Crisis

William LeFebvre, CNN Internet Technologies

Summarized by Joel Sadler

It was a presentation that few will forget. LeFebvre showed in great detail how CNN.com dealt with the traffic load created by the 9/11 tragedy. He opened the talk by presenting some introductory information about how the CNN.com operations actually run. The group that handles the hardware for CNN.com also performs the same function for quite a few other Turner Web sites, including WCW.com, SI.com, TBS.com, and CartoonNetwork.com. All of the Web serving hardware for the various sites is identical, which allows for very simple "swings" of hardware among the various sites when required for special events or other heavy traffic times.

Moving on, Bill presented a time line with a stunning array of data about their traffic load. He showed that their inbound HTTP requests doubled every 7 minutes, with a starting metric of 84,719 hits/minute at 08:45 (all times EDT). By 09:00, they were already up to 229,006 hits/minute! At this point, the traffic monitoring software was shut down for several hours to remove any and all unnecessary load from the network and servers.

Meanwhile, the staff was scrambling to borrow servers from other sites so that CNN.com could continue to serve the exponentially increasing load. Starting with 10 servers at 08:45, they were able to increase that number to 52 by 13:00, including an amazing swing of 20 servers in a half-hour period. In addition to the server moves, they were minimizing the contents of the home page in an attempt to meet the overwhelming demand. At their lightest point, there were only 1247 bytes of HTML, with one small logo and a small picture.

Other interesting statistics of note: CNN.com's previous high traffic record was on 11/8/2000, the day after the US presidential elections. It reached a peak of about 1.2 million hits/minute for a total of about 139 million page views. On 9/11/2001, CNN.com successfully served about 1.1 million hits/minute for a total of about 305 million page views, not including the several hours that monitoring was deactivated. Their best guess as to the actual peak was about 1.8 million hits/minute.

Security for E-Voting in Public Elections

Avi Rubin, AT&T Labs — Research

Summarized by Crystal K. Stockton

Rubin talked about his previous experience with developing a system of e-voting for a public election in Costa Rica, where the voters needed to vote in their home districts. The government has already provided public transportation for those not living in their home district and has made the voting day a national holiday to ensure that everyone has a chance to vote, yet e-voting would help those unable to travel.

Problems they encountered were that each district had a different ballot, adults had little experience with using a mouse, and the computers were limited to regular voting districts. The written software took into account the problem of several different types of ballots, and it was suggested to use touch screens or light pens to compensate for the mouse, but the government did not have the extra funds. Other problems during testing were that all votes were recorded correctly for the primary Republican and Democratic candidates but wiped out the votes for other candidates. A power surge switched all votes from one candidate to another, and without an audit trail it was impossible to redistribute the votes.

Rubin also outlined possible threats to the system, social apprehensions, and technical issues. What types of consequences would there be if an attack were successful? How motivated are these attacks? What type of voting coercion, sale, or solicitations will there be? Is this a secure platform to use? What will the availability of the network be? How can it verify that a living person is voting?

Zope

Michel Pelletier, Digital Creations

Summarized by Armando Rojas-Morin

Zope is an open source Python-inspired object-oriented Web environment. With Zope, Web sites are developed through the Web itself.

One of Zope's strengths is its security system. Users have roles assigned, and actions are protected by permissions. It provides a file-system-like structure with a root folder. Permissions can be asigned to each subfolder. It's a good way to delegate. Because everything is done via the Web, things are not actually files in the file system.

Zope is more than just a server (HTTP, FTP, pcgi). Zope's philosophy is that data, login, and presentation should each be a different layer. This keeps all of the designers and developers happy by never violating their layer.

Zope offers relational database integration and content objects for the data layer; Python, Perl, and SQL for the scripting layer; and page templates and DTML templates for dynamic presentation.

2001: A Communications Anniversary

Peter Salus, Matrix.net

Summarized by Joel Sadler

Peter Salus gave a wonderfully humane and informative talk. He discussed the technological distance covered, primarily in the 20th century, to get us to the current level of communications dexterity. Peter also made mention of some of the leading innovations necessary to such advancement, such as the mechanical calculator, the telephone, and transatlantic radio messaging. He linked seemingly unimportant items together and illustrated their relevance in further advancing communications capabilities.

High points included the first transatlantic radio message in 1901; the first transistor in 1951; Clarke's short story "The Sentinel" in 1951 and its movie adaptation *2001: A Space Odyssey* in 1968; the birth of UNIX and of Linus Torvalds in 1969; Lyons' explication of the UNIX kernel in 1976; and the release of both PGP and Tim Berners-Lee's first HTTP work in 1991.

If I Could Talk to the Animals: What Sysadmins Can Learn about Diagnostic Skills from Another Profession

David N. Blank-Edelman, Northeastern University

Summarized by Crystal K. Stockton

Blank-Edelman began by comparing his profession to that of mechanics and doctors and pointed out several reasons why their work is completely different than that of a system administrator. He went on to say that a more accurate comparison would be to the profession of veterinarians. The reason comparing a system administrator to a mechanic is not accurate is that mechanics can remove and replace parts in determining a problem and fixing it. Their world does not fluctuate much, and they have various instruments available to help with diagnostics. Blank-Edelman believes comparing system administrators with doctors doesn't work either, because doctors treat others of the same species as themselves. They have the luxury of communicating with their patients. Veterinarians have the most similar profession to systems administrators because they work with a large variety of species, cannot easily remove and replace parts, and collect diagnostic information from a third-party source.

Blank-Edelman then talked about different types of decision-making. One type described in depth was deductive logical thinking. This is a classical approach to decision-making, which is usually what people are taught at an early age. Another type of decision-making is naturalistic decision-making where the environment influences your decisions. Types of environmental variables are time, pressure, high stakes, and experience.

After explaining different approaches to decision-making and explaining the reasons for trying to mirror sysadmin and user relations, Blank-Edelman showed a clip from the movie *Doctor Dolittle*. While watching this clip, he explained how a sysadmin's job is similar to a veterinarian's. This was a fun and light way to compare the two professions.

To review Blank-Edelman's slides and learn more about his research, refer to the Web site *http://www.otterbook.com*.

The Problem with Developers

Geoff Halprin, e-smith, Inc.

Summarized by Yolanda Flores-Salgado

"There is a problem with developers. They don't develop maintainable, production-ready, manageable code."

Maintenance is 70% of the software development lifecycle, but most developers can't maintain their software. They suppose and assume a lot, don't consider changes, don't provide enough documentation, and so on. Developers need to be re-educated, but sysadmins can't re-educate developers.

The only way is not to accept the product if requirements are not complete. If an application doesn't satisfy our needs, don't accept it, don't use it. Sometimes this is not possible, but if we could do it, our lives would be better.

An application has a life cycle: install, configure, manage, monitor, build, update, and de-install.

Developers need to know configuration standards. We need standard configuration files and logs. We also need a partitioning of file types – not all files are equal, and to improve file access, each location should be specified separately and set by the administrator.

Applications should be separated from system areas, and developers should give us the choice whether to isolate the application in a simple hierarchy and to have separated data areas.

Configuration management is very important. Proper configuration gives us the choice to manage the application's behavior. For simplicity of use, everything should go in one file if possible.

Sysadmins need control over:

- How to stop and start the application
- Application requirements
- User Management

Sysadmins need monitoring applications. We need standard logs, and logfile management (files rotating, configuring logs, etc.). Sysadmins also need backups. How easy is it to backup and restore the application?

Error handling – does the application trap potential errors? Does it report errors in a consistent format?

Sysadmins need installation instructions. Halprin said, "It is a sad statement that most software installations are done by executing the vendor installation scripts without question."

An application should provide documentation to install, de-install, and upgrade it. We need to understand installation procedures, but, most importantly, we need to be able to control them.

PHP for System Administration

Shane Caraveo, ActiveState

Summarized by Yolanda Flores-Salgado

PHP is a Perl/C-like scripting language designed specifically for the Web. It can be used on almost all platforms (e.g., UNIX, Windows, and MacOS), and it can be used either as a stand-alone language or as embedding SGML, XML, ASP, or JavaScript. PHP provides a lot of extensions supporting all of the commonly used databases (postgres, MySQL, Generic database, oracle), system protocols, and distributing processing.

PHP is easy to use and learn for sysadmins, especially if they are familiarized with C or Perl, and because it is Web browser-oriented, it can run anywhere. Web interfaces are also easy with PHP.

PHP can be (but is rarely) used for system administration. Since GUI interfaces are easy in PHP, PHP can be useful in building interfaces to delegate some sysadmin duties to non-sysadmins.

Some interesting PHP-related sysadmin projects are:

- PhpMyAdmin (PHP and MySQL – provides full MySQL admin capabilities)
- LDAP Admin (LDAP support via OpenLDAP or Netscape SDK)
- PhpQLAdmin (supports qmail, LDAP)
- Mailing List Admin. (simple EZMLM interface, but it could be better using EZMLM and MySQL. PHP provides PHP classes; easy to use. Lacks most options for make/edit lists.)
- proBIND (kindly interface for BIND config)
- PhpCron (simple Web-based cron server in PHP, integrated with crond)

Some PHP resources are: *http://php.net*; *http://SourceForge.org*; and *http://aspn.ActiveState.com.*

Rules of Thumb of System Administration

Steve Simmons and Elizabeth Zwicky .

Summarized by Tim Smith

The presentation was a collection of the wit and wisdom of the system administration field. "The only thing more frightening than a programmer with a screwdriver or a hardware engineer with a program is a user with a pair of wire cutters and the root password" is representative of the slides used during the presentation. After each slide Simmons or Zwicky would tell a quick story related to the slide and would point out the underlying rule of thumb or great truth that could be used by system administrators in their jobs every day. The slides for this presentation are not available online. The material used by Simmons can be found in his sigfile collection at *http://www.nnaf.net/~scs/Fun/sigfiles.html.*

What Sysadmins Need to Know about the New Intellectual Property Laws

Lee Tien, EFF

Summarized by Josh Simon

The short answer to the title, according to the speaker, is "a lot." He provided a general overview of the issues, but when in doubt, always contact your own attorney.

The theme of the legislation of late has been to figure out who controls the technology. Copyright law provides the creator of a work or expression fixed in some tangible medium, including electronic media such as RAM and disk storage, the right to exclusively copy, sell, and distribute their work and the right to authorize others to do so. Copyright infringement is when someone does any of this without authorization. There are two kinds of infringement: direct and indirect. Direct infringements are those where you yourself are the violator. Indirect infrigements are when there is a direct infringement and you're involved intermediately. There are two types of indirect copyright infringements. The first is contributory, where you condone or help the direct violator, have knowledge (which has been extended to mean both "you know" and "you have reason to know"), and materially contribute to the violation, which includes the control of the facilities or the systems. The second type is vicarious, where direct infringement affects the right to control and leads to a direct financial benefit for the vicarious infringer. The example is of a tenant/landlord relationship. Since financial benefits are typically not present for system administrators, vicarious infringement probably doesn't appply to us. However, knowledge or reason-to-know do not apply to vicarious infringements.

So what can we as system administrators do? In smaller environments, we can avoid infringements. Unfortunately, this doesn't scale well. There's the so-called Betamax defense, which says if something can be used for substantial non-contributory use it's okay – but the courts aren't buying this argument yet, because it's only been applied successfully thus far to contributory, not vicarious, infringement.

What about Napster? They should have known there was infringement going on, and they provided the software and hardware (servers), so they've got contributory infringement. They also performed direct violations, and affected the right to control (vicarious) and cost the copyright owners revenue (vicarious). Even if only contributory infringement is involved, you can't foist it off and say it's someone else's problem once you have knowledge of it. So the advice here is to take cease-and-desist letters very seriously.

What about new legislation? Some case law shows that some knowledge is essential. Title II of the Digital Millenium Copyright Act (DMCA) provides safe harbors for ISPs and other providers, though the safe harbors are very complicated. A safe harbor provides immunity for monetary damages only and is intended to limit the legal exposure of the provider. There are four of them defined: transitory network passage, where all you do is deliver bits from one place to another, as in the Usenet model; system and caching, where you provide the hardware and OS but no monitoring; user-stored files, where you provide the disk space; and search-and-retrieval tools, such as Yahoo! The definitions and requirements and exceptions are all very complex, written in legalese, and there's very little case law behind them. In general, though, you have to meet the specific criteria for a safe harbor: you must have an anti-infringement policy, accommodate and not interfere with standard technical measures to protect copyrighted works, and comply with notice and takedown requests. Unfortunately, some of these terms, such as "standard technical measures" and "anti-infringement policies," are legally ambiguous.

The big question becomes who controls the technology of the Internet? The RIAA and others want to control it because it can be used to copy and distribute works to which they own the copyrights. The DMCA, in the opinion of the speaker, is a strategy to control devices, and it doesn't provide exceptions like the Betamax rule; so it requires the right to control access and to make devices to circumvent access controls.

Hardening Windows 2000

Phil Cox, SystemExperts Corp.

Summarized by Jason Wertz

For those of you out there who have to deal with Windows 2000 servers, there is always the question of how to protect your servers. What can you do as an administrator to make sure someone else's system is more attractive to a hacker than yours is? Phil Cox (*phil.cox@SystemExperts.com*) had many of the answers in his presentation, breaking the topic down into several parts. First, determine the purpose of the server. Second, what types of physical security are necessary for the server. Third, what should be done as the OS is installed to promote tight security? Next, what can be done to tighten the security on the server after it has been installed? Finally, he suggested ways to test the servers after they are locked down to make sure they are as secured as necessary.

Before a new Windows 2000 server is set up, its purposes must be established. Determine what services will be offered, which ones won't be offered, which computers the server is allowed to talk to, what domains and workgroups the server is part of, and what protocols the server will be using. If these answers aren't known, the server should not be set up.

After the server's purpose is known, one should decide how to secure it physically. At a minimum, case locks should be used, EEPROM passwords should be activated, and the hard drive should be designated as the first boot device if removable media is usable, and the server is publicly accessible. If the system is critical or highly sensitive, cages should be used as well. Other methods of physical security are up to the administrator.

Once the methods of physical security have been established, the installation can be done. When installing, use NTFS as the file system, set a good admin password, and install only the required network services. Do *not* upgrade from older windows servers if possible.

After installation is complete the administrator needs to figure out what services are actually running on the server. For each service to be kept, startup options need to be set. Unnecessary services should be disabled or deleted. Deletion is the preferred method since a deleted service can't be restarted by a hacker, though certain services are difficult or impossible to delete. System policies such as password policies, account lockouts, auditing policies, user rights, startup/shutdown policies, etc. should be set at this time. Directory permissions should also be checked. Networking must be looked at, and filtering methods should be used to protect various used and unused ports. Time synchronization should also be used. Next, the active directory must be secured. Finally, install service packs and hot fixes.

Once the system seems secure, it must be tested out for security holes. There are many commercial tools that can be used for this. Unless great care has been taken, these final tests will likely show that there are a few bases that still need to be covered. It is much better to find security holes at this stage rather than when a hacker breaks into the system and exposes them.

To download the white paper that is the basis for this presentation and contains all the details for each of these steps, go to: *http://www.systemsexperts.com/literature.html* and download "Hardening Windows 2000" (*tutors/HardenW2K101.pdf*). There should be a version 1.2 of this file soon.

SANs and NAS

W. Curtis Preston, Storage Designs

Summarized by Mark Logan

Preston focused on the differing technologies of SANs (Storage Area Networks) and NAS (Network Attached Storage), and the strengths and weaknesses of each. He concluded with a look at the future of the two technologies and how they will be affected by the advent of NFS v4 and NDMP (Network Data Management Protocol).

The first segment of the presentation discussed the actual architectures of SANs and NAS. SANs is a fiber-channel network of RAID devices attached to a host machine. NAS, on the other hand, is an appliance (often called a "filer," a term coined by Network Appliance) attached to a LAN. NAS typically uses either NFS or CIFS as the network file system. Preston mentioned that running SANs behind NAS is becoming more and more common.

Preston attributed several advantages to SANs, such as reliability due to the ability to design systems with no single points of failure between storage arrays, but he was also fairly critical of the technology, pointing out its very high cost and its difficulty to administer.

Throughout the talk, Preston was much more enthusiastic about NAS technology. He was the first to point out that last year, he categorically warned against trying to run an RDBMS on a NAS appliance. Now, he is a proponent of the practice, after seeing numerous success stories. The advantages he attributed to NAS included ease of administration, its generally superior speed measured against equally priced SAN and local disk solutions, and many of the "goodies" being included by NAS manufacturers, such as snapshots and truly advanced file systems.

He was quick to mention that a NAS system does suffer all the flaws of the network file system it implements, and that NAS presents some problems in the realm of backups. However, NDMP promises to fix many of these problems by allowing backup from filer to self, filer to filer, filer to backup server, and server to filer.

In conclusion, Preston declared that the choice between SANs and NAS was largely dependent on the particular situation. The overall message of the talk, however, seemed to be that NAS was really maturing and becoming more affordable, but SANs still offered more in the realm of high availability and performance.

NETWORK/SECURITY TRACK

Whither End-to-End: Placing Bandwidth and Trust at the Edge

Gordon Cook, The Cook Report

Summarized by Jin-ping Wan

Gordon Cook, the author of *The COOK Report on Internet*, a monthly newsletter on Internet infrastructure development, calls for customer-empowered network infrastructures, in which customers control bandwidth and other resources over telco-empowered network infrastructures. Today's Internet is dominated by a few supercarriers; in the interests of the public, this should change to a scenario dominated by customers' networks intersecting global and other local networks. This is analogous to computing, which was dominated by large mainframe computers 40 years ago, and changed to personal computing due to the proliferation of mini-computers in the 1970s followed by the PC. Gordon uses Canada's CA*net4 to illustrate an edge-controlled infrastructure that has customer-owned networks with fiber bandwidth to the users.

Crypto Blunders

Steve Burnett, RSA

Summarized by Mike Sconzo

Cryptology can be a powerful and secure way to send data, but it must be used properly. Algorithms can range from simple to complex, from almost unbreakable to easily broken. Whatever the algorithm, the use is the same: encrypt data and keep it safe from attackers. Five blunders were introduced in this presentation.

One of the newest blunders is to declare one's algorithm unbreakable. Several crypto schemes have done this and as a result were quickly broken having garnered the attention of people wanting to be "the one who broke the unbreakable." The newest trend is using a security proof to prove mathematically that your algorithm is "perfect." One such case was the Atjai-Dwork crypto system; the algorithm was "proved" to be unbreakable in 1997 and broken in 1998.

The second pitfall is "worshiping at the altar of the one-time pad." Since the one-time pad crypto system was proved to be perfectly secure by Shannon, several companies and individuals have tried to use this to their advantage. The problem resulting from adopting the one-time pad to suit specific needs was demonstrated by Microsoft in 1998. Microsoft created a product that used the one-time pad as part of an algorithm, but the pad was used twice. This allowed people to figure out what the pad was and attack the traffic.

Another problem arises from not using the best available algorithm, which leads to the commonsense question, why should an algorithm be used when it is known to be insecure? This also leads into the next blunder: an incorrect implementation of an algorithm. This was illustrated by a story about a man who called with a complaint about the RSA he had implemented. No matter what he did, his message always encrypted to itself. When asked what he was using as his exponent he replied "1". In the formula $c = m^x \bmod n$, if 1 is chosen for x, then m will always equal c. Lesson learned.

Finally, blunders can stem from an incorrect implementation or just poor security policy. This happens when you "don't protect the key." If the private key is not protected, any crypto scheme becomes near trivial to break. Some famous instances of not protecting the key arose both from Microsoft and Netscape.

It is important to keep an eye on the crypto system that you are currently using. Make sure that it is not out of date, you have a good implementation, the private key is kept private, the algorithm is used correctly, and it is a good system to use. Finally, even if all those are present, the issue becomes one of trust. With third-party systems in place to verify identities, who has to worry about invalid certificates? Then again, maybe we *should* worry.

How Not to Configure Your Firewall

Avishai Wool, Lumeta Corp.

Summarized by Joel Sadler

Wool presented a fast-moving talk on firewall configs, mostly centered around Checkpoint Firewall-1. He opened with some overall policy auditing concepts before moving to common misconfigurations. Unsurprisingly, the most common errors that he's seen are allowance of all DNS traffic (both inbound and outbound), improperly controlled ICMP, and general problems with misuse of the "ALL" directive. Using example client data, he showed specific firewall configurations with serious problems. His strongest recommendation to firewall administrators was to keep their policies as simple as possible. His data showed that as firewalls grew in complexity, not only were new vulnerabilities introduced, but the danger of exploiting existing vulnerabilities increased.

The Future of Computer Security

Moderator: Marcus Ranum, Network Flight Recorder; Panelists: Tom Limoncelli, Paul Proctor, Anne Benninger, John Flowers, and Steve Atcheson

Summarized by Mike Sconzo

The question posed to each member of the panel was "where will we be in 3, 5, and 10 years?" One believed that computer security will get much worse before it gets any better. Although quite a few people think that the majority of the problems will be partially if not completely solved within 10 years. This is because we should have better tools and more knowledge about the problem(s) we are trying to solve.

It was pointed out that we are headed in the right direction, and with the increased realization by management that security is important, there will be more spending on security-related technology. The industry can then produce such true security products as a (nearly) self-correcting operating system and products that enhance security rather then just acting as burglar alarms. These ideas about security and security enhancements will eventually trickle down to IT people, and this will eventually help with the current state of security.

No matter how great the technology is, however, we will still have problems due to human error. This can be mitigated through education. Consolidation of products/ideas is also anticipated so that products will be easily scalable.

The need for a standardized system of evaluation was also brought to light. Some panelists agreed that the business sector (e.g., insurance companies) would influence this. It might eventually become possible to buy network security insurance. The insurance would be priced according to how secure the network was, and this would lead to a security rating system. Introduction of VISA compliance standards might be another way to achieve this.

Several other issues were discussed, such as Public Key Infrastructure, authentication, and risk measurement. Everybody seemed to agree that "management is bad, insurance and government standards are good, and PKI is dead."

GURU SESSIONS

AFS

Esther Filderman, PSC, and Garry Zacheiss, MIT

Summarized by Mark Logan

The AFS session consisted of questions about bugs in various AFS implementations, questions about AFS setup and administration, and commentary about the state of the AFS community. The discussion took place in a packed room containing everybody from longtime veterans of AFS to people who were just curious about AFS.

The first round of questions addressed bugs in certain setups that seemed to be caused by Jumbograms, which are on by default in AFS. Jumbograms can speed up AFS communication in some cases, but the consensus was that they should be the first thing to go anytime strange bugs start to show up.

The discussion then turned to issues of AFS performance, including caching to disk and/or memory, and how AFS performance compares to that of NFS. One attendee told of having dramatically improved workstation performance with a relatively small memory cache, while another shared his experience of running AFS with a 2GB memory cache on an E10K. Allegedly, it was rather sprightly.

Discussion of backup was bound to come up sooner or later, and it centered around alternatives to butc, the most common backup tool used with AFS. However, the only tool about which anybody had anything good to say was Tivoli Storage manager. A few folks who were unconcerned with preserving AFS ACLs reported success using tar and commercial backup products.

Toward the end of the session, the history and future of AFS came up, and Esther and Garry fielded questions about the Transarc implementation, the progress of OpenAFS (which is now reportedly quite stable), and the directions that AFS may take in the future. There was some speculation about the possibility of an AFS Foundation, but the gurus were not optimistic about such an organization actually being founded in the near future.

INFRASTRUCTURE ARCHITECTURE

Steve Traugott, TerraLuna, LLC

Summarized by Tim Smith

Topics suggested for discussion were large systems beyond credible manual reach; industry acceptance and understanding of infrastructure architecture; notations, semantics, and type checking in infrastructure architecture; organizational infrastructure; and turn-key management solutions. However, organizational infrastructure and notations, semantics, and type checking were not discussed before the sessions ended. Traugott began the discussion on large systems by saying that without automation and tools there is an infrastructure such that an infinite number of system administrators could not administrate it. The key to managing such large infrastructures is centralized management of the systems and network. The discussion then shifted to tools used to centrally managed network hardware. On the software and configuration side, how to avoid changing the infrastructure manually was discussed. Manual changes should never be made even though the pressure to immediately fix a problem is great. Instead, the changes should be made using the tools available. If this is done correctly, a machine can be reformatted and all of the customization after the base operating system installation can be recovered automatically.Industry acceptance was the next topic covered. The primary problem faced in this area is describing to recruiters and bosses what an infrastructure architect does. An infrastructure architect is concerned with reducing the cost of ownership of managed systems through careful planning of the infrastructure of the computing environment. They are typically a senior system administrator who can code well and whose intent is to build the view of a single enterprise architecture. Infrastructure architecture can be thought of as something to do after system administration. An infrastructure architect is needed to make the design decisions because if these decisions are not made by someone assigned specifically to the task then they are made during triage situations in emergencies, and this is clearly evident in the resulting infrastructure.Turn-key management solutions for infrastructure architecture do not exist at the moment. The best solutions would be written by vendors, but those would not be acceptable in heterogeneous environments. Any solution produced for a site tends to be site specific and not really sharable. The Arusha Project (*http://ark.sf.net*) was mentioned as an effort to make site-specific efforts sharable between sites using their XML-based configuration language. Standardizing GNU tools and infrastructure policies are other ways to allow sharing.More information can be found on *http://infrastructures.org*.

PKI/CRYPTOGRAPHY

Greg Rose, Qualcomm, Inc.

Summarized by Mark Logan

The PKI guru session started off with the announcement that the AES has been approved. Rijndael, the winning cipher submission, was accepted with few modifications. Rose expressed his satisfaction with the committee's choice and cited Rijndael's propensity for encrypting very quickly as its biggest strength. He added that there was hardly any doubt that all of the ciphers under consideration were secure, so factors such as speed were given greater weight. Discussion focused for a while on problems with current PKI implementations. Top on the list of gripes was the difficulty in issuing certificate revocations. Rose and a few attendees explained several different approaches to addressing this problem. The simplest one was to simply set certificates to expire relatively quickly. One of the more interesting proposals involved storing part of a key on a trusted server, so that to sign or encrypt documents, users would have to pass part of the computation to the server since they would hold only a portion of the key. Then, instead of revoking certificates, an administrator would simply remove the key material from the trusted server, and the relevant user would not be able to sign or encrypt documents. In the second half of the session, several attendees had questions regarding the state of export regulations. There was some trepidation in the room about what the US legislature could be expected to do with regards to cryptography export regulations in the wake of September 11th. Rose felt that regulations were still quite relaxed compared to those of several years ago and that there was little cause to worry for the time being. He justified his stance by arguing that export regulations were never meant to keep cryptography from leaking across US borders but, rather, were meant to keep any non-US business from using US cryptographic technology to do business.

Writing Papers for USENIX Refereed Track

Tom Limoncelli, Lumeta

Summarized by Josh Simon

Tom noted that publishing a paper was good since it helps the community and can change your career (allowing for both peer and management recognition, and providing ammunition when your boss needs to justify your next raise).

How does one start writing a paper? The advice here is to write what you know. Are you doing anything to make your life easier? Automating a task? Writing a cool tool? Working on a neat project? Providing a case study, whether positive ("Here's what we did, and it worked") or negative ("Here's what we did, how it broke, what we did to fix it, and what we should've done to begin with")? Asking yourself, "What have I done that nobody else has" is an excellent way to start. Then following up with the terms and concepts and a statement of the problem, its scope, and how you solved it provides a good basis for your paper.

Don't forget to survey the literature. With the publication of *Selected Papers in Network and System Administration*, or "The Best of LISA" as it's been called, there's a single place to go to find references. Add to that the resources available to all USENIX members on the *http://www.usenix.org/* Web site and you're definitely off to a good start.

Tom also discussed the evaluation process, based on his experience serving on or alongside several program committees. The readers consider whether the paper is enduring and whether it can result in a good presentation. Papers are evaluated on several criteria, including the technical quality of the work, the presentation of the paper, whether it advances the state of the art of system administration, and whether it's relevant for LISA or somewhere else.

If your paper is not accepted, don't consider that anything more than a momentary setback. Papers are usually returned with commentary that explains why it was not accepted and suggestions on where to submit it (if not to LISA next year), along with commentary on the paper's quality, presentation, and so on.

If your paper is accepted, meet your deadlines. Work with your shepherd, whose job it is not only to nag you to make deadlines but also to help you by providing constructive feedback on what is good and what isn't. The shepherd is a resource to help make your paper the best it can be. Remember that they have their own lives to live but that they are willing to help you out – just don't deliver a draft and expect same-day turnaround.

Some additional commentary:

- Both proofread and spell check your paper. Have someone else proofread and spell check your paper. You're too close to it by the final submission deadline, so another set of eyes can help a lot.
- Do your presentation beforehand. Practice in front of a mirror, or present it to your team, department, or company, or to your SAGE local group.
- Give away the ending early. You're not writing a mystery novel; it's a refereed paper. You should identify the problem you're trying to solve and how you solved it in the abstract, the introduction, or both. You should also spell that out early in the presentation.
- In your presentation, consider demonstrating the tool (if your paper is about a cool new tool). Also, consider what your audio/visual needs will be: laptops, transparency projector, microphones, any special needs. The AV team needs as much advance warning as possible.

Finally, we discussed some paper ideas and how best to present them for future conference paper tracks.

WORKSHOP SERIES

CFENGINE

Moderator: Mark Burgess

Summarized by Mark Burgess

The cfengine workshop, led by Mark Burgess, attracted 21 participants from all backgrounds to discuss the current and future developments in cfengine v2. Among the highlights: a discussion, prompted by Steve Traugott, as to whether it is best to determine a configuration as a sequence of known steps (Steve's view) or as the specification of a final state (Mark's view) – the two are not always equivalent; Paul Anderson talked about the need for even higher-level specification languages at the enterprise level; Andy Mayhew and Christian Pearce discussed how cfengine could be enhanced for the enterprise with ancillary tools like CVS and LDAP; and Martin Andrews talked about using cfengine on Windows NT/2k. The workshop produced a stimulating discussion, which is summarized at *http://www.cfengine.org*.

METALISA

Tom Limoncelli, Lumeta, and Cat Okita, Earthworks

Summarized by Josh Simon

The MetaLISA workshop about managing system administrators first discussed the question of how to provide motivation to help retain quality personnel. We decided that providing a good work environment without major stressors would be better than just throwing money (salary) at the problem, that authority and responsibility should both be well defined, and that resources have to be made available to handle problems.

Next we discussed the different types system administrators: the "work 9 to 5, get a check, leave work at work" type and the "computers are my life so I play on them at home, too" variety. One manager organized his group so the former type was given the trouble-ticket queue processing and the latter was given more of the infrastructure and hard install problems.

We then discussed career path issues. Several companies now have multiple paths and levels, such as team lead, project lead, and assistant manager, each with appropriate and well-defined levels of expectations, evaluation scores or results, experience, requirements, and so forth. Providing different levels of responsibilities, independence, authority, and even money (base pay increase) on a path for both technical and management types, junior to senior, seems to work well. Even better is when there are well-defined criteria for promotion and lateral transfers between tracks. Remember, however, to provide allowances for exceptions or case-by-case waivers in your written policies.

Next we covered professionalism. Some people lie on their resumes. Some people wear inappropriate clothing (suit or t-shirt) to an interview or to the job itself, and don't alter their clothing choices even when informed to do so. Some people don't understand the concept of punctuality. Some people don't know how to be tactful, to provide the right level of information, or even to say "I don't know" to the customer. One topic of discussion was how to educate these people to improve these skills. Information sharing – such as email lists, databases, and even IRC channels – helps teams to share knowledge, cross-train people, and provide a way to let everyone contribute. Admitting when you're wrong builds trust for when you're positive that you're right. Encouraging people to ask for help can work, but so can offering help and asking people if they need help. However, the insecure may not respond or take you up on the offer. For these people, it may help to present a situation as a "show me what you did so I can learn." Don't use killing statements (such as "you're wrong") but ask leading questions ("what if").

Next we looked at determining what information is important (to share) and what is not (to keep political fallout from the team)? One technique is to have a staff meeting and say, "Here's the important stuff" and then let folks leave if they don't care about the politics. It's necessary to get folks to realize "best" isn't always "right" and that politics can override the right technical basis. The team needs to be aware that there are politics even if they don't know the details. However, email is often not the best medium for this; in-person and telephone contact may be a better (or, at least, a good supplemental) way to contact and inform people. Also, giving people the framework to put the details in and answering their questions is good. Some people, though, just don't care about the political issues. Sometimes, having face-time in meetings with your people and the lord high political muckety-muck may be useful.

Many people can follow a checklist and don't have problem-solving skills. How do you teach them to acquire the skills? Problem-solving is linked to curiosity, background, and experience. Teaching people skills is important. Using child-raising techniques, such as brainstorming with a timer, may be helpful. Again, you have to be careful to ask leading questions and not use killing statements that make the other person defensive. People need to remember to look at the big picture in order to make informed big-picture decisions so questions of direction get addressed within the group. Also, the problem and scope need to be explicitly defined, because that sets some limits. Finally, the instruction or detail level of the recipient may be relevant; instructions to senior people may be much less detailed than instructions to juniors.

The next major discussion topic was the balancing act between technical and management responsibilities and tasks. Some of the tricks include allowing the people who report to you to assume you're still technical, knowing that the theory can be as good as the technical details, keeping yourself informed about major issues, and using one day a week as a technical day for working on small projects. Also, if you only rise to the point of comfort, whether that's team lead, division lead, project manager, company head, or whatever, you may be better able to find your balancing point. One of the problems is trusting the people to whom you hand off your pet projects to will do the "right thing" with them.

Finally, we discussed moving from an ad-hoc group to a more procedurally based group, formalizing processes by documenting not only how things work, but also why the decisions were made. Pairing people, one to explain and one to write, can work well. Starting with something like script is a good start. Having a cheat-sheet or template can be very helpful. But you have to practice what you preach; document things yourself so your people will. Also, make documenting a requirement for the performance review.

Sysadmin Book of Knowledge Project

Moderators: Geoff Halprin and Rob Kolstad

Summarized by Rob Kolstad

Twelve people spent the day planning out the next phase of the System Administration Book of Knowledge (BoK) project that Halprin started a few years ago.

Halprin and Kolstad opened the workshop with over three hours of presentations that we have independently been delivering to audiences over the past year. The presentations introduced and motivate the Sysadmin BoK.

The Sysadmin BoK is intended to name all of the items that a system administrator might encounter in his or her work. It goes little further than naming the items – it is not a tutorial or "best practices" document. In its barest form, the BoK will end up being a list of 2,000 or so line items (e.g., "backups," "security policy for firewall," etc.).

When annotated with a paragraph or two for each of the line items, the BoK becomes:

- A weighty tome to impress those who wonder what sysadmins do for a living
- The basis for curricula – university/HS, training, advancement, individual career planning (the individual point of view)
- The basis for creating a benchmark for corporate IT/admin maturity (corporate/organizational point of view)
- The basis for creating best-practice documents (refining the knowledge)

In combination with the above, the BoK forms the foundation of system administration as a real profession. It is only the beginning, but as a strong foundation, we believe it is essential.

Currently, the BoK has about 1500 line items created by a core team of a dozen sysadmins and occasionally reviewed by a total of just over 100. The line items have been categorized into a 73 x 44 matrix whose rows are general topics of sysadmin and whose columns are specific properties of those topics (e.g., "security" and "mobility"). Printing the list in 9-point type, two columns, small margins yields 19 pages of line items.

Discussion over the remainder of the day yielded these action items:

- Combine some of the rows and columns to reduce the number of elements in the matrix.
- Add new rows and columns, as contributed over the last few months by reviewers.
- Fill in another 500 or so line items.
- "Factor out" common elements that appear throughout a column (e.g., common security elements).
- Write a paragraph or two for each line item.
- Find a way to present the matrix in linear form (i.e., in a book).

Then we can proceed with the subsequent BoK projects:

- Capability maturity model
- Encyclopedia

The project particularly needs reviewers with a strong background in administering sites with Windows, both on servers and desktops. If you would like to contribute, please contact *kolstad@delos.com*.

The project Web site is *http://ace.delos.com/taxongate*; trivial registration is required so that your contributions can be tracked.

AFS

Derrick "Dana" Brashear, CMU; Ted McCabe, MIT; OpenAFS Elders; and Esther Filderman, PSC.

Summarized by Garry Zecheiss

Twenty people interested in AFS, OpenAFS, and Arla participated in the AFS Workshop, either by giving a short presentation or by suggesting topics and contributing to their discussion.

Derrick Brashear gave an update on the status of OpenAFS. New ports are available, including MacOS X and Solaris 9. There are many new features, including some support for AFSDB resource records, dynroot support, and a new build system using autoconf, as well as some bug fixes, including better RX tuning. Long-awaited features, such as disconnected operations and Kerberos 5 support, are coming soon.

Love Hornquist-Astrand gave an update on Arla. New features include more support for MacOS X and FreeBSD and support for incremental file caching. A lot of RX work is being done, including GSSAPI/SPNEGO support and the removal of unwanted features. Love and Magnus Ahltorp have been working half-time on Arla but the funding for this stopped at the end of 2001.

General discussion covered topics such as Arla-AFS compatibility; cache manager issues, including using memcache; backups – what are people using and how do they cope with the Transarc backup system; migration to Kerberos; AFS infrastructure tools; proactive AFS administration; performance tuning; and getting funding support and publicity for OpenAFS.

Details about the workshop are available from the AFS workshop Web site at *http://www.psc.edu/~ecf/afs-workshop/*.

Teaching System Administration

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

Summarized by Tim Smith

The third annual Teaching System Administration workshop consisted of four sections. The first section focused on learning objectives for a system administration course. These objectives ranged from basic system administration tasks such as creating and managing user accounts to more advanced issues such as needs assessment.

The second morning session focused on assignments for a system administration course. The discussion included how quizzes, tests, labs, and projects could be structured. During the discussion the participants who had already taught a system administration course related what they had done in their course and how it worked. The participants broke up into groups to design a lab for one of the learning objectives discussed in the first session. Presentations of the labs concluded the morning session. During the lunch break participants discussed how to evaluate success in a system administration course.

The first afternoon session began with a group summary of the different discussions held during the lunch break. This discussion lead into the topic of exam questions. Different types of exams, from essay to multiple-answer and true/false, were considered based on the ability to evaluate what a student has really learned and how easy it is to grade. Based on this discussion the participants broke up into their small groups from the morning session to come up with three exam questions. Most of the questions produced by the different groups were short answer, with a few multiple-answer questions and essays.

The final session of the workshop centered on large and real-life projects. The discussion was about how to bring real projects into the class without inconveniencing a business owner while still allowing students to apply what they were learning in class. The discussion shifted to the types of projects that had been assigned by the participants who had taught a system administration course or completed projects in a similar networking course.

The workshop concluded with a discussion of tools, primarily ones that would make grading easier and allow materials to be shared between individuals teaching system administration courses. Ongoing discussions about this issue continue on the sysadmin education mailing list at *sysadm-education@peak.org*.

Advanced Topics

Moderator: Adam Moskowitz

Summarized by Josh Simon

The Advanced Topics Workshop was ably hosted and moderated once again by Adam Moskowitz and co-piloted by Rob Kolstad. Introductions by the 26 attendees generated interesting questions and topics for discussions: random opinions, the Undo command for sysadmins, hot tools, and surprises from the past year.

Random Opinions

People are indeed using SANs and NAS, since they're well suited to specific problems (such as archiving, Fortune 1000 companies, and so on). However, they are not being used for general file services, mainly because the FibreChannel implementation is too expensive for general use.

We also discussed the centralization/decentralization pendulum, which seems to be moving back toward centralization. Perhaps condensing is a better term, since places are condensing locations for their hardware and personnel but still keeping some geographic distance between them. Centralizing administrative functions is different from actual physical centralization, since (to use the SAN/NAS model), users don't care if the disk is local or across the continent as long as the performance is unaffected.

We're moving toward more of an ASP model within a given environment, be it company or infrastructure. The ASP model works well between divisions within an organization but not as well between different organizations, primarily due to trust issues.

The events of September 11th caused a shift in the thinking of some of the tight-fisted financial staff. They now realize how integral computing is to business, so collocation and backups are now more important.

The next major topic was mobility. Without mobility today's commonplace high-speed network infrastructures and reliable file servers make a lot of system administration fairly easy; workstations can be built from images or automated installation processes, and all mutable data lives on centralized file servers, where it's easy to backup and manage.

But mobility changes all that. Mutable data has to be local to the endpoint (e.g., laptop); we can't expect network connectivity to be high-speed, and we have to be able to deal with connections over insecure networks. We have to deal with a host of security issues, find new ways of ensuring data availability, and be able to provide the needed services of various levels of network quality.

Mobility is becoming increasingly important – there are now many organizations where most endpoints are mobile platforms. But IT infrastructures have not yet caught up to this changing reality. To deal with this we will have to abandon our traditional (and previously successful) modes of thinking and use technologies that involve disconnected operation, mobile IP, synchronization, transparent data encryption, and so on.

Wireless computing has changed our behavior; 70% of us in the ATW are on laptops. Our expectations seem to be that we're approaching ubiquitous computing; of those using laptops, about 2/3 use them to access remote services (mail, Web, files) and 1/3 use them as the centralized storage point. This leads to the intrusion of mini-environments into your own macro-environment. Laptops can move from administrative domain to administrative domain and pick up and distribute viruses and whatnot in the process. Managing and keeping them from screwing up your environment is a hard problem.

Recovery-Oriented Computing

Recovery-Oriented Computing (ROC) is targeted to services. A PowerPoint presentation is available, along with information at *http://www.cs.berkeley.edu/~pattrsn/*.

The goals are ACME – availability, change, maintainability, and evolutionary growth – instead of performance (which is what we've looked for in the past 15 years). We're not doing that well.

One of our needs is not just to get real data to improve reliability but to measure reliability and availability. Making the system administration tasks have an Undo function may be helpful. Think about the three Rs: rewind (go back in time), repair (fix error), and redo (move forward again). We're looking to recover at the service level, not just at the server (hardware or component) level.

- Predictability – Having predictable recovery would be a huge improvement even now. Most recovery plans (or even risk mitigation) is pure guesswork now, based on experience and trial and error. Change control and change management need to be more formal and actually predictive of detailed determination.
- Avoidability – Can you avoid the problem to reduce the recovery time? If you can avoid the problem then the need for recovery time is less. This is reasonably important and very hard.
- Repeatability – Making tasks easily repeatable will help reduce complexity and can lead to increased avoidability and thus increased reliability.
- Risk mitigation – A lot of the changes we make at one time – one change – affect multiple machines (such as servers, routers, switches, firewalls, and so on). Rollback within any one system is good, but we need to have rollback in all of them. The problem becomes system-specific; is it a GUI or CLI?
- Tools – They're trying to reduce the MTTR in the MTTR/MTTF equation. This project is more about building recovery-from-something-that-has-happened than making-the-problem-less-likely-to-occur.

Right now the thought is to build a sample (prototype) email system as a starting point.

What about security breaches (intrusion detection)? Something similar can be done; this kind of technology would be good. You could roll back to before the intrusion, install the filter or preventative mechanism, and then roll the good stuff back in again.

Simply changing (fixing, simplifying, etc.) the interface is insufficient. Work does need to be done on SA recovery interfaces but this is beyond the scope of the ROC project.

Hot Tools in Use Today or Coming Soon

Next we discussed the new tools, technologies, ideas, or paradigm we're investigating or using. The list included new IP telephony products; tricks for SSH and CVS; wireless networking; integration and aggregation of alarm, monitoring, and administrative functionality with automation; reducing information replication; load balancing; anomaly detection; miniaturization; mirroring network storage for high-speed failover; VMware; MacOS X; Java; and Perl 6. The list also included business problems as opposed to technology problems.

There was a side discussion about programming languages. Some people like Java, others like C#. Java is the new COBOL in that it's the new business language but not a system language. Some debate ensued, with no conclusion, about whether to teach C, C++, Java, or even Scheme first.

Surprises from the Past Year

Several people mentioned surprises they'd had in the past year. This list includes Cygwin, the PC Weasel, the dearth of middlemen in the DSL/POP/ISP markets, and the number of people running wireless networks without any security.