Upcoming Events

Special Workshop on Intelligence at the Network Edge
Sponsored by USENIX, with support from 3Com
MARCH 20, 2000
SAN FRANCISCO, CALIFORNIA, USA
http://www.usenix.org/events/ine2000

SANS 2000—9th International Conference on System Administration, Networking, and Security
Co-sponsored by the SANS Institute and Sage
MARCH 21-28, 2000
ORLANDO, FLORIDA, USA
http://www.sans.org

SANE 2000—2nd International System Administration and Networking Conference
Organized by NLUUG, co-sponsored by USENIX and Stichting NLnet
MAY 22-25, 2000
MAASTRICHT, THE NETHERLANDS
http://www.nluug.nl/events/sane2000/

2000 USENIX Annual Technical Conference
JUNE 18-23, 2000
SAN DIEGO MARRIOTT HOTEL & MARINA, SAN DIEGO, CALIFORNIA, USA
http://www.usenix.org/events/usenix2000

3rd Large Installation System Administration of Windows NT/2000 Conference (LISA-NT 2000)
JULY 30 - AUGUST 2, 2000
MADISON RENAISSANCE HOTEL, SEATTLE, WASHINGTON, USA
http://www.usenix.org/events/lisa-nt2000
Submissions due: February 16, 2000

4th USENIX Windows Systems Symposium
AUGUST 3-4, 2000
MADISON RENAISSANCE HOTEL, SEATTLE, WASHINGTON, USA

9th USENIX Security Symposium
AUGUST 14-17, 2000
DENVER MARRIOTT CITY CENTER, DENVER, COLORADO, USA
http://www.usenix.org/events/sec2000
Submissions due: February 10, 2000

4th Annual Linux Showcase and Conference
Sponsored by USENIX and Atlanta Linux Showcase, in cooperation with Linux International
OCTOBER 10-14, 2000
ATLANTA, GEORGIA, USA
http://www.linuxshowcase.org

4th Symposium on Operating System Design & Implementation (OSDI 2000)
Co-sponsored by IEEE TCOS and ACM SIGOPS
OCTOBER 23-25, 2000
PARADISE POINT RESORT, SAN DIEGO, CALIFORNIA, USA
http://www.usenix.org/events/osdi2000
Submissions due: April 25, 2000

14th Systems Administration Conference (LISA 2000)
Sponsored by USENIX & Sage
DECEMBER 3-8, 2000
NEW ORLEANS, LOUISIANA, USA
http://www.usenix.org/events/lisa2000
Submissions due: June 6, 2000

6th USENIX Conference on Object-Oriented Technologies and Systems
JANUARY 29 - FEBRUARY 2, 2001
SAN ANTONIO, TEXAS, USA
http://www.usenix.org/events/coots01
Submissions due: July 27, 2000

For a complete list of future USENIX events, access http://www.usenix.org/events
IN THIS ISSUE . . .

LETTERS TO THE EDITOR

CONFERENCE REPORTS

LISA '99

DSL '99

USITS '99

SECURITY

39 The Network Police Blotter by Marcus J. Ranum

42 Building a Windows NT Bastion Host by Stefan Norberg

OPEN SOURCE

51 Source Code UNIX by Bob Gray

SYS ADMIN MANAGEMENT

55 Politeness in Computing by John Nicholson

59 Interviewing for Sysadmins by Dave Clark

64 Chunks by Steve Johnson and Dusty White

65 Musings by Rik Farrow

PROGRAMMING

68 Using Java by Prithvi Rao

72 Effective Perl Programming by Joseph N. Hall

78 Java Performance by Glen McCluskey

BOOK REVIEWS

81 The Bookworm by Peter H. Salus

82 Applying UML and Patterns reviewed by Clyl Flpyt

STANDARDS REPORTS

83 A New Editor by Nick Stoughton

83 Introductions by David Blackwood

84 POSIX Revision Project by Andrew Josey

USENIX NEWS

85 The Times, They Are A'Changin' by Andrew Hume

86 New Projects Funded by Gale Berkowitz

86 Board Meeting Summary by Gale Berkowitz and Ellie Young

87 20 Years Ago in USENIX by Peter H. Salus

88 Report of the Nominating Committee

SAGE NEWS

91 Ready! Set. No? by Tina Darmohray

92 From the SAGE President by Barbara Dijker

93 SAGE Update by Gale Berkowitz

ANNOUNCEMENTS AND CALLS

4 14th Systems Administration Conference (LISA 2000)

6 mtd by Rob Kolstad
In this issue

by Jane-Ellen Long
Managing Editor
<jel@usenix.org>

In and out, changes for the new year. In acknowledgment of SAGE's valued and weighty presence in the USENIX community in general and in ;login: in particular, the mag you hold in your hands bears the official subtitle of The Magazine of USENIX and SAGE. Tina Darmohray, who's done a heroic job as SAGE editor for years, with this issue begins her even more taxing role as co-editor of ;login:. Rob Kolstad, a man not only brilliant, witty, and charming, but apparently possessing limitless energy as well, has agreed to continue as co-editor. He will also act as chair of ;login:’s Editorial Board, now in the process of formation. And Rik Farrow, in addition to his popular column "Musings," will continue to serve as editor for special “theme” issues.

The SAGE features section has been incorporated into the main body of the magazine. As you will see on the redesigned Contents page, feature articles are sorted by topic. Tabs at the right-hand page margins lead you through the sections; the running feet show the article title. SAGE News remains a separate section, with its familiar colored background to help you locate it easily. Please let us know what you think of our new look; email login@usenix.org.

More news: on the Web, ;login: issues and conference proceedings over a year old are now free to all. Membership is still required for print subscriptions and for access to the more recent issues and sets of proceedings.

With this issue, we welcome a new Standards Report editor, David Blackwood. Nick Stoughton continues to serve as the USENIX Institutional Representative - see p. XX for details. Marcus Ranum begins a series on security-related issues, and John Nicholson begins one on legal issues.

Now, go on in. The Y2K fuseless bomb’s safely out of the way. Enjoy the issue, and your next millenium.

Cover Photo: LISA ’99 Reception at the Museum of Flight: Stewardesses modeling the uniforms they once wore.
DEVELOPMENT OF NETBSD
from David Maxwell
david@vx.net/david@maxwell.net

Dear Rik,

You followed the 386BSD->FreeBSD history, but not the NetBSD (<ftp://ftp.netbsd.org/pub/NetBSD/misc/release/NetBSD/NetBSD-0.8>)one:<http://www.netbsd.org/Misc/history.html>. NetBSD 0.8, also a derivative of 386BSD, was announced/released April 19, 1993.

The NetBSD and FreeBSD developers (to be) had discussed things at length before that April, and in essence, agreed to disagree. The FreeBSD folk wanted to focus on the 386 platform and put their full energy into it; the NetBSD folk wanted to continue in the UNIX heritage of portability, and take it further.

In 1994–95, Theo de Raadt, who had contributed to NetBSD, had an argument with some of the NetBSD core team. Not all the details are public. I became aware of this when Theo posted a message to one of the NetBSD lists claiming that his messages were being censored, and so he had put up a Web page to publish his complaints. I followed the link to the page, and I was convinced quickly (in large part by Theo's tone) that Theo was in the right.

Theo took the current release, NetBSD 1.1, and relabeled it OpenBSD. Theo has since then worked to raise the profile of OpenBSD, focusing on security. (Part of his argument with the NetBSD core team was over the lack of speed with which his changes were integrated into the code.)

Theo calls NetBSD an "academic research platform" — that is, not worth considering for production systems. My usual response is to ask, "If NetBSD is just an academic research platform, why was OpenBSD completely based on it?"

I mention this because in your article you said you were told by your anonymous informant that "The NetBSD group ... is more interested in experimentation than in having a rock-stable version of BSD." That sounds like a small twist on the "academic research" line that OpenBSD tries to sell.

So here's the sales pitch: Linux and FreeBSD claim to support multiple platforms. I looked today at <http://www.freebsd.org> and <http://www.linux.org>, and I couldn't find any mention of a non-Intel installation. I know that Red Hat and some of the other Linux distributions do provide support for other platforms; I'm just indicating that it's still not an important thing for them.

OpenBSD has support for Alpha; Amiga; HP300; I386; Mac 68k; MVME 88k; PowerPC; PMax; and Sparc.

NetBSD has support for Alpha; Amiga; Arm32; Atari; Bebox; HP300; HPC-MIPS I386; Mac 68k; Mac PowerPC; MVME 68k; NEWS-MIPS; Next 68k; OFPPC; PC-532; PMax' SH3; Sparc; Sparc 64; Sun 3; VAX; and x68k.

NetBSD has not been (and probably won't be in the future) about being popular, it has been about doing it right. The same source tree compiles on all the above platforms. Yes, there are some platform-specific driver files, but they are kept separate in the system.

Linux and FreeBSD may claim they support multiple platforms, but the code is not integrated into their source tree. As changes are made to Intel Linux, the Alpha Linux and Sparc Linux and PowerPC Linux (....) development teams must rework that code into their out-of-date copy of the source tree. This may change at some date, but only if it becomes important enough to the Intel developers.

NetBSD was the first free UNIX with USB support, the first to do binary emulation, and the first in other arenas, but I've babbled on long enough.

NetBSD is not about experimentation or research, but it's about doing the "popular" thing, either, or having a pretty graphical install utility, or having version 0.001 drivers for some new card. NetBSDers (and core in particular) aren't into wild self-promotion, but what they have accomplished deserves a tremendous amount of respect.

Incidentally, part of the rationale for the research tag comes from having a very clean source-code base. Things are very well organized, such that someone who wants to write something new has a good idea of the structures they're diving into. Many advanced things such as RAIDframe were developed and integrated into the code set. IPv6 support is standard as of November 21. IPSec is already available as patches and will be in the base very soon. UVM is amazing too.

I hope I've demonstrated that NetBSD deserves more than 4 lines of mention in your article.

DIVERSITY
From Max Southall, MIS Director, Kelme USA Inc.
<max@prninfo.com>

Hi Rik,

Interesting that your October Musings on StarOffice have been somewhat fulfilled and beyond, now that it's being given away by Sun and the subsequent proliferation of the Windows-based port. I think at this point it must be over a million downloads, as well as shipping-charges-only CDs ordered.

From experience with most of the office systems that have pretensions of being upwardly mobile (UNIX, Windows, Mac), I have to draw that painfully obvious (to us sysadmins, anyhow) conclusion that the proliferation of Windows has brought with it uncontrollable administration costs.

Pretty well the only practical solution I've actually seen implemented by those who've tried to stick by Windows is the hiring of additional and progressively cheaper bodies to try to keep it all going somehow. And concomitantly, the laying off of the fewer more expensive bodies, because, as they say, it doesn't matter how smart you are under the MS scenario, because it takes just as long to reboot yet again. Could this be the real meaning of "Windows for Dummies"? In any case,
letters

costs keep rising and the level of service sinking.

You know something’s got to give when it takes more time to resuscitate a user’s scrambled PC than it does to restore a well-managed Sun server that serves dozens or hundreds of such users.

Now in my humble opinion, what’s prevented management of the desktop has been essentially the abandonment of the desktop-application market to Microsoft by the major UNIX vendors. UNIX finally matured as the OS platform best suited to the thoroughly networked environment we all find ourselves in, but at the same time, ironically, with none of those PC-styled “killer apps” left that are needed to woo away the disenchanted PC shops.

Except, maybe, StarOffice. A UNIX clone of the lumbering toad MS Office, transformed into a charming thin-client prince? Hey, that’s the ticket!

So I think that Scott McNealy’s thin-client application-services vision for StarOffice is a mite convenient and maybe disingenious. Not yet. As you noted, this is pretty compatible to MSOffice, right down to being a fair imitation of bloatedware installation heft. Also, StarOffice wasn’t even Sun’s idea, although as my friend Gerry Singleton points out, the synergy with ex-Sunner Andy Bechtolsheim can’t hurt. Trying it out last year, my opinion was, after finding it just didn’t quite cut it, that Sun Microsystems ought to buy it and make sure all the rough spots were shined, so that there would be user-level office software available that wouldn’t end up telling everyone in the enterprise where Microsoft wanted them to go today. That and a cup of Java could eventually get us all off the MS dime.

The lesson from the emergence of Linux is that the only strategy with any chance of competing with Microsoft, regardless of merit, is one that gives software away to gain significant market share. Because Microsoft with its enormous accumulated wealth can afford to dump its products until its competitors go out of business, we have seen over and over companies who have pioneered successfully in the Microsoft arena be absorbed or disappear soon after Microsoft decided to enter their markets.

In the case of Linux, whose distribution is modeled in a way even Microsoft can’t compete with economically, Microsoft is worried enough to speculate internally in hysterical fashion.

McNealy’s free distribution of StarOffice punishes Microsoft in the only way it understands — becoming subjected to the same strategy it aimed at everyone else, namely, amputation of cash flow from key product sales.

OK. So that’s the fun strategy for the folks in Mountain View. What about the strategy for MIS?

We need to have manageable systems that encompass the desktop. We can’t have systems becoming ever more unmanageable under an unworkable PC paradigm, or, in the case of what Microsoft has disingenuously offered as enterprise management solutions, with all the important decisions outsourced to Redmond and made with full attention to Microsoft’s cashflow needs rather than MIS’s.

We’re currently running StarOffice on all three platforms to which it’s ported. Because of Sun’s stability and also its focus on its core products, we are deploying Sun servers. And we chose Sun earlier this year after some soul-searching, before the StarOffice acquisition.

There is a window of opportunity right now for vendors like Sun. People are dissatisfied with the Microsoft enterprise path for very serious manageability reasons and are willing to entertain a shift to a more viable approach at this moment. I think that the acquisition of StarOffice and its release in this way are showing that this is the way to go. Personally, I hope that they will build it better, and the customers will come. I just can’t stand the thought of system administration being reduced to carrying a CD fanny pack from user machine to machine, forever. And that’s what’s happened to some of my formerly UNIX colleagues.

cfengine Alert!

Mark Burgess, author of cfengine and the recent ;login: articles on its use, has notified us that the following unauthorized domains have recently sprung up:

<www.cfengine.org>
<www.cfengine.com>
<www.cfengine.net>

He warns people to be wary of them because they might be used to disseminate Trojan-horse versions of cfengine. The official Web site for cfengine remains at Oslo College:
<http://www.iu.hioao.no/cfengine>

complete with checksums for your safety and assurance.
David Parter then presented the best paper awards:
Best Student Paper: “A Retrospective on Twelve Years of LISA Proceedings,” by Eric Anderson and Dave Patterson, University of California at Berkeley.
The LISA 2000 program chairs were announced: Remy Evard and Phil Scarr.

KEYNOTE ADDRESS

Getting the Space Shuttle Ready to Fly
Joe Ruga, IBM Global Services
Summarized by Bryon Beilman

Joe Ruga spoke about his time at Rockwell International as a systems administrator and operations support. He gave a high-level view of the growth of the company and how he thought sysadmins played a role and interacted with the business units of Rockwell.
Some interesting insights that Ruga shared with the audience were:
- High-level management decisions do affect the way one does system administration.
- Befriend the users and their management, so that there is a symbiotic relationship. Be cooperative, not adversarial.
- Document your processes. It does not do anyone any good to hide your
knowledge, and it does not make you any more valuable.

- Document your systems and show the documentation to the other people. They will forget in the first 10 minutes anyway.

Ruga then talked briefly about how he defines goals using purpose, scope, and concept.

**REFEREED PAPERS**

**Using Electronic Mail**
Summarized by Jim Flanagan

**ssmail: Opportunistic Encryption in sendmail**

Damian Bentley, Australian National University; Greg Rose, QUALCOMM Australia; and Tara Whalen, Communications Research Centre Canada

Greg Rose described ssmail as an interim hack to sendmail to reduce the opportunities for snooping attacks while email is in transit. The example of conference attendees being observed logging into home systems as root using the WaveLAN cards distributed at the conference was used as a supporting example of how trivial such passive attacks can be.

The obstacles to a “proper” solution to email snooping are the fact that encryption is not as widely deployed as it should be, and that there is no control over the paths emails take.

Defending against active attacks requires an authentication infrastructure that does not currently exist; so ssmail restricts its threat model to passive attacks (i.e., snooping), and the authors adopt the stance that while snooping cannot be eliminated, removing as many opportunities for snooping as possible constitutes progress. Email snooping is not as uncommon as most people think; there were 26 known occurrences of sniffers installed on backbone segments in a single year in Great Britain.

ssmail is a modification to sendmail that will encrypt an SMTP session wherever possible, but will interact normally with non-ssmail MTAs. An ssmail server will advertise the encryption capability during the EHLO phase of an ESMTP negotiation. Both the message body and the envelope (MAIL From:; RCPT To:; etc.) are encrypted.

When encryption is negotiated, the two parties calculate a one-time session key using Diffie-Hellman key agreement. This key is then employed in a stream cipher (either an RC4-alike or SOBERT32, a cipher developed by the authors, which has a shorter setup time) to encrypt the message traffic. Because the Diffie-Hellman algorithm is expensive and would swamp a busy mail exchanger, ssmail caches the session keys and will reuse these in a faster key-generation algorithm.

Other approaches to solving this problem include S/WAN, FreeSWAN, and IPSec. Similar work includes SMTP over Transport Layer Security (TLS), which is not as efficient, because it doesn’t cache keys, and integrating PGP into MTAs, which would probably swamp key-servers. ssmail is currently in beta test, and users outside Australia will require an export license.

In response to questions from the audience, Greg Rose told us that they did not consider compression, since they wanted to minimize the impact of modifying sendmail, and that ssmail was modular enough to import into other MTAs; in fact someone had already ported it to qmail. Asked about a specific type of denial-of-service attack, Rose reminded the questioner that ssmail takes as a threat model only passive attacks, and DoS attacks are active.

**MJDLM: Majordomo-based Distribution List Management**

Vincent D. Skahan, Jr., and Robert Katz, The Boeing Company

MJDLM is a suite of tools built around Majordomo that allows Boeing to replace paper-based company-wide announcements with short email messages (on the order of 40K each) that point people to more detailed information on an internal Web site. List membership is determined by an employee’s status in Boeing’s corporate personnel database.

There are about 125 ongoing lists, generated every week, with the possibility of building temporary targeted lists for special purposes. Because of the possible impact of sending messages to, in some cases, 140,000 recipients, the messages need to be approved by Boeing Public Relations; not just anyone can send messages.

List creation is kicked off from a Web interface and sent to DBA staff, who construct an SQL query based on the request. A general sanity check (number of recipients, etc.) is done on the results, and then a list is built or rebuilt. Large changes in the size of a list result in staff notifications. Alternate databases can be used to generate lists, or for additional selection criteria.

The flow time for mailings runs from 12 seconds to six hours, depending on the audience. The authors look at the last bounced message to place a lower bound on the total flow time of a mailing, from sending to last recipient delivery. Bounced messages are sent to a procmail-filtered mailbox and categorized by the cause of the bounce. The addresses in the lists are invariant, layer-of-abstraction-type addresses that are translated to real delivery addresses by sendmail.

Sendmail 8.8.8 is used for its ability to employ additional alias databases. The translation process can take up to an hour for large mailings, and one planned
improvement is to populate the lists with real addresses when they are built.

JDLM is hosted on a single production server, with a redundant standby and two to three geographically distributed fail servers.

Users don’t have the option to unsubscribe, but the team experienced little resistance from the user community, and the system provides a communications channel from the CEO – wherever in the world he happens to be – directly down to the line workers.

The question from the audience was, “How did you get HR to give you access to the personnel database?” to which the answer was, “The CEO told us to do it.”

RedAlert: A Scalable System for Application Monitoring

Eric Sorensen, Explosive Networking, and Strata Rose Chalup, VirtualNet

The goals behind RedAlert are the ability to integrate it into any environment and use intelligence to differentiate obscure failure modes from ordinary transient failures (“to page or not to page, that is the question”). The motivation for the project came from dissatisfaction with large commercial network-monitoring systems that do everything but what you want without the assistance of vendor professional-services staff, and having to write “meta monitoring” systems on top of commercial monitoring systems.

RedAlert’s architecture provides a central (or multiply distributed) “alerting” daemon for the aggregation of status traffic and dispatch of alerts, with “testing” clients written around a provided client API. The system is written in object-oriented Perl, and clients are subclassed from the RedAlert::Client module.

Client Communication with the alerting daemon is done by serializing Perl code with the commonly available Data::Dumper module, and sending it via a TCP socket. Separating the alerting and testing functions like this makes it easier for sysadmins to incorporate existing system-monitoring scripts and tools into the RedAlert framework.

The alerting system supports alpha paging, SNMP traps, and email for notification. The daemon is highly configurable and allows for detailed definition of notification thresholds and methods, and different messages based on the category of alert received from clients. The presentation mentioned, but did not elaborate upon, the ability to treat certain alerts as being diagnostic for larger problems (e.g., are all the printers down or is there something wrong with the network?) and only send notifications for the larger problems.

After the talk, Elizabeth Zwick stepped up to the mike to verify that she heard the speaker say that the clients sent evaluable code to the server without any sort of authentication or control over reconstitution, and Eric Sorensen acknowledged that this was an area where there was room for improvement.

The Way We Work

Summarized by Josh Simon

Deconstructing User Requests and the Nine-Step Model

Thomas A. Limoncelli, Lucent Technologies/Bell Labs

Tom Limoncelli provided insights on how he developed the nine-step model of systems administration to help reduce user complaints and to get the problem reported correctly the first time more often. The steps can be broken into four phases:

The Greeting (“Hello!”)

1. The Greeting

Problem Identification (“What’s wrong?”)

2. Problem Classification

3. Problem Statement

4. Problem Verification

Planning and Execution (“Fix it”)

5. Solution Proposals

Solution Selection

6. Execution

Verification (“Did it work?”)

7. Craft verification

8. User verification/Closure

Skipping steps can lead to solving the wrong problem (steps 2–5), choosing a solution that doesn’t solve the problem (step 6), making a mistake executing the solution (step 7), not checking our own work (step 8), or having the user call back with the same problem (step 9).

Adverse Termination Procedures, or, “How to Fire a System Administrator”

Matthew F. Ringel and Thomas A. Limoncelli, Lucent Technologies/Bell Labs

Matthew Ringel discussed how to fire a system administrator, as well as what to do when you’re the one being fired. The paper itself contains several case studies from which the authors created a three-tier model of areas of concern:

- Physical access, such as to the campus or facility or closet or desk drawers.
- Remote access, such as being able to access systems or networks or information remotely, as in login access.
- Service access, such as access to information services over the network (databases, intranets).

In summary, if you’re in the unenviable position of firing a system administrator, you need to ensure that all three tiers of access are closed properly, because leaving one or more undone can result in a disgruntled person with superuser privileges having access to your systems, networks, and data. A last word: whether you’re the one doing the firing or the one
Organizing the Chaos: Managing Request Tickets in a Large Environment

Steve Willoughby, Intel Corporation

Steve Willoughby discussed not only software, but also the infrastructure to support customers' needs. Status reports are essential; data that proves what and how much you do is absolutely required for management to increase, or, sometimes, even maintain headcounts and budget. Electronic mail and simple scripts are okay for managing problem reports but tend not to scale well in an enterprise environment. Willoughby's group designed and implemented a new system to meet the needs of both customers and system administrators.

Having service-level agreements (SLA) with senior management on both the customer and support sides is required. Intel also rotates its senior people onto the help desk, automates processes, and allows the user to control the closure of a ticket. They've found that this system scales better, results in a lower administration/user ratio, and results in users having more control over their problem reports and feeling happier about the process.

Future plans include more work on root-cause analysis to help resolve problems before they become disasters.

Tools

Summarized by John Talbot

GTrace – A Graphical Traceroute Tool

Ram Periakaruppan and Evi Nemeth, University of Colorado at Boulder and Cooperative Association for Internet Data Analysis

GTrace is a graphical network mapper based on the traceroute program. It was designed to view true physical locations of network hops in an effort to determine network path efficiency. GTrace is written in Java for system portability and wide platform use. Like any network-tracing program or suite, GTrace has fallen victim to the lack of any real correlation between the IP numbering of a network and its actual physical location. This is most prevalent in global networks and network clouds that mask large geographical ranges of networks and routing equipment behind multiple gateways and intelligent active mesh networks.

As with any interesting problem, there are always interesting solutions. The location-detection problem would be easily solved if the LOC resource records in DNS were available for every name and IP reachable on the Internet, but these records are not generally used by most organizations for considerations of security and overhead. There is no IP-to-location master database anywhere on the Net – such a database would be massive to implement and daunting to maintain on a full-time basis. If the maintainers' IP ranges were used, there would often be discrepancies between the billing addresses of the maintainers and the actual location of the networks under such authority. While none of these problems has a direct solution, some information gleaned from these sources can be used to rule out erroneous information in the data collection process of GTrace. Some solutions have been intuitively applied to the data-collecting features of GTrace that assist in the location-determination process.

The developers of GTrace have used some novel techniques to zero in on location data. Using an even-step search and comparisons of known round-trip times (RTTs) from previously measured or known sources, erroneous location information can be excluded and more suitable location information can be deduced. This method has been deemed as the "clarifier" part of GTrace that marks such flagged RTTs for further inspection and prevents inaccurate information from providing answers for physically impossible situations and data-transfer rates. For the known quantities GTrace comes with an initialization database that contains machine, host, city, organization, and even airport information (no, you can't use GTrace to book a better airfare, sorry). As an extension, the NetGeo online lookup server has been created to track an impressive 76% to 96% of RIPE, APNIC, and ARIN WHOIS records. GTrace also has extensions to let the user/programmer add customized databases, file stores, and text files for additional geographical and data lookups.

For the user-interface and program extensibility, GTrace provides a sleek interface for mapping location information and onscreen segment and network hop data-display tables. Additional features, such as the flexibility to use third party traceroutes, the ability to add new maps, and a zoom feature, make GTrace a very adaptable and versatile tool. For more information, see <http://www.caida.org/Tools/GTrace>.

rat: A Secure Archiving Program with Fast Retrieval

Willem A. (Vlakkies) Schreuder and Maria Murillo, University of Colorado at Boulder

The rat archiving program is built to be a small and fast means of backup and restore that is secure, robust, and extensible with extended support. Schreuder and Murillo have implemented rat to use a new archive layout specifically designed for fast access. Schreuder notes that the archived filenames reside only in the archive table of contents (TOC) and not the archive for space minimization, and the date stamps and modification times are stored as 64-bit integers (jokingly referred to by Schreuder as Y2038K compliant). It was noted as a caveat that the archive can become corrupted if it
Cameron's truncated, since some or all of the TOC can get lost. A file-pointer layout can be used for TOC-rebuild purposes.

For implementing security, rat uses MD5 and PGP for encryption and checksums. For performance selection, rat offers an open ability to choose from several compression and extraction options. Also, individual configuration needs and file-compression options can be specified by using a personalized .ratrc file, making rat extremely versatile. The librat library enables the rat archiving and extraction procedures to be at the program level. The Qt library is used to implement a GUI interface for accessing rat archives at the user level.

The rat paper was widely accepted by the attendees at the conference. It is important to note that it is always a good thing when a presentation is followed by a feast of deep technical questions from some of the greatest talents in the group. One attendee suggested that optimization ideas could be handled on I/O levels below the file system itself, and Chreuder, displaying his deep understanding of this new technology, walked through a detailed explanation of seek, open, and close operations that could be used to perform such operations. Other suggestions – using signed integers for the modification/date stamps, and storing ACL information in the archive metadata – were well received by Chreuder.

More information can be found at http://www.perl.com/rat/index.html.

Cro-Magnon: A Patch Hunter-Gatherer

Jeremy Bargen, University of Colorado at Boulder and Raytheon Systems Company; Seth Taplin, University of Colorado at Boulder and ITIR, Inc.

Cro-Magnon is a tool that has been designed to hunt down patch and software updates from the Internet and other named sources, and compile updates and accounting for upgrading as needed. Currently, there is no widely accepted process for full-cycle retrieval, installation, and accounting for system updates. At the system level, no generalities apply in regard to version control and patch location, configuration, and implementation. Cro-Magnon provides a starting point for a scalable, portable, and versatile method of system-update recovery.

At the heart of the Cro-Magnon suite is an engine that is surrounded by download, authentication, notification, and GUI mechanisms and controls. It is written in Perl and thus can provide virtually infinite module flexibility. However, module implementation is not standard on all UNIX platforms and becomes even tougher ("if not impossible") on NT. Complex module variations and large config files are needed to keep track of large, heavily varying system layouts, since not every system in a heterogeneously operated environment would need to be at the same revision level at the same time.

Ongoing development is planned for Cro-Magnon and its documentation. Greater stress testing is planned for the Cro-Magnon engine. The configuration-file layout may get broken into sections to alleviate the need for a flat master file; why not modularize the config, since the process that runs it is modular? Also, there is an open door to implement existing tools, such as wget, to aid the engine functionality.

While Cro-Magnon doesn't automatically apply the patches, it can save system administrators a large percentage of the time involved in system updates, since retrieving and comparing current against future patches (for those of you who don't patch-and-pray) is 90% of the work. It would be nice to see some standardization in the UNIX patch world. I can imagine vendors sending out their systems updates and software with a Cro-Magnon module, so you install it once and the process takes care of itself for future updates. For a tool that was designed by software developers to simplify their system maintenance and headaches, it has the potential to end a lot of tedium for others, not just for its creators.

Thinking on the Job
Summary by Jim Flanagan

A Retrospective on 12 Years of LISA Proceedings

Eric Anderson and Dave Patterson, UC Berkeley

The authors won the Best Student Paper award for this paper.

Eric Anderson provided a quick overview of the categorization of the 342 papers presented at past LISA conferences, calling out the trends, patterns, and insights gained from the study.

The major pattern: Papers were written either from the point of view of system administrators or from that of academics. The former work tends to be practical and realistic, though repetitive, and the latter work tends to be extensive and detailed, but irrelevant. Why? Since system administrators tend to be busy, they end up all solving the same problems, whereas the academic isn't close enough to the day-to-day work to understand the real problems faced by system administrators. Eric Anderson urged the two camps to work together to produce thorough, relevant research into tools for system administration.

Two other categorizations of the data were presented: the source of the problem the work was trying to solve (the source model), and the tasks focused on in the work (the task model). The main insight gained from examining the papers using the source model is that while system administrators divide their time about equally among configuration...
management, maintenance, and training tasks, the content of the papers written did not reflect this division. Papers related to configuration-management problems were most prevalent.

Based on this detailed examination of the task model, they recommend moving toward a single methodology for OS and application software installation and package management. Anderson also mentioned that end-user configuration customization hasn’t received a lot of attention in recent years.

A trend seen in the task area of configuration management is that corporate mergers, acquisitions, and divestitures, as well as growth in the IT industry, are driving the need for more site moves and related work. This is causing paper authors to look toward a theory of site design that facilitates site moves. Also, a more mobile user community is inspiring growth in the number of network-configuration-management papers.

It was found that more energy was being spent in papers on the performance of backups than on the more critical performance of the restores. The areas of technology trends, security, and archival storage are neglected among the papers.

For email, the noticeable trend is that there were many papers in the earlier years, then a pause in mail research until 1996, when the Internet began to swell and spam, scalability of delivery, and security became bigger problems.

Anderson concluded by repeating that the work by system administrators is repetitive, and that a database of related work would help to alleviate this problem. System administrators might also find benefit in providing guidance to academics looking for research topics rather than striking off on their own to develop solutions to problems.

The raw data and categorizations are available from the authors, who encourage further analysis of the material.

One questioner from the audience wanted to know how the authors determined that systems administrators spend roughly one-third of their time on the three problem sources. Anderson replied that they surveyed members of the community.

Managing Security in Dynamic Networks
Alexander V. Konstantinou and Yechiam Yemini, Columbia University; Sandeep Bhatt and S. Rajagopalan, Telcordia Technologies

Alex Konstantinou began by admitting that this work fell into Anderson’s “irrelevant academic” category. He then defined a “dynamic network” as one in which the network elements, services, and policy can change. To maintain security, one must manage the configuration of the elements and services in a way that maintains the policy when something changes.

Configuration management is difficult because it is human-intensive and involves distributed, heterogeneous data. Errors are often introduced because there is no way to verify that configurations actually reflect policy, and mistakes have to be undone by hand. For this reason, a network tends to be reconfigured only if there is a compelling reason to do so.

Conversely, policy decisions have complex implications for the configuration of elements and services. A simple change in policy might require changes to switches, VPN configurations, filesystems, routers, and more. System administrators should be making changes in a more abstract layer, not at the network-element level.

The authors’ proposed solution involves placing a Unified Configuration Semantic Layer between the policy definition and element configuration that employs consistency checking, change propagation, and rollback and recovery functions. Their work builds on the NESTOR network-element-management system developed at Columbia. NESTOR maintains consistent configurations by imposing constraints (such as “All hostnames must be unique”) in the form of Object Constraint Language, which is part of the Unified Modeling Language (UML). If the constraints are not fulfilled, then either an error is flagged or policy script can be executed.

The authors use NESTOR constraints to model security policy and also to provide a first attempt at an abstract “universal platform” that can be mapped onto various network-element configuration models.

Deployment of NESTOR for security management involves the creation of a policy, the abstract modeling of the network elements and services, instrumenting the actual network element interfaces, translating the policies into constraints and policy scripts, then deploying and populating a NESTOR server with the above. It should be in the interests of network-element vendors to provide fully instrumented interfaces to their products, if a standard universal platform specification existed.

The authors project that the role of the system administrator will shift to the manipulation of abstractions rather than the direct configuration of elements and services, because the latter does not scale to large, complex networks.

A question raised from the audience was what happens when the NESTOR server fails for some reason? In that case, if configuration changes were needed they could still be done by hand, but these would not be protected by NESTOR’s constraint checking and rollback functions.
A member of the audience asked why, if Prolog allows the creation of new primitives, you would use a preprocessor. The answer was that they wanted to enforce strong typing, something Prolog does not support. More discussion ensued about the implications of accidentally using literals instead of variables and the severe damage that could be caused by the quirks of Prolog. The consensus was that safety features would have to be included before such a Prolog-based system would be a reasonable system-management tool.

Network Infrastructure
Summarized by Bryon Beilman

NetReg: An Automated DHCP Registration System
Peter Valian and Todd K. Watson, Southwestern University

Managing Southwestern University's residential network DHCP usage was becoming difficult. They needed to create a unique, simple, maintainable, and inexpensive method to register and validate a DHCP user on the residential network. Since DHCP is virtually anonymous, they needed to be able to verify who was actually plugged into the end of the computer.

The solution that Peter Valian presented met their requirements and involved a unique way of forcing the users to register their IP addresses using the DNS server fields of the DHCP information. Before they are registered, the DHCP records force them to a fake DNS root server that resolves all addresses to the registration page. Once they register and enter their university account name and password, the software modifies the DHCP configuration file and allows the user to use the network. They are working out some security issues, but the system is low-maintenance and helps to ensure that only authorized and registered users can use their student network. More information can be found at <http://www.southwestern.edu/ITS/netreg/>.

Dealing with Public Ethernet Jacks – Switches, Gateways, and Authentication
Robert Beck, University of Alberta

The University of Alberta faced the problem of managing their public Ethernet jacks. There were many mobile laptops, quake servers, PCs with root access, and the possibility of nonuniversity users just walking up and plugging into their network. They wanted the same level of control that they had on their multi-user UNIX systems, and the solution should be consistent and easy for the end user. After investigating many solutions, they decided to use an authenticating gateway, which forced a user to authenticate before allowing access.

They wanted to make sure that users cannot "snoop" each other's packets, to prevent (or limit) spoofing, and to disallow broadcasting of unknown traffic. Their solution involved using a gateway based on OpenBSD that blocks all outbound traffic using packet filters until they authenticate. The user can telnet to the gateway and authenticate, and the traffic is allowed through the gateway. They also monitor ARP tables using swatch on syslog to monitor IP spoofing and take action.

They also use an ident server that rewrites all outbound mail addresses with the users' real names and addresses (that they used to authenticate), so they cannot fake their email addresses. The system works well for them and it is easy for the students to use. More information can be found at <http://www.ualberta.ca/~beck/lisa99.ps>; the code for this solution can be obtained at <ftp://sunsite.ualberta.ca/pub/Local/People/beck/authid/>.

NetMapper: Hostname Resolution Based on Client Network Location
Josh Goldenhar, Cisco Systems, Inc.

Josh Goldenhar developed NetMapper to help provide a framework for resolving hostnames based on the client host's location within a network hierarchy.
NetMapper can provide a method to resolve general or virtual hostnames to an interface-specific or real hostname, provide an optimized NFS server selection for remote filesystems, or organize your organization. The paper describes a use of NetMapper to modify the local /etc/hosts file, but it can be used for many other configuration needs. The user will always have to enter the organization's network into the NetMapper, which includes definitions of mappings between network and network groupings.

Goldenhar gave the example of using NetMapper as a customized Netscape wrapper that allows Netscape to start with different URLs depending on the client location. This can be used to direct the user to the local cafeteria, help-desk number, or some other category that is based on the network grouping.

The second example demonstrated how it could be used to route trouble tickets that came in from a Web form to the local help desk for a traveling user. Salespersons or other people on the road can get their problems routed to the geographically nearest help desk to allow rapid resolution.

The tool is flexible and can do more than was mentioned. More information can be obtained at <ftp://ftp.eng.cisco.com/josh/NetMapper.tar.gz>.

File Systems
Summarized by Mike Newton

Enhancements to the Autofs Automounter
Ricardo Labiaga, Sun Microsystems, Inc.

Ricardo Labiaga spoke about the enhancements his company has made to the automounter. The major improvements over the last version are new abilities to browse maps, improved concurrency, and better reliability. The new version, present in Solaris 2.6, was redesigned with three components: the autofs filesystem, a kernel virtual filesystem that triggers the mounting and unmounting of filesystems; the automounter daemon, a user-level process that performs the mounts and unmounts; and the automounter command, a user-level program that installs the initial entry points. The old automounter had problems with filesystems disappearing, inability to perform concurrent mounts, and occasional hang-ups. These problems have been corrected. The browse function was implemented with lazy mounting. For instance, if /foo/bar were mounted when you changed to that directory and then did an ls, all the subdirectories would be listed, but only by doing a stat. The mounts of the subdirectories would not be triggered. This means that in Solaris 2.6 you can use automounter for home directories, and when you ls /home you would see all the users' directories listed, not just the ones you've actually entered or your own. The enhancements to the autofs automounter are welcome update to a very useful product.

Moving Large Filesystems On-Line, Including Exiting HSM Filesystems
Vincent Cordrey, Doug Freyburger, Jordan Schwartz, and Liza Weissler, Collective Technologies

Vincent Cordrey described the authors' method of moving data from one data-storage system to another. You would use a "just plain copy" of files to a new system, or what was described as forward relocation, reverse relocation, or a hybrid of the two. The forward and reverse relocation methods were the meat of the paper, since the just plain copy is what the name implies (i.e., using tar, cpio, or cp to copy files). The forward and reverse relocation methods require you to have the old and the new filesystems online at the same time. With forward relocation you replace files on the old filesystem with links pointing to the new filesystem, after you copy the files to the new filesystem. However, you have to make sure links are copied as well as files. Another drawback is that files are still being created in the old filesystem. Eventually you should get to the point where the old system contains only links to the new system. You can then mount the new filesystem in place of the old. With reverse relocation, the new filesystem is mounted in place of the old, and symbolic links are created to all the files in the old filesystem. With this method, new files created are created on the new filesystem. The drawback is that you will need a long enough downtime to create the new filesystem with links to everything in the old filesystem.

Systems
Summarized by Bryon Beilman

Service Trak Meets NLOG/NMAP
Jon Finke, Rensselaer Polytechnic Institute

Jon Finke described how he integrated NLOG/NMAP with ServiceTrak (two previously developed tools) to simplify the analysis of information and make it useful. NMAP is a port scanner that identifies TCP/IP fingerprints for OS at open ports, while NLOG provides some data management and a Web interface. The author used these tools with ServiceTrak to map out the network and identify security exposures.

This combination of tools allows the user to identify site-configuration errors verify that some new work has not inadvertently turned on a service, and validate the security settings of a host. Some of the lessons learned are that host grouping is very useful, knowing the OS is very handy, and there may be some policy issues with running this kind of tool on your network. More information can be found at <http://www.rpi.edu/~finke>.
Design and Implementation of a FailSafe Print System

Giray Pultar, Coubros Consulting LLC

Giray Pultar described the system he developed to provide fail-safe printing for a heterogeneous environment. In his configuration there is no spooling on the client; all jobs are sent to the printhost, and there is no printer information on the client. This generic client setup provides easier and more consistent printer setup. Only one host communicates with the printer, so the queue information is more accurate. He has a dynamic printmap based on LPRng, does printjob routing, and allows notification using ephyr.

A nice feature of the architecture is that you can print to low-cost printers attached to the back of an X-terminal while still utilizing the centralized spooling model. The system can also route print jobs from VM and VMS to all printers on the system. Pultar can be reached at giray@coubros.com.

Installations
Summarized by Jim Flanagan

Automated Installation of Linux Systems Using YaST
Dirk Hohndel and Fabian Herschel, SuSE Rhein/Main AG

YaST is a SuSE-specific tool for automating system installations that has shipped with SuSE Linux since version 5.2, but most administrators are not aware of its capabilities. Hohndel characterized his talk as "stealth marketing" for the "other Linux distribution." YaST's goal is to provide unattended, automated, flexible, reproducible installations that are easy to manage and control remotely.

While proprietary UNIX vendors have tight control over the booting process, an obstacle for Linux is that most PC systems have poorly implemented, nonstandard BIOSes. One can usually count on being able to boot from the floppy, and this becomes the lowest common denominator, though NIC-based network booting solutions are becoming popular. Floppies and unattended systems are, however, an impediment mismatch. SuSE system come up running after an install, without having to reboot the system, since some systems can hang because of BIOS-related problems. This improves the unattended install process.

The SuSE boot process provides a way to put the system definition on the boot floppy. This can be defined entirely or you can factor the common configurable parts into a system definition and get the network configurables from a DHCP server. Info files can also be defined for certain classes of hosts, and hosts can be in several classes.

To account for differences in disk layout, YaST uses heuristics to determine how to put filesystems on the available partitions. Package selection can also be predefined with a config file, which can be built by going through a package installation interactively once, and then mas-
that performs pathological error checking such as comparing the hosts and ethers tables against reality. Config also knows something about the installation infrastructure and chooses the correct boot/install servers for a given client, and it can mark nonstandard hosts so that they don't get Jumpstarted.

To automate the actual installs, another tool was created, called Start, which, after a few last-minute sanity checks (such as whether users are logged in), forks into multiple processes that log into the clients to kick off the Jumpstarts. Given that each client takes about 15 seconds to initiate, a single-threaded application is not sufficient for hundreds of hosts in a one-hour window.

The status of the installation was available on the Web, so that a small team of admins could react quickly to any problems that might arise during the process. Heiss was about to describe how users were notified about impending reboot of their machine when the hall reverberated with a loud “Warning, Warning,” and I thought that we were all going to have to leave the building. But the warning continued: “Your machine is about to be Jumpstarted. Please log off.” This was one of the suite of warnings that could be piped to /dev/audio if a user was still logged on to a machine that was targeted to be Jumpstarted.

For their infrastructure, rather than waste a machine as a boot server on each subnet, the team used multi-homed hosts on several subnets each. The bandwidth requirements, based on estimates of 500MB/client, 200 clients/hour, give 200MB/second at the server end. This will keep a server with three switched 100-BaseT interfaces fairly busy and will have a significant impact on your network; Heiss recommended that shared Ethernet be avoided in this situation. The net booting process results in about 60 SFS93 NFSOps/sec/client, and so an Enterprise 3000 class machine can serve about 150 clients. Jason also recommended that the data be striped (RAID0 or RAID5) to increase the performance.

During the question session, one person asked how to deal with locally installed software. Jason replied that though they had to deal only with dataloss clients, local apps could be installed using a Jumpstart finish script.

**Automated Client-Side Integration of Distributed Application Servers**

Conrad E. Kimball, Vincent D. Skahan, Jr., David J. Kasik, The Boeing Company; and Roger L. Droz, Analysts International

As part of Boeing’s “Single Glass” initiative to build a computing environment where an engineer needs only one desktop computer, the authors needed a way to present a unified and integrated view of applications from all workstations, and from both the shell and the CDE desktop, for about 5,000 workstations with 200 applications (comprising some 8GB of space).

The solution involved separating the public view of the application file space from a private view, so that applications could be upgraded or moved behind the scenes without the users modifying their behavior. Multiple version of applications can be maintained, and the applications are built using the private namespace. Both the private and public views exist under a /boeing directory, with the private application-directory hierarchy mounted from several file servers under /boeing/mnt.

The public directory hierarchy is then script-generated as a series of symbolic links in /boeing/bin, /boeing/lib, etc.

In response to a question from the audience, Vince Skahan said that they had attempted an NFS implementation of this scheme, but met with limited success and are going to stick with NFS.

**INVITED TALKS**

**Deep Space BIND**

Paul Vixie, Internet Software Consortium

Summarized by John Talbot

Who better to present the Deep Space BIND talk than Paul Vixie! Welcome to deep history of BIND with a scope targeted on the protocols, implementations and special interests that have established DNS for well over a decade and left it virtually unchanged for nearly 15 years, and on the DNS MIBs, completed in 1992.

Recently BIND services have improved significantly. New resource records and classifications have been implemented, but Vixie noted that deployment of many resource records has been difficult over the years because of the overhead required to maintain such records and the questionable usefulness of the information that they represent for public Internet DNS queries.

BIND 8.2.2 was released a few weeks before the conference, and BIND 9 has been in production for about a year. BIND-4 was feature-frozen in 1995 at version 4.9.5 and has had only security and bug fixes released since then. The latest release, BIND-8, version 8.2.2, features greater security, performance, usability, and RFC conformance. Also, BIND-8 has features for selective zone forwarding and an asynchronous resolver for processing multiple transactions using pthreads to enhance performance. Vixie advised all to move away from BIND-4 since it “just does wrong” with such attributes as panics on oversized messages, promiscuous data sharing, and the compression of names.

BIND-9 was a complete ground-up rewrite with the objectives of open source, basis on IETF standards, scalability, and a “carrier grade” production-quality product. Surprisingly, Paul Vixie has had no hand in the coding for
ND-9, since his massive BIND expert-
has been required for ongoing sup-
port of BIND-8, and he is “planning on
iring” from being the BIND-master, as
is colloquially known. Vixie modestly
ated this aspect as a “good thing,” since
was of the opinion that the BIND-8
should have been written from the
und up as well and many of the
ND-4-isms were brought into the
ND-8 release simply because of pro-
ummer familiarity. Those doing the
write have performed a complete
structuring of BIND and placed new
phasis on security, performance
ability, and RFC conformance.
her efforts are also in place to expand
usability of DNS. Extended DNS has
ed it harder to add security to the
tent protocol. Some transaction sig-
d have been proposed to address
orization and signed keys. Secure
S (DNSSEC) implements zone
thenticity through public-key encryp-
using a parent-child keytrust for
information and transaction-signa-
e (TSIG) relationships between
wn servers. Also noted was the fact
that caching verifications is gen-
ally bad for security. One problem of
is that the GSSAPI in WIN2K does
not implement the normal ISC TSIG and
not compatible with the current ticket
format.

A final note, BIND has been released
under a BSD-style licensing agreement to
omote broad implementations of
ND, which Vixie hopes to benefit an
bunding economy.

Four-Star Approach to Network
agement
If R. Allen, WebTV Networks, Inc.;
ial Williamson, Global Networking
Computing, Inc.

summarized by Altaf Rupani
is session attempted to provide an
ative to traditional all-inclusive sin-
-vendor network management solu-
tions. The speakers advocated a modular
pproach to network management.
Their philosophy for the management
vironment at WebTV was to avoid the
endor approach: “Deploy monolithic
lication/framework and solve all
blems directly or with add-ons.”
Such an approach results in a complex,
complete and virtually unmanageable
plementation and would also over-
shoot their budget.
They split their requirements into four
parts and then identified tools address-
ing those requirements.
- Trending and thresholding - Cricket
- Alert management - Netcool
- Workflow management - Remedy
- Dashboard approach to problem
olving (a single interface to the above
3) - Homegrown solution

A modular approach allows incremental
improvement in the network-manage-
ment infrastructure. It also reduces the
isk of having a large implementation of
vendor-specific product to address a
small need.
However, it requires a lot of effort in
implementing each of the components
and then making them work together.
Such a solution may be less reliable, since
it contains many components working
together that may not have been tested
thoroughly before. It also requires con-
siderable knowledge of each of those
components.

Their conclusion: Although the four-star
approach requires effort and care during
plementation, it provides administra-
tors and managers with tools that direct-
ly apply to their site and gives them con-
trol over their environment.

Microsoft's Internal Deployment of
Windows 2000
Curt Cummings, Microsoft,
formation Technology Group

summarized by Eric Jones
The goals of the Microsoft deployment
were to:
- Showcase how to migrate to Windows
000. Clearly, if Microsoft believes that
their customers should migrate to this
new operating environment then they
need to show the confidence to take
the same step.
- Provide feedback to development.
Since no one has deployed Windows
000 on this large a scale before, ITG
will certainly expose bugs and weak-
nesses in both the software and the
igration tools. By feeding this in-
formation back to development, they
can ensure an easier transition for the
customers.
- Clean up sins of the past. Like many
organizations, as Microsoft's internal
computing infrastructure grew, it
became somewhat disordered. The
Windows 2000 rollout offered an
portunity to restructure the com-
pany's domain model and reduce admin-
istrative complexity. The Windows
000 Active Directory structure made it
much simpler to model the corpo-
rate structure.
- Establish a new desktop-management
rogram. Some of the new features of
Windows 000 enable more sophisti-
cated centralized management of desk-
top configurations. Since every
achine in the enterprise needed to be
built anyway, this was a great oppor-
tunity to bring them all under man-
age.

The planning process began even before
the first beta release of Windows 000.
At this stage they decided on a geograph-
ic organizational structure and a five-
phase rollout.
Phase I was done using Beta 2 of Windows 2000. It was rolled out to 6,000 workstations in the engineering groups in Washington.

Phase II was also restricted to Washington, but included 15,000 workstations. At the same time, 10 resource domains were collapsed to five organizational units (OUs).

Phase III included 25,000 workstations.

Phase IV included 48,000 workstations and collapsing 150 resource domains to 50 OUs.

Phase V, which was not complete as of LISA, was full deployment worldwide. The expected completion date was mid-December.

Cummings discussed the challenges that this migration faced. These included resistance from local administrators worried about losing administrative control of their systems in the consolidated admin structure, lack of tools for synchronization of data across AD “forests,” and the need to continue to support NT 4 for ongoing interoperability testing.

Real World Intrusion Detection
Mark K. Mellis, Consultant, SystemExperts Corp.

Summarized by David J. Young

I've never been too excited about the topic of security, since it brings to mind an image of the corporate security guard rummaging through my backpack looking for “bad things” as I enter or leave my place of employment. Intrusion detection, on the other hand, invokes a stimulating "cat-and-mouse" response, much like the adventure described in The Cuckoo's Egg.

Mark Mellis gave an excellent presentation on what intrusion detection means, how it impacts your organization, what kinds of intrusion detection to implement, and how to deploy intrusion detection.

An often overlooked but extremely important first step in implementing intrusion detection is to establish policy. What are you trying to protect? Who assumes the risk? How do you protect the company when under an attack? Who has the authority to take down the site in an emergency? Questions like these need to be addressed before an effective intrusion-detection strategy can be deployed. Intrusion detection may involve decisions and actions regarding sensitive issues. Privacy concerns or other company policies may impact how you approach your implementation.

Effective intrusion detection also requires comprehensive training. Subscribing to mailing lists and attending conferences and tutorials help people to stay current with the latest methods. Just as important is for your staff to be familiar with all of your tools used for intrusion detection. Simulate a real-life attack to test your staff's ability to detect, classify, and respond to an external threat. Include the real decision-makers so that they too are prepared to make the important decisions.

There are four main types of intrusion detection: network, host, application, and analysis.

Network intrusion detection offers real-time analysis. Some "smart sniffers" provide this ability, but typically network activity is logged for later analysis. Newer routers offer dynamic reconfiguration based on real-time events. This means they dynamically create and destroy path(s) through the firewall by looking for signatures in network traffic. Network intrusion detection is generally nontrivial to setup and maintain.

Host intrusion detection is an area most familiar to sysadmins. It involves instrumenting the host with tools to monitor host activity. Some of the more popular tools include tripwire (file integrity using checksums), klaxon (port masquerader), tcp-wrappers (track connections), and syslog (log system events).

Application intrusion detection is analyzing unusual application behavior. An excellent example is a typical e-commerce configuration. A Web application running in a demilitarized zone speaks SQL to a database on a secure net. It is assumed that the Web application makes bug-free SQL database queries. If there are SQL errors in the database logs, it may indicate that someone has compromised the Web server and is performing ad hoc queries against the database.

Analysis is another important component to a good intrusion-detection strategy. A restricted-access machine is used as a centralized logging server to store syslog and other data for daily analysis. Simple hourly/daily/weekly reports are then generated, such as:

- Top 10 logins (who/where)
- Login idle more than 3 weeks
- su events
- System reboots
- Router reconfig
- SUID files added/deleted
- Login successes/failures

The information contained in these reports may indicate an unusual event or trend that requires a proactive response.

Intrusion detection is not a project but, rather, a process. It is the detection of, classification of, and response to a network or system event. Implementing different types and levels of intrusion detection, and correlating and analyzing the results, will help you to detect and respond to real-world intrusions.

The System Administrator's Body of Knowledge
Geoff Halprin, The SysAdmin Group

Summarized by John Talbot
Neither the threat of government censorship nor that of the conference center burning down could have kept Geoff
Alprin from delivering his message of developing a maturity model for system administration. Alprin, who endured a series of general fire-control-system false arms during the presentation (what a coup!), opened his talk by describing a number of electronic-communications regulations recently invoked in Australia.

Which, based on uninformed decisions and a lack of understanding of what it would take to administer such regulations, effectively result in censorship and reduced protection of copyrights.

Alprin's message was that if we do not take more of an interest in what is happening around us, it will happen to us. Developing a better definition of the body of system-administration practices could help prevent many of the problems we face; it can enhance the ability of people, businesses, and governments to make informed decisions about practices that depend on system-administration support. Alprin noted that with the increasing use of e-commerce, the need to take a disciplined approach to system administration has moved into the spotlight. It is no longer just the concern of IT departments.

The Systems Administrator's Body of Knowledge (SA-BOK) is being designed to help address these and other system-administration issues through defining the profession and its core elements. One of the key steps toward this goal is to define a taxonomy schema that provides a foundation for expectations, deliverables, and functionality of system administrators.

Alprin listed the roles of system administrators as troubleshooter, "the walking encyclopaedia," toolsmith, researcher, student, technical writer, strategist, tactician, and even a "doctor and counselor" to some. System administrators face such problems in the workplace as lack of understanding from management, lack of accurate reporting metrics, lack of standards, lack of time for proactive work, lack of boundaries (where the job's role starts and stops), and the demands of ever-increasing business needs.

Core to all of these is a lack of clear understanding of what our role really entails, with a consequent inability to communicate the needs of that role (time, money, resources) to other communities such as management and government. To help manage such problems, better definitions of what system administrators do, what is needed to do their jobs, and methods to identify difficulties must be developed. Also, the system administrator's image and visibility must be clarified, so that system administrators can readily answer the question, "Where the heck were you when it hit the fan?"

As the system-administration field grows, greater emphasis is being placed on availability, standards, and the nature of the job. Meanwhile, the system administrator is expected to understand every detail of a constantly changing environment. How can the system-administration profession maintain a positive development role under such pressures?

Alprin stated point-blank to his fellow systems professionals, "We need to grow up."

In the path to professional growth lie many obstacles and requirements. The many unique features of the job of systems administration make its taxonomy difficult. Established models rely heavily on predefined iterations to develop a series of procedures that can be followed by less skilled people, whereas system administrators are faced with a continuing stream of unique problems. We must therefore turn our attention to the core competencies and disciplines of system administration, and to the higher-level processes and standards that should be found in mature organizations. Inherent operational costs, technology turnover, and the pressure to succumb to a "just fix it" philosophy tends to override a total-solution implementation, and so the conflict grows.

Halprin pointed to essentials like shared mental models to enhance shared ideas, benchmarks and site evaluations to build organizational maturity, and establishing degrees and certifications in the system-administration field to propel personal development. Several organizational models, including ISACA COBIT, SEI CMM, and the PM-BOK, address similar fields and issues, and can be drawn upon.

Halprin identified 15 areas of systems administration disciplines: change management, problem management, production management, asset management, facilities management, network management, server management, software management, data management, data security, business continuity planning, performance management, process automation, capacity planning, and technology planning. We are all responsible for each of these areas, but we typically worry only about whichever one is hurting us most today. By taking a step back and quantifying these responsibilities, we can then take a proactive stance, planning improvements to each of these areas, and reaping the benefits in reduced stress and increased availability.

Halprin finished by describing the phases of the Taxonomy project, which is a long-term project with the goals of:

- building a reference framework to define systems administration
identifying the required core skills, knowledge, and disciplines so that people can be more effectively trained in the field

- capturing "industry best practices"

These goals are being ambitiously pursued in corresponding phases:

Phase One is the SA-BOK, which seeks to define the domains and subdomains of responsibility and the concepts, knowledge and tasks associated with each domain.

Phase Two is to define levels of maturity with respect to each of these domains, so that organizations can assess their maturity and plan improvement programs.

Phase Three is to capture industry best practices in each of the domains, to provide an industry-wide shared model of the best practices, contributed to and used by all.


Building Internet Data Centers
Jay Yu and Bryan McDonald, GNAC, Inc.

Summarized by Altaf Rupani

The speakers outlined the need to understand the service levels required for the business. For example:

- Uptime requirement – Is it an e-commerce datacenter, an internal business function datacenter, a development datacenter, etc.?

- What kind of data integrity/security is required?

- Network bandwidth requirements.

- Disaster-recovery requirements.

Once the requirements and service levels have been identified, the decision must be made whether to build and maintain the datacenter locally or outsource it. The decision should primarily be based on the resources available and costs associated with each option. Some points to consider:

- Power resources available both immediately and in the event of future expansion.

- Communications links.

- Air conditioning.

- Network cabling.

- Identifying a disaster recovery site for the location.

In addition, building a datacenter requires interaction with many people in the facilities world. Past experiences and recent ventures in the outsourcing world indicate that it would be wise to outsource the datacenter, unless certain business requirements make it mandatory to house the datacenter locally. In the latter case it should be noted that building datacenters is generally a time-consuming process, and that it’s important to organize finances well in advance.

Professional assistance in datacenter design could help address such questions as:

- What are the connectivity requirements?

- How much bandwidth is required?

- How much redundancy is required?

- Who will be the service providers?

- How much space should be allocated for the systems?

- What would be the ideal location (in the building) for the datacenter?

- How will you staff the datacenter?

Follow the N+1 rule: provide for N+1 quantities of resources when N are required.

Approaching a Petabyte
Hal Miller, University of Washington
Summarized by Josh Simon

Hal Miller, the immediate past president of SAGE, gave a talk on what it’s like to approach a petabyte of storage.

A petabyte is 1,024 terabytes, or approximately \(1.1 \times 10^{15}\) bytes. (For the curious, the next orders of magnitude are exabyte [EB] and zettabyte [ZB].) The trends are toward explosive growth but with bandwidth bottlenecks. The desire seems to be the equivalent of "dial tone" for IP networking, computing, and storage. This is all well and good, but how do we get there and make it work?

The problems a petabyte presents are many. Miller touched on some of them. 1PB is approximately 100,000 spindles on 18GB disks. Mirrored five-way, that’s 500,000 spindles and two copies offsite. Mirrors take 70,000 spindles, plus RAID drives, spares, and boot blocks, so we’re talking around 1,000,000 total spindles. At $1,000 per, that’s $1 billion just for the disk – this excludes the costs of servers, towers, networking, and so on. Where do you put these disks? What are the power and cooling requirements for them? How do you perform the backups? How recoverable are the backups? Where do you store the backups? How can you afford the storage, the facilities, the power, the cooling, the maintenance, the replacement of disks?

Who faces this problem? Oil companies (geophysical research), medical research...
Company mergers create an exercise in scaling. This is where you and the network enter. The finished product typically does not look like either company. A merger makes the new company larger and can make life easier. The first rule to adopt is to automate as much as possible. More important, though, is the use of standards. This one small rule will save time when you are in the process of integrating two networks and people are continuously asking questions. When you begin to merge the networks, you may lose some functionality. Prepare for this by making a flowchart of what and when things are supposed to happen, then update it as you go along.

Remember: Employees are forgiving. Customers are not.

Many times, as the network admin, you may not know about the merger until the public does. Your most critical activity during this time is dealing with senior management. There may be times when senior managers request something that is not feasible; simultaneously, you are asking management to lay out specific policies and guidelines about the network you are designing. Before you set up security and usage quotas, these policies need to be in place. Requesting these policies and guidelines as soon as possible, pending a merger, is in your best interest. Ideally, get as much information as possible regarding merging sites and departments. Which ones are going to be restructured? Stay away from “stupid network tricks”; either fix it or not — there is no middle ground. An interim fix will always come back and bite you in the end.

Here are some helpful hints that may save future confusion and make the transition smoother. Check for interoperability. For example, are you using ATM on one network and FDDI on the other? Focus on industry standards. Is your addressing global, or private? If you have both, which one will remain? Do not forget to leave room for your new company’s requirements, and for growth as well. Look for tools that can help you do these things, such as Cisco Works 2000.

State of the Art in Internet Measurement and Data Analysis: Topology, Workload, Performance, and Routing Statistics

kcclaffy, Cooperative Association for Internet Data Analysis

Summarized by Tony Katz

kcclaffy gave a very interesting talk about the difficulties in Internet measurements. She broke down assessment into four parts: topology, workload characterization, performance evaluation, and routing. According to her, one of the main problems is the lack of tools.

kcclaffy presented a multifunctional tool, called Skitter, which was created by Caida. She used Skitter to display large sections of the networks coming out of California. The 3D graphics were impressive, looking like a multicolored spider web. Skitter is also able to do dynamic discovery of routes, much as routing protocols do on a router. We were able to see a breakdown of the different types of protocols going across a given Internet line. For example, you might see 20,000 TCP packets travel between San Francisco and Los Angeles in a given time period. This is all well and good, but it means absolutely nothing without a point of comparison.

What you want is the ability to determine average usage from an ISP to their customers or from a main branch to a satellite branch. Once an average is calculated, it provides a starting point for future measurement. You can determine whether something went down locally or if it is a widespread problem. All of this sounds great, and I myself have wondered how to implement this for my own network. The big problem is that there are literally thousands of data streams coming into a single recording point. It
takes quite a while to decipher the pertinent information and transform it into something usable.

I can see the appeal, as kc does, in doing the research, simply because it is fascinating. As she pointed out, though, it takes years just to collect the data and then more years to understand and interpret it. Very few people are doing this type of work, which means it may take quite a while to have a complete measurement tool. It is a tedious job, but such measurements will allow us to find and fix problems on the Internet before they turn into a major crisis.

**PRACTICUM**

**Look Ma, No Hands! Coping with RSI**

Trey Harris, University of North Carolina, Chapel Hill

Summarized by Joe Morris

Trey spoke from real-life experience on the topic of repetitive stress injury (RSI). He’s heard the myths, seen all the doctors, and gone through lots of trial and error. Above all, he’s experienced a lot of pain that has had major impact on his ability to do his day-to-day job. He dispelled myths like “Can’t happen to me,” and “Can’t get any worse.” People must be aware of what can happen to them and take corrective measures as soon as possible.

For those who develop RSI, voice dictation is one option. Good packages are hard to find and are still susceptible to problems. Naturally Speaking is one such tool that Trey demonstrated for the audience. He could actually talk almost naturally and it kept up with him. However, when it came to writing Perl code, the results were disastrous. Passwords are a real problem too, unless you have your own private office. Any computer that takes on voice dictation requires lots of spare RAM for reasonable performance.

With respect to the RSI diagnosis, there is still much uncertainty among doctors. It’s important to get second and third opinions. Some doctors prematurely offer surgery — often not the best option, because it’s invasive. Carpal tunnel syndrome (CTS) is a subset of injuries under RSI. There are many other ways people can injure their fingers, arms, and wrists. The most common cause of RSI is using the smallest muscles to do repetitive tasks. To avoid RSI, do the opposite — use the largest muscles to do repetitive tasks. Small muscles were never intended to do the type of computer work we do today. Remember, RSI is not limited to computer work.

**Buzzword Bingo**

Panel moderated by Dan Klein, consultant

Summarized by Liza Weissler

Any opinions are those of the panel and not necessarily widely held.

The idea was to define various buzzwords and whether the average system administrator needs to worry about each right now, in three or nine months, farther out, or never. But in practice, the discussions centered upon definitions and not too much on the worry factor. (Errors in definitions below could be from the panel or be errors resulting from rapid note-taking by the summarizer.) Buzzwords discussed included:

- dot-com enabled – PHB (pointy-haired boss) — speak for “can you get us a Website?”
- brochureware – see #1.
- garage-band ISP – three guys with six Linux boxes.

USB, Firewire – Both are high-speed peripheral interfaces. USB is definitely here. Firewire has a small installed base and is more suited to consumer and specialized applications (e.g., downloading video); it may become dominant in the home in 3-9 months.

FM200 – a.k.a. "halon++", a less toxic, less caustic material for fire suppression, basically works by removing all available oxygen, so it can still kill you if you’re too far from the exit when the stuff is released. Very expensive — will be a while before it is widely adopted.

OODB – object-oriented databases. “Idea has a long way to go before becoming useful” (Greg Rose).

petabyte – three orders of magnitude more than a terabyte. Good for reading/writing, but not reproducing; problems with hack and dump abound. “If you’re not storing video or satellite feeds, not common.” Of course, people said a few years ago that nobody needed a terabyte either, so … start worrying. You may have a few years to do so.

J++/Visual J++ – PHB. Compatibility issues for programmers, but generally sysadmins don’t need to worry.

CRM – customer relationship management. More PHB stuff … although there was much discussion here. Refers to software to produce statistics that are useful for writing proposals/justifications/reports but may not necessarily do much for how you deal with your customers.

ERP – enterprise resource planning. “Accounting with human resources.”
Affects sysadmins in that software (e.g., Oracle Financials, Peoplesoft) needs to be installed/maintained.


SAN – storage-area network, multiple disks and tapes connected via fiber. Big stuff, especially in backups.

NAS – network-area storage – basically Network Appliance and other systems of their ilk.

fiber channel – an optical disk-connect technology. (FCAL = fiber channel arbitration loop; FCVI = fiber channel virtual interface.) More or less a replacement for SCSI. Differing from USB and Firewire in scope/scalability, it is basically a datacenter tool. (Note that channel is IBM-speak for bus.)

* M. 7 L. – mark-up languages.

Discussion focused on XML, the extensible mark-up language, becoming a standard for electronic data interchange (EDI) – think storing of data with formatting in a form that is easily readable like HTML. Some say it will replace HTML and perhaps PDF. Brief mention of SGML, Standardized General Markup Language. XML is a useful subset of SGML, as is HTML.

SCSI fast, wide, ultra, differential . . . “All colors of dead chickens” (Brent Chapman). SCSI (OK, we know that’s small computer system interface) was basically 8 bits at a given clock rate. “Fast” doubled the clock rate; “wide” moved to 16 bits. “Ultra” doubled the fast clock and moved to 32 bits. Note that as SCSI moved to fast, wide, ultra, the maximum cable length dropped. Differential includes more error-checking on the bus; incidentally, the maximum cable length is “nearly back out to where it should be.” Differential regular voltage is 12 volts, low-voltage is 3 volts (w/ 10,000 rpm drives). Thing to note here is that if you mix regular and low voltage, something will fry. Also in all this mess is the wonderful array of connectors, adapters, and compatibility of different types of devices.

SSA – serial storage architecture. Serial bus-based disk architecture from IBM. Cool stuff, but then so was Betamax.

SSL – secure socket layer – a Web thing. If a sysadmin doesn’t know about this already, something’s wrong. Chapman noted that one needs to plan certificates carefully with respect to server names, etc., since the certificates are not easy or quick to change (or cheap). Also that certificate use does not equal authentication, but is merely a useful addition (e.g., some sites combine certificate use with cookies). Too much to discuss here in three minutes.

PKI – public-key infrastructure. Important, and we have none. Attend a tutorial on cryptography/security and you’ll see why.

ASP – active server pages. Goes with PHP (pointy-haired protocol). Some dissenting opinions here . . . basically used to generate dynamic content and is “one step smarter than CGI.” Others called it “Visual Basic for the Web.” Brent Chapman’s summary was, “It’s not pretty, it’s not the way we would do it, but we don’t have to do it.”

ASP, take 2 – application service provider. Theory of multiple businesses sharing really expensive applications that would normally be installed on each business’s intranet. Not e-commerce per se. Stay tuned.

IAP – Internet application provider – refers to e-commerce sites sharing backend engines (e.g., eBay and others using someone else’s auction engine).

DSL – digital subscriber line and its variants. “ISDN on steroids – but not a dialup.” “How to take one lousy 50-year-old pair of copper wire and achieve reasonable network speed.” Seen by some as the true beginning of a paradigm shift; used with VPN (virtual private networking) will replace corporate dialup – it is already cheaper in most cases to go this route than to maintain a modem pool and pay long distance/800# phone bills. Estimated that 70% of U.S. residents live close enough to a switching station to get 384kb data rates. But one needs to check whether local ISPs have the capacity to support the number of subscribers at such rates – many tier 2 and 3 ISPs cannot. Some sites that rely on employees to use DSL/VPN for access may find that they don’t themselves have enough bandwidth – especially as employees start doing things that were previously considered impossible with it.

ISO9000 – a standard. “Do you have processes? Are they written down? Do you follow them?” – and that’s it. Quality of the processes doesn’t matter, just whether or not they are repeatable and consistent. If your process is to shoot your customers and you do it every time, you can be ISO9000 certified. (You’ll also be in jail.) Another opinion on this was that it meant “it is better to be up than fast; it is better to be reliable than good.” “6-Sigma” in this category, too.

A Couple of Web Servers, a Small Staff, Thousands of Users, and Millions of Web Pages . . . How We Manage (sort of)

Anne Salemme and Jag Patel, MIT

Summarized by Seann Herdejurgan

MIT developed its Web servers on the basis of the assumption, “If you build it, they will come.” The university currently has 600,000 Web pages on 1,000 Web servers. They decided to use existing resources. AFS is used extensively because of its scalability and security using Kerberos. They also use Apache-SSL, Fast-CGI, and Java servlets.
In 1994–1997, MIT’s Web environment had:
- email forms (using generic CGI)
- image-map support
- restricted access (MITnet only)
- search engine (Harvest)
- campus map
- certificate-based authentication
- Web-publishing training

In 1998–1999, MIT’s Web environment had:
- discussion groups
- search engine (Ultraseek)
- Web-usage statistics
- server-side includes (SSI)
- restricted access for individuals or groups
- secure file transfer
- recommended WYSIWYG editor (Dreamweaver)

Upcoming items in MIT’s Web environment:
- Internet-wide events calendar
- secure credit-card transactions
- module-based publishing (XML)
- database-generated content
- better “indexing” of Web content
- next-generation portals

Lessons learned:
- ability to scale for growth
- AFS + 1 Web server is not enough
- special-purpose Web servers required
- what users see (on the Internet) is what they want
- influence people through guidelines and relationships
  
  <http://web.mit.edu/guidelines.html>

---

**Budgeting for SysAdmins**  
Adam Moskowitz, LION bioscience Research, Inc.; and Gregory H. Hamm, GPC USA, Inc.  
Summarized by Liza Weissler

This was an excellent practical discussion covering not only elements of a budget in detail but Moskowitz and Hamm’s tips with respect to the purpose and people of budgets and the budgeting process.

The purpose of a budget, essentially, is to serve as a very detailed planning tool, describing what you want to do next year and why. It is a way to get funding – but not necessarily the only way. It is an instrument to foster discussions about what your company/department is doing and hopes to accomplish, as well as a means to find out whether everyone is “on the same page.” It also allows you to be able to answer questions from other employees and departments so that they in turn can plan their own budgets.

The scary part of developing a budget, especially one’s first time through the process, is coming up with the numbers. Moskowitz and Hamm counsel that you’re not expected to know all the numbers, but simply with whom to talk to get them. Users, your boss, “the bean counters” (accounting and purchasing), and “the suits” (department heads, VPs, directors . . . maybe even your CIO, CFO, CEO for elements of the business plan, and if the company structure allows it and it’s not a bad idea for your environment) – all of these people can be extremely useful to you. Too often the bean counters and suits are seen as adversaries or, worse, stupid. In reality, they’re neither – they simply have different jobs from yours. If you play to their strengths and take advantage of what they can offer you, you’ll be happier and more likely to get what you and your company need – e.g., purchasing could help you out with the numbers on just how much toner you ordered last year, while different levels of management can tell you about hiring plans and company directions.

The last general guideline Moskowitz gave was to plan to have your budget cut, because they always are. If your budget is structured into reasonable categories with not entirely obvious slash points, you can whittle down the budget to your own liking, as opposed to having it done to you.


---

**Inexpensive Firewalls**  
Simon Cooper, SGI  
Summarized by Jordan Schwartz

As the use of the Internet grows, so does the need for inexpensive firewalls to protect the security of internal systems.

Simon Cooper described the needs and uses for an inexpensive firewall and how to build and administer the systems.

Inexpensive firewalls are dedicated systems using available or low-cost hardware and free or low-cost software. It was pointed out that these are not no-cost systems: a substantial time investment is needed, and these firewalls do not provide maximum security or the highest reliability available. Appropriate areas of deployment for inexpensive firewalls are departmental networks, small businesses, homes, and personal domains.

The talk covered various aspects of building a firewall, including determining firewall needs; hardware; OS and software selection; OS hardening; kernel
defenses; and filtering software information and examples, services, build tips, and experiences. The administration section discussed securing remote-administration connections and maintaining system integrity.

**Ethics**

Lee Damon, Qualcomm; and Rob Kolstad, SANS Institute

Summarized by Eric Jones

This session's stated objective was to try "to avoid making egregious first-order mistakes and move on to second-order mistakes."

The speakers began with an attempt to define what ethics are and a discussion of why they might be important. The issue of ethics for system administrators has taken on a higher profile in recent years because of the increasing amount of data – including sensitive data – stored online.

They then went on to discuss some of SAGE's six canons of ethics and how some of them may not be entirely realistic.

Finally, the speakers led an audience discussion of several scenarios that a system administrator might face. They ranged from questions of when to inform a manager about employees misusing company resources to what your response should be to a request by a manager to search for child pornography in an ISP customer's home directory.

These scenarios weren't intended to show us what the "correct" response in a given situation is, but to show us how reasonable people with similar goals will, nonetheless, think differently on ethical matters.

**NETA/LISA-NT/Security Highlights**

David Williamson, GNAC, Inc.; Gerald W. Carter, Auburn University; Greg Rose, Qualcomm Australia

Summarized by Carolyn Hennings

A review of three recent conferences replaced a session that had to be cancelled. The program chairs from each of the conferences spoke briefly about the highlights of each.

**University Issues Panel**

Moderated by Jon Finke, Rensselaer Polytechnic Institute

Summarized by Carolyn Hennings

William Annis of the University of Wisconsin described how they have managed the growth in one group within the university. He related how they had developed a detailed planning document for centralizing and standardizing the systems and the implementation of configuration to ensure consistency across the environment.

David Brumley of Stanford University discussed how his organization deals with computer security and incident response. The goal of the security office was to provide a secure, fast, and reliable network without firewalls; provide technical assistance with technical implementations; and provide a point of contact for incident reporting, handling, and follow-through.

Robyn Landers from the University of Waterloo discussed their solution for residence-hall networking. She described how the process for students to get connected and how the university implemented an automated system of limiting the amount of network traffic allowed to individual students. This "rate-limiting" has prevented network overload and has encouraged students to share resources.

Kathy Penn from the University of Maryland described their backup procedures and policies. She emphasized the importance of documented procedures for doing backups and restores. She suggested that overview information, as well as cookbook-type instructions, is necessary. Documenting the policies regarding frequency of backups and the creation of archival copies, how long the archives are kept, and what you don't back up are necessary. Additionally, provide information on how to request restores and how long it should take to do the restore. A policy for who can request restores of information is critical.

**WORKSHOPS**

**Advanced Topics Workshop**

Adam S. Moskowitz, Moderator and Chair

Summarized by Josh Simon

Once again the Advanced Topics Workshop was wonderfully hosted and moderated by Adam Moskowitz. The 30 or so of us each discussed our environments and mentioned some of the problems we were seeing. We then looked at some of the common themes, such as hiring and growth (virtually everybody present had open positions), scaling (especially at the enterprise level), some tools, and areas where we felt there had to be improvement (such as system administrators being able to speak the language of business in order to justify expenses).

The afternoon session of the workshop included some predictions for what we thought would be coming in the next year (wireless LANs, load-balancing hardware, LDAP, the lack of adoption on a widespread basis of Windows 2000, the lack of adoption on a widespread basis of IPv6, an increased demand for H.323 proxies for video conferencing, at least one major DNS outage lasting 24 hours, no new top-level DNS domains like .web and .biz, and no major problems when the century rolls over). Lest you think that we're omniscient -- or that we even consider that as a possibility -- we also
looked at our success rates from the previous four workshops. We were right about some things, dead wrong on others, and one to four years ahead of our time on still others. So take these predictions with a grain (or bushel) of salt.

Finally, we wrapped up the workshop with a discussion of some problems we’re facing (a VMS-to-UNIX transition in one place, the administration of customers’ router passwords in another, and so on), with possible solutions bandied about. We also briefly touched on some interesting or cool stuff we had done in the past year. A lot of us were doing Y2K remediation and documentation.

**GIGA LISA Workshop**

Joel Avery, Nortel Networks, chair

Summarized by Doug Freyburger

At the workshop, we broke into four groups, each of which discussed two topics. I covered NT-UNIX integration and internal firewalls. The group also discussed the “most daunting problem.”

**NT-UNIX Integration**

Cooperation isn’t good enough. Integration is about sharing as much as it makes sense to share between the two operating-system environments.

Password sharing solutions: Sites have started storing password data in various types of databases and have written utilities to reencrypt the passwords for each system. Used were a custom Oracle database, a Radius/Informix utility normally used to control modem dial-ins and routers, and a hacked Kerberos.

File sharing solutions: NetApps and Auspesen support both file systems directly. Smaller sites get along with Samba on UNIX and Dave on Macs.

Patch maintenance: Active versus passive maintenance schedules; no integrated solution, though.

Dataless clients: With file sharing, the dataless model makes excellent sense in both worlds.

No solution presented: Unified user-profile storage on NT to match the user-account-based dot files on UNIX. Since user-configuration information was moved into a database in NT, how can it be moved from machine to machine as a user roams, and how can other users be prevented from accessing a user’s email?

**Internal Firewalls**

As the Internet reached 250K nodes, people started making firewalls. Now that large companies have more than 250K nodes inside their networks, internal firewalls are being installed.

They are for resource constraints. The firewall is to protect the group that installs it, so they are local responsibilities. This got called “directional protection.”

Use NAT to redirect by service.

Interesting tidbit: One in 700 employees is actively hostile to his or her employer. I wonder who came up with this and if it is true.

SUNROC versus NTRPC is a knotty problem. Netmeeting is a bear.

With multiple firewalls, asymmetric routing becomes a serious problem because IP packets do not record their path, so routers can choose between redundant paths. With firewalls, this becomes a serious problem.

**Most Daunting Problem**

Someone had a pair of datacenters to build the next day. He would run AIX on the main servers, and he wasn’t an AIX wizard yet. The group spent an hour asking questions and making recommendations. He took notes the whole time.

---

**SAGE Community Meeting**

Summarized by Carolyn Hennings

Peg Schafer opened the SAGE Community Meeting with a number of announcements, which were followed by a question-and-answer session.

Current activities include preparations for the LISA 2000 conference, December 3–8 next year in New Orleans and the LISA-NT conference scheduled for July 30–August 2 in Seattle.

The board was pleased to announce that SAGE was recently able to purchase the sage.org domain name.

SAGE-WISE has formed, representing Wales, Ireland, Scotland, and England.

The topic of “understanding what we do” has been the focus of a number of efforts including the SAGE Taxonomy working group, the salary survey, the occupational analysis survey being conducted by the SAGE Certification working group, and the results from the “Day in the Life” survey. There is increasing activity in how system administrators are educated as well as in helping match
mentors with individuals who want to improve their system administration skills.

The question-and-answer session seemed to focus primarily on the need for more publicity and marketing for SAGE that conveys the value it offers to system administrators and to businesses.

**SAGE Mentoring Project BOF**
Summarized by Carolyn Hennings

The primary purposes of this BOF, which was led by Michael Ewan, were to identify individuals who were interested in serving as mentors and to provide the opportunity for individuals who would like to be mentored to step forward. The discussion also centered on the process of matching up individuals with mentors and how SAGE can help with the logistics of the mentoring relationship.

**SAGE Taxonomy BOF**
Summarized by Carolyn Hennings

Geoff Halprin opened the BOF by asking a number of questions of the audience. The discussion centered on how different organizations have attempted to standardize the work that system administrators do. The group discussed the work proposed by Geoff in his first draft of a "Body of Knowledge" for systems administration and how it can be used. It was suggested that a method of evaluating an organization's competence in each of the "Body of Knowledge" areas would be beneficial.

**LISA99 TERMINAL ROOM**
By Dave Bianchi

The name Terminal Room has not been accurate for a long time; it should probably be renamed to "Internet Connection Room."

The terminal room, managed by Lynda McGinley and staffed by volunteers, was actually two rooms: one room with 30 PCs running Linux and a separate room with 40 Ethernet connections for laptops and the Axis Webcam. In addition, 10 modems were set up to allow access to the network from a Sheraton hotel room by dialing a four-digit extension; four of these modems were accessible from other hotels.

The Internet connection was a framed T1 provided by Earthlink. A wireless point-to-point connection from the Convention Center to the hotel was set up for the conference and paid for by GNAC. The networking equipment consisted of Cabletron and NetGear hubs.

As an experiment, 120 Lucent Technologies Wavelan 802.11 Turbo Bronze wireless PCMCIA cards (in both 2 and 11 Mb speeds) were available for checkout with a credit card; they were all checked out in the first couple of hours. Five wireless bridges (or Access Points) were provided to support the Wavelan cards, including one in the hotel bar.

The PCs were rented from Houlihans. Terminal room volunteers Dave Putz and Connie Sieh provided a set of six custom CDs and diskettes for Linux installation that made the installation and configuration go very quickly and smoothly. The PCs were installed with a minimum of software, but did include Netscape and ssh. Dave also provided a Tcl program that monitored the use of the PCs and enabled him to gather usage statistics at the same time. Dave's usage graphs indicate that a majority of the PCs were busy most of the time that the room was open.

USENIX conference attendees have come to depend on the terminal room at large conferences. Because of this, USENIX is looking at the feasibility of providing Internet connectivity at every conference and workshop.
Using Production Grammars in Software Testing
Emin Gün Sirer and Brian N. Bershad, University of Washington

Sirer's work is motivated by the desire to test and debug a Java virtual machine. Known testing techniques include formal methods; manual code analysis; manual test-case generation; and a technique favored perhaps too often, "release the system and wait." The goal for this project was to provide automated test generation via some sort of specialized scripting language. Sirer's system, a language called lava, enables users to use a production grammar to produce test cases. The syntax of lava is very similar to regular YACC-style parsing grammars, except that the grammar is used in reverse, to generate test inputs instead of parsing them. Lava has provisions for specifying limits and weights on productions, as well as a means of generating context-sensitive output. The output of the tool is a set of test cases that may vary widely. These test cases may be used to detect "gross violations of type safety," compute time complexity, and verify the correctness of code transformations.

One problem Sirer had to address was the "oracle problem": because the test cases may be complex, it is unclear what the correct output should be, compared to the output the system computes. While comparative methods are sometimes feasible, as with testing a Java virtual machine, the oracle problem remains for systems that have no other implementation available for comparison. To address this, Sirer has developed an auxiliary tool that generates lambda expressions that reflect the computation desired; as rules in lava are traversed, the lambda expressions are composed. The result is that when a test case is completed, the production rules have also generated a lambda expression in Scheme that reflects the analogous computation. Thus, the output of a JVM (for example) can be compared to the output of a Scheme interpreter when the corresponding lambda expression is evaluated.

This project has two major contributions: on one hand, test generation is made simple with lava, with very high test coverage and control over the types of tests; the other contribution is the integration of the test generation with a description of the expected behavior.

Jargons for Domain Engineering
Lloyd H. Nakatani, Mark A. Ardis, Robert G. Olsen, and Paul M. Pontrelli, Bell Laboratories, Lucent Technologies

This project is a study of the use of jargons, a family of DSLs that share a common syntax and customizable interpreter, in real-world domain-engineering problems. A model expressed in a jargon can be easily transformed into a multiplicity of related products — for instance, C code and its documentation. Because making jargons does not entail designing their syntax or writing their interpreter from scratch, they are very easy to make. In fact, they are so easy to make that domain experts with no language-design experience can easily make jargons for their own use (therefore called "Do It Yourself" jargons). To test this claim, Nakatani et al. had teams of domain experts working within the FAST domain-engineering process make their own jargons. Each team had access to a consultant who was a jargon expert.

Each team successfully made the jargons they needed, confirming that jargons are easily made by domain experts who are not experts in language design and implementation.

Jargons share all the benefits of conventional DSLs — including domain-specific expressiveness, high-level abstraction, and evolution — while avoiding the pitfalls of DSLs such as high language-development and maintenance costs. In addition, because jargons share a common syntax and interpreter, they are easily composed as long as care is taken to avoid keyword conflicts. Composability means that a complex problem can be broken up into simpler subproblems, each subproblem modeled in its own jargon, and the models merged to express a model for the entire problem. Jargons manage thereby to avoid the Tower of Babel syndrome that would otherwise be a consequence of the proliferation of DSLs.

Slicing Spreadsheets: An Integrated Methodology for Spreadsheet Testing and Debugging
James Reichwein, Gregg Rothermel, and Margaret Burnett, Oregon State University

The goal of this project was to provide an easy way for spreadsheet users to debug spreadsheets. In doing so, the authors had to take into account that not only are spreadsheets modelled and offer immediate feedback, but the spreadsheet "developers" are unlikely to understand testing and debugging theory and methodologies. Therefore, they wanted a system to lead the user through debugging and fault localization that would be easy to use. A previous system allowed users to identify correct output of cells when inputs were entered, and it marked untested cells in red. This paper adds the idea of fault localization, which uses a backward dynamic slice to trace user-determined errors back to cells in the spreadsheet that could be causing the
packet requesting multicast, instead of one packet per request. There are many different levels of network programming: a program could be installed at the routers; the packet could carry the program; or the program could be a "switchlet" – a combination whereby the packet provides input instructions and the switchlet, resident on the router, interprets those instructions. One application of switchlets is the Queue Management Switchlet (Hicks et al., 1999), which implements Flow-Based Adaptive Routing (FBAR).

This research is being carried out in the SwitchWare project at the University of Pennsylvania. SwitchWare uses a three-layer architecture: the top active packet layer, a service layer, and an OS layer. While the service layer is written in a GPL, the active packet layer uses a DSL called PLAN. PLAN is a scripting language for composing active-network services. PLAN is declarative and functional but has no looping mechanism. The routers are protected from malicious code by typechecking, and the network is protected by resource limits on the code. The typechecking is Anytime Typechecking®: types can be checked either statically or dynamically.

Considerable security problems need to be addressed in implementing active networks. Already there exist security problems in just researching active-network technology. SRI runs an active network testbed with machines hosted at different sites across the country. Each site wants to determine its own security policy, but they must all agree on a security protocol. For authentication, a public-key system is preferable but requires a public-key infrastructure (PKI) to establish trust, handle certificates, and express and check authorizations. There are several DSLs for policies; one, the Query Certificate Manager (QCM), achieves general policies, transparent policy distribution, diverse distribution strategies, local autonomy, and a formal model.

Thus, in this system, policy-directed certificate retrieval is possible. However, sometimes it becomes necessary to revoke a certificate. For this, Certificate Revocation Lists (CRLs) must be maintained and distributed. Fox and Lamaccia (1998) have studied CRLs.

In this talk, we saw how DSLs can be used in multiple ways in emerging technologies. Further questions are:

- When is a DSL architecture advantageous?
- Is there a high overhead in introducing a DSL?
- Is there general support for DSL design, development, testing, and deployment?

INVITED TALK

Language Technology for Performance and Security, or, Making Life Better, Not Just Easier

Peter Lee, Carnegie Mellon University and Codilla Systems Incorporated

This research is motivated by a focus on safety-critical systems, which are those systems in which the cost of failure is "unacceptably high" – for example, those used by the space program or for airplane guidance systems. These systems are everywhere, used all the time. Such systems typically are required never to crash; always meet deadlines; be reconfigurable without the need to shut down; and be secure, trustworthy, lightweight, extensible, and adaptable. With time, safety-critical systems won’t disappear; instead, they will be more integrated with daily activities. Programming-language technology will provide the technology for safety, because the same characteristics that make languages easy to program in can also make them easy to reason about.

The idea of proof-carrying code is to provide easy access to remote resources while maintaining invariants, protecting the key, and matching the allowed behav-
ior (according to the resource protector) with what could happen while running a piece of client code. The idea is that the host publishes rules about what is allowed and a set of verification conditions. The verification conditions can be used to produce a proof checker and a proof generator. Anyone who wants to use the resource uses the verification conditions to produce a proof generator, then feeds the program in to get a proof. The program then carries the proof (say, as part of its header) to the resource. If the untrusted client has used anything but the correct proof generator or verification conditions, the proof checker at the host will detect it and reject the program.

DSLs can provide safety policies specific to the domain. Once we have and can manipulate safety policies, we can use them to generate certificates. Consider a certifying compiler, in which the compiler "explains" why the target code produces preserves the safety properties of the source. Then, by certifying the compiler and the source, we can conclude that the target is safe. However, instead of certifying the compiler, which would be the equivalent of doing a formal proof of correctness, the strategy instead is to have the compiler provide annotations in the output to allow the theorem prover to reconstruct the translation process. This is what Lee's implementation of a certifying compiler does. The compiler generates optimized code from Java source code. The compiler outputs both code with annotations and a proof, which can be verified by a theorem-prover on the host side, given the hints in the annotations. The compiler is mostly off-the-shelf; the theorem-prover is hidden, and the binaries are in standard format.

Thus, the use of DSLs allows for reasoning to be done about code that is optimized and in standard format, thereby making it easier and more efficient to verify and run untrusted code.

2nd USENIX Symposium on Internet Technologies & Systems (USITS '99)

BOULDER, COLORADO
October 11–14, 1999

Keynote Address
Summarized by Steven Bird

E-Commerce—An Optimistic View
Udi Manber, Yahoo! Inc.

Dr. Udi Manber posed the rhetorical question, "Will e-commerce change the world?" Using the meteoric rags-to-riches success that Yahoo! embodies, Manber described the transformation from bricks and mortar (BM) that e-commerce enables. The end result is still quite debatable, but Manber believes that the winners will share several traits. One of these traits will be the ability to provide a one-stop shop that will satisfy the bulk of visitors' needs. The winners will also realize that these needs are not strictly material, and that the direct translation of a BM operation to an online presence will prove to be tragically shortsighted. Manber believes that a successful online strategy must include an abundance of means by which the users can establish community. He described the development of email, chat rooms, clubs, and message boards as all being critical elements in the success of Yahoo! In addition, an intuitive user interface (20,000 help page hits out of a total of 200 million accesses) shows Yahoo!'s success here will be crucial.

Manber finished with a few speculations on what the future might hold. One prediction is that online advertisements will evolve from their present annoyance status to a source of useful connections and resources. His contention is that the ad strategy of today is equivalent to fishing with dynamite. A second and even more novel idea is a "Universal ID" that everything carries and that a person could "click" on to buy. The product would then be mailed to the purchaser and the owner of the item that had been clicked would receive a commission.

Udi Manber, winner of the 1999 Annual Software Tools Users Group award, is chief scientist at Yahoo! Before joining Yahoo! in 1998, he was a professor of computer science at the University of Arizona. He has written more than 50 technical articles (three of which won USENIX Best Paper awards), co-developed Agrep, Glimpse, Harvest, and the Search Broker, and wrote a popular textbook on design of algorithms.

Shared Caching
Summarized by Steven Bird

Scalable Web Caching of Frequently Updated Objects Using Reliable Multicast
Dan Li and David R. Cheriton, Stanford University

Dan Li presented a method to address the issue posed by frequently updating objects in a Web cache. To avoid the repeated unicast that this would require, she proposes the use of MMO (multicast invalidation followed by multicast delivery using OTERS) to avoid the negation of the benefits provided by multicast. This is achieved by grouping objects into volumes, each of which maps to one IP multicast group. The benefit from reliable multicast, with volumes of appropriate size, were shown to outweigh the cost of delivering extraneous data. Li demonstrated the scalability of this approach using trace-driven simulations. The bandwidth saving vis-à-vis conventional approaches increased significantly as the audience size grew. Li presented a strong argument that MMO provides efficient bandwidth utilization and service scalability. This should help to make strong Web-cache consistency for dynamic objects practical.
Hierarchical Cache Consistency in a WAN
Jian Yin, Lorenzo Alvisi, Mike Dahlin, and Calvin Lin, University of Texas at Austin

Jian Yin described a means of improving cache consistency using a flexible, efficient, and scalable tool. Using two primitive mechanisms, split and join, to manage consistency hierarchies and to address the fault-tolerance performance challenges of consistency hierarchies, Yin was able to demonstrate this as a promising configuration for providing strong consistency in a WAN in a two-level consistency hierarchy. His arguments were supported with the use of synthetic workload and trace-based simulation. One particularly promising configuration for the provision of strong consistency on a WAN is a two-level consistency hierarchy in which servers and proxies work to maintain consistency for the data cache at the client.

Organization-Based Analysis of Web-Object Sharing and Caching
Alec Wolman, Geoff Voelker, Nitin Sharma, Neal Cardwell, Molly Brown, Tashana Landray, Denise Pinnel, Anna Karlin, and Henry Levy, University of Washington

Alec Wolman’s group examined the sharing of Web documents from an organizational point of view. In light of the fact that performance-enhancing mechanisms on the Web primarily exploit repeated requests to Web documents by multiple clients, organization-based caching can possibly offer efficiencies. Wolman et al. evaluated the patterns of document-sharing access (1) among clients within single organizations and (2) among clients across different organizations. To perform the study, Wolman used the University of Washington as a model of a diverse collection of organizations. Within the university, he traced all external Web requests and responses, anonymizing the data but preserving organizational-membership information. Analysis of both inter- and intra-organization document sharing allowed them to test whether organizational membership was significant.

The results demonstrated a surprising lack of sharing within the organizations they delineated (~2% over random). In addition, there was an overarching commonality between the organizations studied in that there were 850 top servers handling over 50% of the requests analyzed. Also, lots of content is uncachable. In the question-and-answer period following the presentation, it was speculated that the university is perhaps more homogeneous than was initially appreciated.

Applications
Summarized by Steven Bird

The Ninja Jukebox
Ian Goldberg, Steven D. Gribble, David Wagner, and Eric A. Brewer, University of California at Berkeley

When presented with the appalling waste of resources represented by the numerous unused CD-ROM drivers at the UC Berkeley computer lab, the Ninja group sprang into action to create a realtime streaming directory of audio from these sites. After they wrote the porting software for streaming delivery, the MP3 revolution arrived and the Ninja Jukebox idea was spawned. The goal was to develop a service that allowed a community of users to build a distributed, collaborative music repository to deliver digital music to Internet clients. The project’s success led to an abundance of music, 17 days’ worth, and the associated copyright and filtering obstacles this presents. Interface development efforts were then focused on the development of simple collaborative filtering based on users’ song preferences and ownership. The Jukebox, implemented in Java, was designed to allow rapid service evolution and reconfiguration, simplicity of participation, and extensibility. DJ software was developed to profile the users’ preferences in a portable and selectively accessible manner. Presenter Steven Gribble concluded with the assertion that the careful use of a distributed component architecture enabled rapid prototyping of the service. He also felt that use of carefully designed, strongly typed interfaces enabled the smooth evolution of the service from a simple prototype to a more complex, mature system. Future modifications include the possibility of using digital cash to provide general access. He is optimistic that the newer version of Java will operate more efficiently and meet the bandwidth demands more gracefully.

Cha-Cha: A System for Organizing Intranet Search Results
Michael Chen, Marti Hearst, Jason Hong, and James Lin, University of California at Berkeley

A standard search engine retrieves Web pages that fall within a diverse range of information contexts but presents these results uniformly in a ranked list.

Michael Chen presented a novel search engine that is based on the premise that intranets contain information associated with the internal workings of an organization. This engine, named “Cha-Cha,” organizes Web search results in a manner that reflects the underlying structure of the intranet. This “outline” is created by first recording the shortest paths in hyperlinks from root pages to every page within the Web intranet. After the user issues a query, these shortest paths are dynamically combined to form a hierarchical outline of the context in which the search results occur. Pilot studies and user surveys suggest that some users find this structure more helpful than the standard display for intranet search. Currently a quarter of a million pages are indexed. More information is available at <http://cha-cha.berkeley.edu/>.
A Document-based Framework for Internet Application Control

Todd D. Hodes and Randy H. Katz, University of California at Berkeley

Todd Hodes presented a novel document-based framework for manipulating the components that comprise distributed Internet applications. In the framework, XML documents are used to describe both server-side functionality and the mapping between a client's applications and the servers it accesses. This system contrasts with explicitly context-aware application designs, whereby location information must be explicitly manipulated by the application to effect change.

Instead, Hodes and Katz interposed a middleware layer between client applications and services so that invocations between the two can be transparently rehydrated, and have found this useful for a subset of application domains, including one example domain of “remote control” of local resources (e.g., lights, stereo components, etc.). Hodes went on to illustrate how the framework allows for (1) remapping of a portion of an existing user interface to a new service, (2) viewing of arbitrary subsets and combinations of the available functionality, and (3) mixing dynamically generated user interfaces with existing user interfaces. The use of a document-based framework in addition to a conventional object-oriented programming language provides a number of key features. One of the most useful is that it exposes the mappings between programs/UI and the objects to which they refer, thereby providing a standard location for manipulation of this indirection.

Techniques
Summarized by Steven Bird

Sting: A TCP-based Network Measurement Tool
Stefan Savage, University of Washington

The tongue-in-cheek theme of Stefan Savage's presentation was that TCP represents an “opportunity” rather than a transport protocol. The novelty of this perspective led to some creative developments and garnered Savage the Best Student Paper award. Savage developed Sting, a tool to quantify one-way packet loss. This feature is not available in ping. Sting is able to accurately measure the packet-loss rate on both the forward and reverse paths between a pair of hosts. This achievement is accomplished by leveraging the behavior of TCP.

In TCP one knows the number of packets sent and the number received. This is enough for ping to work, but determining one-way packet loss requires more information. First, you need to know how many packets were received at the other end. TCP has to know this, given that it is a reliable protocol. The second required piece of information is the number of ACKs that were sent to you. ACK parity requires that for every data packet received there is an ACK sent. Savage proposed a two-phase algorithm. Phase one is the data-seed phase and involves sending n in-sequence TCP data packets and counting the number of ACKs received. These are the probes of the network loss rate. The second phase is the hole-filling phase, which discovers which of the packets sent in phase one were lost. A new packet is sent that has a sequence number one greater than the last packet sent in the data-seed phase. If the target responds with an ACK for this packet, then no packets have been lost. If any were lost there will be a “hole” in the sequence space, and the target will respond with an acknowledgment indicating exactly where the hole is. This is filled with each subsequent retransmission, and a lost packet is recorded.

Using fast retransmit, which imposes upon the receiver the responsibility of sending an ACK for every packet that is out of sequence, can optimize this. Skipping the first packet will force an ACK for every packet sent. A second tweak involves the transmission of packets that differ by only one byte and thereby optimize the use of the receiver buffer. Firewalls and load balancers can become problematic when they send unwanted resets that would disrupt this metric, so they were kept at bay by advertising a zero-sized receive buffer that prevented them from sending. The findings Savage reported indicate that the forward packet loss rate is much less than the reverse packet loss. He felt that this asymmetry is due to the large differential in data transmission in the reverse versus the forward direction.

JPEG Compression Metric as a Quality-Aware Image Transcoding
Surendar Chandra and Carla Schlatter Ellis, Duke University

Transcoding is a generic term for any transformation process that converts a multimedia object from one form to another. The goal of this work was to increase the effectiveness of the transcoding technique applied to Internet data access. With the use of JPEG images, the efficacy of transcoding was assessed to arrive at a “quality-aware transcoding” metric that explicitly trades off image information with reductions in object size and/or clarity.

Surendar Chandra presented techniques to quantify the quality-versus-size trade-off characteristics for transcoding JPEG images. He analyzed the characteristics of images available in typical Web sites and explored how to perform informed transcoding using JPEG compression. The effects of this transcoding on image
storage size and image information quality were then demonstrated. He also presented ways of predicting the computational cost as well as potential space benefits achieved by the transcoding. He felt these results will be useful in any system that uses transcoding to reduce access latencies, increase effective storage space, and reduce access costs.

Proxy Implementation
Summarized by Rick Casey

Secondary Storage Management for Web Proxies
Evangelos P. Markatos, Manolis G.H. Katevenis, Dionisis Pnevmatikatos, and Michail Flouris, ICS-FORTH

Disk I/O is a known factor in limiting Web-server performance, contributing as much as 30% to total hit response time. A primary reason is that each URL in Web caches is stored in a separate file. An obvious method of improving system performance would be to reduce the overhead associated with file maintenance. The authors proposed a storage-management method, called BUDDY, for storing several URLs per file. By identifying URLs of similar size (“buddies”) and storing them in the same file, disk I/O is reduced. Although BUDDY reduces file-management overhead, it makes no effort to reduce disk-head movement induced by write operations to various “buddy” files. To improve write throughput, the authors proposed STREAM, which, in addition to storing all URLs in a single file, writes URLs contiguously in this single file, reducing the number of disk-head movements (much as log-structured filesystems do). A third suggestion was to improve read throughput by clustering read operations together (LAZY-READS). Finally, to restore the locality present in a client request stream, the authors proposed to use Locality Buffers (LAZY-READS-LOC), which attempt to store URLs requested contiguously by a given client in contiguous file locations.

The results were tested with a combination of trace-driven simulations and experimental evaluations. Traces from DEC were used to compare Squid’s file management method, BUDDY, STREAM, LAZY-READS, and LAZY-READS-LOC. The conclusion was that disk-management overhead can be reduced by as much as a factor of 25 overall by using these algorithms. Because disk bandwidth will improve faster than disk latency, the authors believe such algorithms will be an increasingly valuable means of improving Web-server performance.


Compression Proxy Server: Design and Implementation
Chi-Hung Chi, Jing Deng, and Yan-Hong Lim, National University of Singapore

Automatic and optimized data compression of Web objects was examined as a means of improving server performance and reducing use of network bandwidth. The authors acknowledged that with faster, higher-capacity systems, where the compression-to-transfer-time ratio is higher, there is less need for compression. Still, there are many portions of the Internet where better compression on a proxy server would help overall network latency. Compression can be either explicit (decompressed at the client) or implicit (compressed and decompressed at the server). The problem of automatic compression is constrained by the HTTP protocol, the many file types of Web objects, and their varying sizes. Therefore, accurate, rapid classification of these objects is needed to select the best compression algorithms. Compression can be performed on an entire file, a data block (a single Web object), a data stream, or not at all. The benefit of compression must be considered with respect to the added overhead.

Implementing the compression methodology encountered three design issues: encoding of compression messages, memory allocation, and choice of a data structure. The Squid proxy server was modified to test the methodologies, using a trace collected at a Singapore college over a year. Experimental results revealed the distribution of Web objects (file types) by total bytes to be: GIF image 33%, JPEG image 12%, text 31%, and octet-stream — binaries of MPEG, MIDI, or other applications — 24%. File sizes within these categories were recorded. Results of compression effectiveness were “highly encouraging.” Bandwidth saving was measured by file type and size; whole file compression was the highest, at 37%. Overall, about 30% of bandwidth was saved in this experiment, and compression/decompression was less than 1% of Web access latency (even on an “outdated” proxy server). The authors conclude that such Web-server compression is worthwhile and should be considered as a bandwidth-saving mechanism, particularly since it could cooperate with other techniques.

On the Performance of TCP Splicing for URL-Aware Redirection
Ariel Cohen, Sampath Ranganathan, and Hamilton Slyle, Bell Laboratories, Lucent Technologies

This research examined a software switch that supports URL-aware redirection of HTTP traffic, known as “content-smart switching,” using TCP splicing. The purpose of the splicing is to improve the performance of the switch. Ariel Cohen pointed out that while several vendors are beginning to announce such switches, little or no implementation or performance information is available. It was also noted that a hardware-based URL-aware switch has been reported by IBM researchers.

The switching functionality was implemented using a loadable module in the
Linux kernel. A user-level proxy accepts connections from clients and decides which server will receive incoming requests. The proxy then removes itself from the data path by requesting the kernel to splice the TCP connection between the client and the proxy with the connection between the proxy and the server. The loadable module is actually two components: sp-mod, which monitors the connection, and NEPP (Network Element for Programmable Packet Injection), which performs low-level header modifications. This worked with the Linux ipchains firewall to filter packets.

Performance results were tested using the WebWatch HTTP generator on five clients. The experiments were run for three minutes each with a concurrency setting of 75 in a thread pool of size 30. The servers and clients were fast PCs (400-550MHz) connected over Fast Ethernet on Lucent’s intranet. Performance impacts were observed with and without TCP splicing. In all cases TCP splicing resulted in a significant performance improvement. At the average Web object size of 10KB, there was a 58% increase in connections and a 38% decrease in CPU utilization. Performance gains were, of course, more striking for larger objects.

Prefetching
Summarized by Rick Casey

Prefetching Hyperlinks
Dan Duchamp, AT&T Labs - Research

This paper, which developed a new method of prefetching Web pages into the client cache, won the Best Paper award for the conference. Dan Duchamp did an excellent job of presenting the highlights of his research without bogging the audience down in the details, and he honestly revealed where he came up short.

Duchamp began with two basic premises: (1) the next URL to be requested by a client is likely to be one embedded as a hyperlink in one of the last few pages requested by that client, and (2) past access patterns of a large population of clients are likely to be relevant to a particular client.

The basic method is:
1. The client sends to a page’s server a record (called a “usage report”) detailing which of that page’s hyperlinks it referenced.
2. The server aggregates such information from many clients.
3. When responding to a GET, the server attaches a summary (called a “usage profile”) of past usage reports for that page.
4. On the basis of a page’s usage profile, the client decides whether to prefetch any of its hyperlinks. Usage reports and profiles are passed via a new HTTP extension header.

The paper presents a brief but comprehensive summary (not included in the presentation) of related research in the extensive area of prefetching, which falls into three categories: software systems; algorithms, simulations and/or prototypes; and methods establishing bounds. The features distinguishing Duchamp’s method from previous work are: it has been implemented; how information on access patterns is shared by the server over clients occurs in near-realtime; is client-configurable; many previously uncacheable pages can be prefetched; both client and server can cap operations to limit impact on overhead and bandwidth; and it operates as an HTTP extension.

The overall results were very positive: client latency improved greatly (slightly over 50%), and less of the cache was wasted (about 60% of prefetched pages were eventually used).

Both client and server modifications can be implemented as proxies, eliminating the need to alter browsers or Web servers; however, there are disadvantages to a client proxy.

Other “gotchas” were: time-dependent accesses; objects set with zero expiration time; inaccessible HTML; sabotage in (using prefetching to overload the network); privacy concerns; and the fact that usage patterns are beginning to have commercial value, raising payment issues.

The server-side implementation is a proxy based on W3C’s HTTPd. Two client-side implementations exist: a modification of the Mozilla browser from Netscape and a proxy based on HTTPd. Performance was evaluated for prefetch accuracy, client latency, network overhead, program space overhead, and program time overhead.

Mining Longest Repeating Subsequences to Predict World Wide Web Surfing
Jim Pitkow and Peter Pirolli, Xerox PARC

This was somewhat theoretical examination of the topic of predicting user “surfing paths,” the sequence of Web pages that a given user will visit within a given Web site. The goal of the research was to develop a model that had limited complexity while retaining high predictive accuracy. The utility of predicting users’ surfing has applications in improved searching, better recommendations for related sites, latency reduction through prefetching, and Web site design.

Various Markov models were compared to assess their ability in pattern extraction and pattern matching. Two techniques were motivated: longest repeating subsequences (LRS) and weighted specificity. LRS is a means of identifying the most information-rich subsequences in navigation log files. This data was then
Markov models to compute conditional probabilities for sequential transitions; that is, if a user is on a page, what is the probability of the user clicking any of the available links? The models had to be compact enough to be of practical use. These models were about 130KB in size, small enough to reside in each thread of a Web server. Using a single data set from Xerox, the models were able to predict the correct sequence 27–31% of the time. The speaker cautioned that these results are tentative and need to be corroborated by future work. Finally, he presented a picture of “information scent,” a visualization of user paths within a given Web site, with examples of “good info scent” and “bad info scent.” While the algorithm producing the visualization was not discussed, it was presented as an alternative model to determine what information to prefetch to users.

Architectures
Summarized by Rick Casey

Active Names: Flexible Location and Transport of Wide-Area Resources
Amin Vahdat, Duke University; Michael Dahlin, University of Texas at Austin; Thomas Anderson and Amit Aggarwal, University of Washington

This paper described a new framework for organizing and delivering distributed services over the Internet, called Active Names. The research is motivated by the fact that Internet services are increasingly distributed across various machines, but the limitations imposed by DNS (Domain Name Service) have resulted in many confusing suggestions for improving it. Active Names is meant to be a general design solution that encompasses much previous research on extending DNS. Its main points are (1) it provides flexible end-to-end naming abstraction or WAN services and (2) it provides a framework for composing customizations provided by both clients and servers. The benefits would be increased network performance (reduced client latency) and a standard, unified approach to operating networked services.

The key concepts in the design are active names, namespaces, delegation, and after-methods. Each active name identifies a name and a namespace in which that name should be interpreted, and each such namespace is embodied by a Namespace Program. Unlike URLs—which map to a specific IP address and specify where a service will be run—namespace programs are location-independent. Namespace programs accept incoming data, determine the next namespace where output will be sent, and construct an after-methods list to send with the output. The interface to namespace programs facilitates composability (the ability of one namespace to call other namespaces) in two ways:

(1) through delegation, where one namespace passes a name to a sub-namespace, and
(2) through after-methods that specify a chain of Active Name services remaining to be run to finish name resolution for a request. These namespace programs execute within a resolver virtual machine that provides security and limits resource use.

The authors have demonstrated a fully functional core of the system and have built several useful applications. In response to questions, the authors explained that the system provides basic facilities on which applications enforce security and provide end-to-end fault tolerance, and that providing higher-level support for security and fault tolerance would be useful future work. The test system was built at Texas, Duke, and Washington using Java, and results indicate that Active Names can significantly reduce client latency in distributed services, in one case providing a fivefold reduction.

The kinds of questions raised by distributed processing have created an area of active research, including Active Services, Active Networks, Intentional Name System, and Transaction Processing monitors. However, the authors believe each of these has limitations that Active Names overcomes.


Person-level Routing in the Mobile People Architecture
Mema Roussopoulos, Petros Maniatis, Edward Swierk, Kevin Lai, Guido Appenzeller, and Mary Baker, Stanford University

A platform for truly mobile, person-centric communication was the topic of this presentation. The goals are simple: maintain person-to-person reachability, protect privacy, and be deployable within the existing infrastructure. The primary focus is a merging of Internet and telephony communications. The Mobile Person Architecture (MPA) basically depends on routing all communications to a personal proxy, which acts like a router between the person it serves and any incoming communication. The proxy is a trusted software daemon under the control of the user, who tells it where he or she will be and how to respond to any communication. The personal proxy cooperates with a tracking agent, a rules engine, and a dispatcher. How these components were implemented in Java was described. The dispatcher is responsible for content conversion, which ensures that content arrives in a suitable form depending on where a person is at the time. The proxy is designed to be as easy to install and operate as any Web service, to help ensure its success (though no market testing of this has been done). Related research projects were described: cellular phone projects in Japan, the Iceberg project, the TOPS architecture, the SPIN project by the
Canadian National Research Council, and transcoding proxies. All these have shortcomings when compared to MPA, which has an API that allows future extensions to incorporate any new communication service. It is interesting to note that the research was supported by a group of Japanese organizations, including NTT Mobile Communications Network, Inc., a phone company. More information is available at <http://mosquitenet.stanford.edu/>.

A User’s and Programmer’s View of the New JavaScript Security Model
Vindu Anupam, David M. Krzot, and Alain Mayer, Bell Laboratories, Lucent Technologies

This was a straightforward examination of the security weaknesses of JavaScript and of how the author and his team implemented a new security model using the public-domain Mozilla source code. The improvements they made are likely to have been implemented in Navigator 5.0, which was scheduled for release in late 1999.

JavaScript, of course, is the general-purpose scripting language invented at Netscape that runs within a browser. Meant for manipulating objects within the browser environment, it offers an adversarial programmer the means of attacking the client system. The presentation focused on the features of their new security model. This is based on two basic components: access control, which regulates what data a script can access; and trust management, which regulates how trust is established and terminated. The security policy is configurable to a great extent by the end user, from very strict to relaxed, and offers access to low-level settings or acceptance of predefined policies. This contrasts sharply with the current situation, in which a user can choose only to turn JavaScript on or off. A security policy can also be set at the organization level and installed via a service integration. For each type of security violation, the user can define what action should be taken — whether to stop, continue, or deny the requested access.

The programmer’s view of the new model was described, with code snippets illustrating how security policy is implemented in trust management. The utility of this feature was shown in an e-commerce example which requires automated cooperation between business sites. The methodical process by which the authors tested their new security layer was described. The addition of the document.ACL attribute, a key innovation in the new model, is currently before the W3C as a proposed standard. The new security model has been offered to the Mozilla open-source-development community for scrutiny before its implementation by Netscape.


Works-in-Progress Reports
Summarized by Steven Bird

PerDiS: Persistent Distributed Store
Marc Shapiro, INRIA Rocquencourt and Microsoft Research Cambridge

Marc Shapiro described a persistent distributed store or Internet caching for cooperative engineering. It exports the abstraction of a shared memory across the Internet. Shapiro believes that PerDiS is particularly simple to use, claiming that large centralized programs (including a 400,000-line CAD tool) have been ported with relative ease. Application programs allocate arbitrary objects inside clusters (i.e., files), and objects refer to one another with native pointers. Between an application program and the shared store, writes are buffered in a transactional log. This allows engineers to work on shared designs without interference. PerDiS has two major modes of operation. In a LAN, the store is kept coherent, whereas sharing over a WAN follows a check-in/check-out mode.

PerDiS is open source.
<http://www.perdis.esprit.ie.org/>

PaperFinder
Athanassios Papathanasiou
<papathan@cs.rochester.edu>

Scientists always need to stay informed about developments in their fields. The increasing number of printed and electronic papers makes it increasingly difficult for a single person to keep up with all the relevant information that she or he might be interested in. There are simply too many sources of (potentially) useful information, many more than any single person has the time to track. This project developed PaperFinder, a tool that continually searches digital libraries of scientific publications, filters the relevant papers, and delivers them to interested scientists through a friendly user interface.

USEwebNET
Athanassios Papathanasiou
<papathan@cs.rochester.edu>

When a user wants to find information about a specific topic, he or she sends a query to a search engine (e.g., AltaVista) which replies with several URLs. Every time the user wants to find new information about the same topic, AltaVista returns the same URLs, flooding the user with unnecessary information.

USEwebNET is designed to relieve users from the long waits and information flood associated with the traditional search model. Specifically, USEwebNET is a network tool with a user-friendly interface designed to retrieve documents about selected subjects (or updated versions of selected documents) from the Web and present them to the user along with information about them, following the user’s preferences.
Improving Web Searching Performance Using Community-based Filtering
Liddy Shriver
<shriver@research.bell-labs.com>

Members of a community with shared interests search for similar things on the Web. Shriver and her group are employing community-based filtering to use the results of successful past searches by members of a community to guide new searches. They analyzed logs from a Web proxy server and found that searches done by members of a community are often repeated. Her group developed a prototype search assistant, SearchLight, which augments existing search engines by offering hints based on these past searches. Their analysis shows that SearchLight will offer hints 20% of the time and in some cases will decrease search time significantly.

Distributed Object Consistency Protocol
John Dilley
<jad@pimlico.hpl.hp.com>

The Distributed Object Consistency Protocol provides for stronger object consistency in Web proxy cache servers than HTTP can currently deliver. Dilley’s simulation of the protocol showed that it can deliver content to users with lower response time while consuming fewer origin server and network resources than caches using the traditional Alex consistency protocol.

The Flash Web Server
Vivek Pai
<vivek@cs.rice.edu>

The Flash Web server is a high-performance Web server developed using a novel concurrency architecture. Flash combines the high performance of single-process event-driven servers on cached workloads with the performance of multiprocess and multithreaded servers on disk-bound workloads. Pai has found that the Flash Web server is easily portable, since it achieves these results using facilities available in all modern operating systems.

Webcard: A Java Card Web Server
Peter Honeyman
<honey@cit.umich.edu>

Webcard is a TCP/IP stack and Web server written in Java that runs on a Schlumberger Cyberflex Access smartcard 16KB eeprom and 1.2KB of RAM. ISO 7816 Smartcard and Java Card 2.0 compliant, Webcard handles one connection at a time and has minimal state maintenance (filename and TCP port are it) and three states (listen, established, and finword 1). It uses no options, no retransmissions, no checksums — who needs them when you use the sequence number as a file offset? — and no returns. Webcard only supports IP with a 250 byte mtu. Try it yourself at <http://smartys.cit.umich.edu/>.

Defeating TCP Congestion Control in Three Easy Steps
Stefan Savage
<savage@cs.washington.edu>

How to coerce a remote Web server to send at any rate you choose:

1. ACK Division. In TCP the sending point increases its congestion window by one segment with each successive ACK it receives. Action: Send multiple ACKs regardless of the packets received, up to 1,500 per packet, and watch your window grow quite rapidly.

2. Duplicate ACK Spoofing. TCP recovery after three duplicate ACKs at a sender involves retransmitting the packet and increasing the congestion window by one packet per duplicate ACK received. Action: Send a stream of duplicate ACKs, and for every additional duplicate ACK sent the congestion window is grown by one, which allows you to control the size of your window and thus the rate of transmission.

3. Optimistic ACKing. This involves sending an ACK for a packet 35 msec early, with the consequence being that the sender will send subsequent packets early.

The above strategies are all implemented in the TCP Daytona that Savage put together using fewer than 75 lines of code on Linux. Those who are interested can see the full article, "TCP Congestion Control with a Misbehaving Receiver" in the October ’99 ACM Computer Communications Review.

Automating Usability Assessment for Information-centric Web Sites
Martí Hearst
<hears@sims.berkeley.edu>

Hearst is investigating the use of simulation of user behavior relative to the usability of a given Web page’s content and structure. This is anticipated to permit designers to choose among design alternatives before implementation. Modeling tools used have included trace-driven discrete event modeling and Monte Carlo simulation.

Appliance Data Servers
Armando Fox
<fox@cs.stanford.edu>

Fox is exploring how to connect input-centric consumer devices (digital cameras, handheld scanners, etc.) to the Internet service infrastructure, while maintaining a “no-futz, point and squirt” user experience. These devices are intended to allow users to inject data into the infrastructure – for example, a digital camera that uploads images to a Web page or automatically emails them to mom. Key obstacles to achieving this goal are finding a way to attach metadata to the input data and modifying the default action of your device. Currently Fox is developing an info-daemon that embodies a protocol gateway and verb extractor that behaves in a protocol- and device-specific fashion. It will figure out
how to extract a verb that accompanies the data from one of these devices and canonicalize the data and the verb into a yet-to-be-determined format. The rest of the infrastructure would use this within a fixed piece of software (with the exception that you may want to plug in additional modules later on). An example was shown, a digital camera that has an IR port annotated with the command “Send this to mom.” The first thing the info-daemon does is to look up the verb and convert to a canonical form with a browser edit option. Then the command is entered into the service infrastructure. The current prototype accepts PalmPilot IR and HP JetSend.

Caching Policies
Summarized by Rick Casey

Using Full Reference History for Efficient Document Replacement in Web Caches
Hyokyung Bahn, Seoul National University; Sam H. Noh, Hong-Ik University; Sang Lyul Min and Kern Koh, Seoul National University

This project focused on a better algorithm for directing the updating of documents in Web caches. It seeks to improve on previous algorithms by having the ability to optimize on any performance measure. The algorithm, Least Unified Value (LUV), uses the full reference history of documents to estimate the probability of being referenced.

Web caches have been much studied by the research community, but the authors believe the LUV algorithm is best to use in a Web cache replacement policy. According to their research, it offers the best overall effectiveness, performance, and robustness. The algorithm is basically a cost value computed for each cache document. The value is a weighted average of the document’s reference potential multiplied by its weight, where weight is a function of cost and size. Reference potential is the likelihood of re-reference, a function of past references. Cost can be considered in several ways, depending on the performance measure in which you are interested.

Since the LUV algorithm makes use of full reference history for each cache document, it might seem a burden to implement. But the speaker offered a proof that collapses the computation of the cost component. Implemented in a heap, this reduced computation of LUV to time \( O(\log(2)n) \), where \( n \) is the number of cached documents. Experiments were done using traces from NLANR and DEC, though filtering was done on UDP and CGI requests and on requests larger than the size of the cache. Performance was compared to the nine cache algorithms for hit rate, byte hit rate, and delay-savings ratio. In most cases, the authors conclude that LUV outperformed all other algorithms irrespective of cache size. This algorithm considered only in-cache documents; future research will consider a perfect-history LUV, which includes replaced objects, for possible performance improvements.

Providing Dynamic and Customizable Caching Policies
J. Fritz Barnes and Raju Pandey, University of California at Davis

This paper investigated an infrastructure that allows customization of Web caching policies at the client, proxy, and/or server. The research is motivated by the fact that current Web-caching policies act on all Web objects uniformly; given the diversity of Web objects, cache performance could be enhanced by customization of the cache policies.

The research presented an object-oriented analysis of cache objects, specifying how customization policies can be applied; namely, in prefetch, routing, placement, coherency, removal, and “miscellaneous,” the last category intended to address any protocol extensions. Implementation of the policies is accomplished by CacheL, a domain-specific language based on cache events developed by the authors. Two caching systems were built to evaluate the effectiveness of these ideas: DavisSim, an event-based cache simulator based on the Wisconsin Cache Simulator, and PoliSquid, an extension of the popular Squid Web cache. Analysis of the performance focused on whether caches benefit from customization and what overhead it demands. The analysis was broken down by client-customized, cache-customized, or server-customized policies. Results indicated that implementation was feasible and advantageous. Overhead was moderate, estimated at an 8.5% increase in latency without optimization. The possibility of cache-policy customization was demonstrated, but further evidence of performance improvements awaits future work.


Exploiting Result Equivalence in Caching Dynamic Web Content
Ben Smith, Anurag Acharya, Tao Yang, and Huican Zhu, University of California at Santa Barbara

This work presented a proposal for a new protocol for enhancing Web caching and a prototype for implementing such a protocol. The basic idea is to identify query requests that have essentially equivalent or similar results and to service these subsequent requests from the cache. The usefulness of this is most apparent in image maps and queries conditionally qualified over some range.

The protocol, called Dynamic Content Caching (DCCP), classifies Web client requests according to three types of locality: identical, equivalent, or partially equivalent requests. Currently, Web cache hits can only identify requests with the same URL (identical request). DCCP goes further by allowing identification of requests with identical content (equivalent), or content that can serve as a tem-
porary placeholder for a request (partially equivalent). This is accomplished by an extension mechanism in HTTP 1.1 for cache control directives.

Examples were shown using image maps, a weather service that uses ZIP codes to qualify queries, and a news service applicable to geographic regions. These are equivalent or partially equivalent requests that can be exploited by DCPC, implemented using the Swala cooperative Web cache. Evaluations were based on cache hit ratio and generated traffic using two real traces and one synthetic trace. Results were promising. For a map retrieval trace with three levels of error tolerance in matching, hit ratios can reach over 60% at a 10% error tolerance. The authors acknowledge that DCPC has a memory overhead cost, but this can be controlled by imposing a bound. They encountered difficulty in implementing efficient search when using complicated string-matching, and also did not address POST-based queries, which they plan to study in future work.

More information is available at <http://www.cs.ucs.edu/research/swala>.

Server Implementation
Summarized by Rick Casey

Efficient Support for Content-based Routing in Web Server Clusters
Chu-Sing Yang and Mon-Yen Luo, National Sun Yat-Sen University

This paper explored the advantages of a clustered Web server that uses a new content-aware request-routing. It offers a survey of the clustered-server approach to servicing high-traffic Web sites and suggests improvements in routing requests to improve overall performance via content-aware processing.

Using a clustered-server approach at high-traffic Web sites, where the initial node directs requests to specialized servers, is an advantageous approach undisputed in the literature. The basic methods for accomplishing this were summarized: client-side, DNS-based, TCP connection routing, and HTTP redirection. The issues ignored by these approaches that the authors identified were session integrity, load balancing, differential services, and content deployment.

The research focused on the difficulties that the TCP protocol imposes on a server-directed solution, primarily the difficulty of migrating the established connection. In their design, the dispatcher decides how to route on the basis of the data structures for a cluster table, a mapping table, and a URL table. The dispatcher maintains an awareness of the TCP connections, releasing them when necessary. Since the overhead of a new connection is prohibitive, the dispatcher conveys packets to the backend servers, modifying the packet IP and TCP headers before forwarding.

The design was implemented in a loadable module for the Linux kernel. WebBench was used to evaluate performance on a heterogenous collection of back-end servers. Compared to a “content-blind” server cluster, the content-aware cluster had greater throughput, averaging about 20MB/sec more after 16 client connections. The authors acknowledged that the extra overhead, limited scalability, and the dispatcher as a single point of failure were drawbacks to their approach. The advantages were higher performance, better routing decisions, and general content-aware intelligence that might be useful in future configurations.

Rapid Reverse DNS Lookups for Web Servers
William LeFebvre, Group Sys Consulting; Ken Craig, CNN Internet Technologies

This paper presented a mechanism that supplants the usual way Web servers find which clients are contacting them using the Domain Name Service (DNS). Identifying a client is important for targeted dynamic content (e.g., Web-page ads) for advertising-supported Web sites. Conventional DNS lookups are prohibitively slow for busy Web servers; thus rapid reverse DNS lookups could have great significance to advertisers. The author’s team implemented this design for CNN, one of the most heavily used news sites on the Internet, where it has exceeded expectations since its implementation in March 1999.

Surprisingly little other work has been in this area. The design depends heavily on the multithreaded capabilities of Netscape’s enterprise server API. Basically, a Rapid DNS server is placed between the (modified) Web server and the conventional DNS server. Quick answers are provided to its client Web servers on the front end (about 2 milliseconds or better, on average), using a bucket hash keyed on IP address. The Rapid DNS server makes periodic queries to the true DNS server off the back end. The connection from front to back is through a fixed-sized stack called a “leaky bucket” because of its LIFO design, which drops an increasing backload of requests off the end. Negative caching, whereby a cache entry is maintained for unknown domain names, significantly improved cache hit rates.

A trio of Rapid DNS servers was used to support the CNN Web farm of about 60 Web servers. Performance results were impressive: even with over 250 client connections, servers sustained queries in excess of 400 operations per second. Future research will investigate different queuing and caching policies. The code for this project was developed for CNN and remains proprietary.
Connection Scheduling in Web Servers
Mark E. Crovella and Robert Frangioso, Boston University; Mor Harchol-Balter, Carnegie Mellon University

This research approached a common goal — improving Web-server performance — through a novel approach: applying scheduling theory to Web-server design and operating-system architecture. Task scheduling is always a consideration for the CPU, the disk I/O, and the network interface, normally all under control of the operating system. Using a simple idea from scheduling theory, the paper proposed placing scheduling more under application control, using a new algorithm based on the idea of “shortest-connection-first.” Basically, this means that for tasks where the size is known, it is best to schedule shorter tasks earlier; in practice, the metric for this translates into Shortest Remaining Processing Time (SRPT). The detailed analyses of this idea suggested that a four- to five-fold increase in throughput could be achieved without penalizing longer tasks.

Implementing the idea was problematic because task scheduling is traditionally under the control of the operating sys-

were that SRPT scheduling matters very significantly. Overall throughput was improved in all tests, and not at the expense of longer tasks. This somewhat surprising result is explained by the average file-size distribution of Web requests being heavily skewed toward smaller file sizes.

These promising results encourage further research, which will be directed toward servers allowing more precise scheduling control and dynamically adjustable job priorities.
Greetings

... and welcome to the first installment of what will be a regular series of columns in ;login:. I'm a security nerd, so I will primarily be sticking to that topic, with occasional forays into other issues. Over the course of the last 14 years or so, I have evolved through the various stages of the techie life cycle: system administrator, programmer, network manager, security manager, project leader, product manager, consultant, chief technology officer, and chief executive officer. Why is that important? Because it's taught me that perspective is everything, and your ability to put things in perspective is somehow (I can't quantify) related to the variety and depth of experiences in your personal and professional life. So I find myself sometimes in the unique position of trashing positions I once held or waving off complex technology issues as "mere details." All I can ask is that you bear with me; I promise I won't write anything that hasn't got some underlying point that I've thought out.

Security Today

Let's talk about the state of enterprise security today. That's a huge topic, of course, but it's going to be important to us all for a long time to come. Obviously, the Internet is a big chunk of that problem, but security concerns will eventually push their way into virtually anything that's doing computing. Assuming that hasn't already happened. From the "30,000-foot view" there really isn't a huge difference between a company's intranet and the Internet. About the only difference I usually see is that security is ignored on the intranet and paid attention to at the Internet connection. This results in a sense of security, since there's a separation between the things we control and the things we don't control. That's very important to management since it breaks things neatly into things that are our problem and things that are not our problem. When you see people trying to break big problems into smaller, more tractable problems, that's a sure sign that they are trying to manage complexity. Managing complexity is a difficult problem, so let's examine a couple of ways in which it applies to security.

I sometimes have the privilege of addressing rooms full of technical people who are interested in security. This is a great chance to conduct quick unscientific polls. One of my favorites has to do with browsers. You ask a roomful of technical people to raise their hands (and keep them up) if they have had the following happen:

1. A browser crash in the last hour (one or two hands go up).
2. A browser crash in the last 24 hours (5% of the hands go up).
3. A browser crash in the last week (60% of the hands go up).
4. A browser crash in the last month (the rest of the hands go up).
5. A browser crash in the last year (general laughter).

The next question is:

Given that you've proven to your satisfaction that you're using unreliable technology, how many of you engage in e-commerce or online stock trading using a browser?

This question is usually followed by nervous laughter, and a brave few admit it and raise their hands. I don't know about you, but I buy lots of stuff online. The reason is...
Perhaps the browser in Dreamcast will never have a security flaw. If so, it'll be the first secure browser ever.

Managing Complexity

Indeed, the browser itself is a tool for managing the complexity of the Internet, circa 1992. Goodness, those ftp commands had such a wretched interface! Let's give the average user a way of getting on the Internet by just pointing and clicking, and all the details of HTTP, ftp, telnet, etc., shall be hidden from them by an overlay of graphics that "do what I mean" at the click of a mouse button. Newer-generation browsers are so complex they take on some of the properties of operating systems. They dynamically load programs, search paths for plug-ins as UNIX shells do, call other executables, maintain their own file systems and caches, and so on. Indeed, there is one browser maker that claims its browser is an operating system or is such an important part of the operating system that the two can no longer be separated. Yeah, whatever. But what are the implications for security? I'll tell you: they're bad. Any time a system tries to do things for the user and hides the details, there's a good opportunity for a bad guy to dupe the system. Putting a browser-style user interface on something is a great way of reducing the apparent complexity of a system – but it replaces that with the complexity of the browser.

Firewalls are devices for managing complexity at a network level. When I built my first firewall, it had to service only ftp, telnet, SMTP, NNTP, and DNS. Today's firewalls are expected to operate in a service environment that is hugely more complicated. The number of services deployed across the typical firewall has gone up dramatically, while our comprehension of them and their implications has gone down. How many people know intimately all the features and hooks of some of the cool new Web apps? Probably the people who coded them and nobody else. In many cases, the security of the protocol is that the protocol is unpublished and changes constantly from version to version. The firewall, in its highest-level view, is a thin layer of incompatibility that is deliberately placed between two areas that are incomprehensibly complex. The firewall breaks all the stuff we don't understand and trust and, we hope, protects us by doing so. What's scary to me is that more and more applications "understand" firewalls – by which the designers mean they tunnel holes through them so that the mysterious
undocumented protocols will still work. Firewalls have now become just another piece of complexity to kludge around.

Intrusion-detection systems are devices for detecting deviation from expected complexity. If I expect my network to contain a mix of applications of a certain type and it starts seeing traffic of another type, it means my network has gotten more complex without my permission, and that usually spells trouble. Vulnerability scanners are tools for assessing whether our complex systems are in expected configurations. System managers no longer have time to understand the jillions of things that could be wrong with their systems and have to rely on a piece of software to put a nice interface on it all by summarizing what needs to be fixed and why. At every level where we simplify the complex, we lose some information – and we lose our ability to understand what is going on behind the scenes. I suspect that if you asked some kids if they knew there was an IP stack in their Dreamcast, many of them would wonder if it plugged into the expansion slot or the memory-card port. I believe we need to be building the next generation of systems so that they are accessible to the nontechnical, but the more layers of paint and duct tape we put around the underpinnings, the harder it is to see and understand the implications of all the cracks underneath.

Appliances are merely the next trend in managing complexity. But will we eventually have too many appliances? I was looking at an ad the other day for a plug-in fileserver. Put the 10BaseT connector in, power it on, and it’s an instant fileserver. How is it secured? Well, presumably, since it’s an appliance, it’s remotely controllable. I’m not saying it’s a bad product, but it’s designed to appeal to the nontechnical, and as a direct result of being nontechnical they probably won’t even think about the security questions. We’re seeing the same thing with the new generation of home high-speed Internet connections. Instead of intermittently dialing up to the Internet, a whole new population leave their machines connected 24 hours a day, where they can be quickly scanned and pillaged at leisure. It’s not a new security problem – plain old dialup has the same issues – but the customer base is increasingly less sophisticated, and the service providers are reluctant even to breathe a word about security, because then they’d have to educate their customers. Worse, they’d have to educate their customers about a problem with the service they propose to sell. That’s bad for business.

**Today’s Toy Is Tomorrow’s Business Tool**

Complex and poorly understood technologies are being rushed into customers’ hands at an ever-increasing rate. One of the things that fascinates me about the Internet is how quickly an application can gain a massive installed base. For example, one of the online messaging systems in widespread use was signing up 100,000+ users a day. That was before it became really popular. It was a toy application and had essentially zero security built into it. It had no good authentication or encryption, and eventually it came to support file transfers and remote URL sharing. Now it’s probably still a toy application, but I bet that within two years people will be using it to negotiate mergers and acquisitions, hold product strategy meetings, or issue stock buy/sell orders. It’ll still have essentially zero security built into it. How can we, as security people, get application designers to build security into version 1.0 of their software? Would the world be a better place if we could put security gurus in a time machine and send them back to whisper in Tim Berners-Lee’s ear, “Put security in it, this is gonna be big”? I don’t know how Tim would answer, but the usual application developer’s answer would have to be, “That’s too complex, I haven’t got time, I’m on a tight release schedule.” In other words, the way the application designers manage the complexity of security is by leaving it for later.
Okay, I've rambled enough. Next column, we'll talk about less nebulous high-level stuff and try to pick on something more technical.

I'd also like to run a contest, in which the winner will get a cool T-shirt. The winner's entry will be in the next column. Your mission, should you choose to accept it, is to write a computer-security haiku. Email entries to <mjr@safe.net> with a subject line reading "haiku." I'll notify the winner before the next issue.

building a Windows NT bastion host

This article presents a checklist for converting a default Windows NT installation to a bastion host. It makes little or no attempt to explain or discuss the features it implements. Therefore I suggest that you first read all the Knowledge Base articles I've listed and the other referenced documents. If there is something you don't understand after having read those articles, do not continue. Read them again or look for additional assistance.

What Is a Bastion Host?

A bastion host is a computer system that is exposed to attack and may be a critical component in a network security system. Special attention must be paid to these highly fortified hosts, during both initial construction and ongoing operation. Bastion hosts can include:

- firewall gateways
- Web servers
- ftp servers
- name servers (DNS)
- mail hubs
- victim hosts (sacrificial lambs)

The American Heritage Dictionary defines a bastion as:

1. A projecting part of a rampart or other fortification. 2. A well-fortified position or area. 3. Something regarded as a defensive stronghold.

Marcus Ranum is generally credited with applying the term bastion to hosts that are exposed to attack, and its use is common in the firewall community. Ranum says:

Bastions are the highly fortified parts of a medieval castle; points that overlook critical areas of defense, usually having stronger walls, room for extra troops, and the occasional useful tub of boiling hot oil for discouraging attackers. A bastion host is a system identified by the firewall administrator as a critical strong point in the net-
work's security. Generally, bastion hosts will have some degree of extra attention paid to their security, may undergo regular audits, and may have modified software.[1]

Bastion hosts are not general-purpose computing resources. They differ in both their purpose and their specific configuration. A victim host may permit network logins so users can run untrusted services, while a firewall gateway may permit logins only at the system console. The process of configuring or constructing a bastion host is often referred to as hardening. The effectiveness of a specific bastion-host configuration can usually be judged by answering the following questions:
- How does the bastion host protect itself from attack?
- How does the bastion host protect the network behind it from attack?

Extreme caution should be exercised when installing new software on bastion hosts. Very few software products have been designed and tested to run on these exposed systems. See Chapman and Zwicky[2] for a thorough treatment of bastion hosts.

Install NT

Start with a clean system. The machine should not be attached to a public network while you are doing the installation/configuration. If you have to have a network connection, make sure it’s an isolated, trusted network segment. Do not have any other operating systems installed on your bastion host. Install Windows NT 4.00 US-ENGLISH. Use only NTFS. If you’re installing NT Server, make it a "stand-alone" member server. This server will not be able to participate in a domain environment. Do not install IIS 2.0. If you want to run IIS, install it from the NT option pack.

As for network protocols and services, install only TCP/IP and do not install additional network services.

Consider removing everything except WordPad in Add/Remove Programs -> Windows NT Setup.

Install Software

Install any third-party software. This might be a Web server such as IIS 4.0. To install IIS 4.0 you have to have SP3 or above already on the system. This doesn’t change the fact that you have to reinstall SP5 afterward.

(Re-)Install the Latest Service Pack

Install the latest service pack for Windows NT 4.00. (At the time of writing, this is Service Pack 5.) If you choose to make a backup of old files during the SP installation, be sure to remove the old files afterward. We do not want to leave the possibly vulnerable binaries on the system.

Install Available Hotfixes

Install all available hotfixes, which are available from ftp://ftp.microsoft.com/bussys/winnt/winnt-public/files/us/v400/. These should include only Windows NT OS fixes, not any application-specific fixes.

Remove Unused Network Services

Remove all unused services with the Network application in the Control Panel. This should leave you with a configuration like the one shown in Figure 1.

Only the RPC configuration for the port mapper (RpcSs) is left. IIS will not start without it.
Note that when you remove the Workstation service, you will get a message every time you start the Network application in Control Panel: “Windows NT Networking is not installed. Do you want to install it now?” Always answer NO to this question.

Another caveat is that User Manager for Domains (usrmgr.exe) stops working when the Workstation service is not running. Replace it with User Manager (usrmgr.exe) from NT Workstation.

**Disable NETBIOS**
By unbinding the WINS Client in the Network application from all adapters, we get rid of all listeners on the NETBIOS ports: Network -> Bindings -> All protocols -> WINS Client -> Disable.

Also disable the WINS Client driver in Control Panel -> Devices -> WINS Client -> Disable.

**Configure TCP/IP Filters**
Configure TCP/IP security by specifying the ports that are allowed inbound (TCP or UDP) on each network adapter. This is done in Network application -> Protocols -> TCP/IP -> Advanced -> Enable Security -> Configure.

*Skip this step if you are going to install other packet-filtering software on this host later on.*

**Example: Web Server**
The configuration shown in Figure 2 allows only connections to tcp/80.

No UDP is accepted. ICMP cannot be blocked.

**Disable Unused Services**
Everything should be disabled except the following (excluding any applications we want running on the system, of course):

- EventLog
- NT LM Security Support Provider
- Protected Storage
- Remote Procedure Call (RPC) Service

The processes that should be running are:

- smss.exe Session Manager
- csrss.exe Client Server Subsystem
- winlogon.exe The logon process
- services.exe The main service handler process
- pstores.exe Protected storage
- lsass.exe Local Security Authority
- rpcss.exe The RPC end-point mapper
- explorer.exe The Explorer GUI
- loadwce.exe Explorer-related
- nddeagnt.exe Explorer-related

**Encrypt the System Accounts Database**
Run the syskey.exe utility (with the key on disk option). This will provide basic protection against password-cracking tools like L0pht Crack (<http://www.l0pht.com/>).
Apply Policies and ACLs
Run the Microsoft Security Configuration Editor (SCE) in command-line mode. The command-line version of this tool is included in the hpt.zip archive, available at my Web site (<http://people.hp.se/tnor/>). This SCE is a part of the Service Pack 4 CD. Our configuration file is called bastion.inf. This file is an ASCII text file. You can take a look at it in your favorite editor, but it's best viewed with the SCE Microsoft Management Console snap-in.

C:> secedit /configure /cfg bastion.inf /db %TEMP%/secedit.sdb
/verbose /log %TEMP%/secelog.txt

This will make a number of changes to your configuration. Here is a summary of the most significant changes:

**Account policies**

**Password policy**
- Enforce password uniqueness by remembering last passwords: 6
- Minimum password age: 2
- Maximum password age: 42
- Minimum password length: 10
- Complex passwords (passfilt.dll): Enabled
- User must logon to change password: Enabled

**Account lockout policy**
- Account lockout count: 5
- Lockout account time: Forever
- Reset lockout count after: 720 mins

**Local policies**

**Audit policy**
- Audit account management: Success, Failure
- Audit logon events: Success, Failure
- Audit object access: Failure
- Audit policy change: Success, Failure
- Audit privilege use: Failure
- Audit process tracking: No auditing
- Audit system events: Success, Failure

**User rights assignment**
- SeAssignPrimaryTokenPrivilege: No one
- SeAuditPrivilege: No one
- SeBackupPrivilege: Administrators
- SeCreatePagefilePrivilege: Administrators
- SeCreatePermanentPrivilege: No one
- SeCreateTokenPrivilege: No one
- SeDebugPrivilege: No one
- SeIncreaseBasePriorityPrivilege: Administrators
- SeIncreaseQuotaPrivilege: Administrators
- SeInteractiveLogonRight: Administrators
- SeLoadDriverPrivilege: Administrators
- SeLockMemoryPrivilege: No one
- SeNetworkLogonRight: No one
- SeProfileSingleProcessPrivilege: Administrators
- SeRemoteShutdownPrivilege: No one
SeRestorePrivilege Administrators
SeSecurityPrivilege Administrators
SeShutdowmPrivilege Administrators
SeSystemEnvironmentPrivilege Administrators
SeSystemProfilePrivilege Administrators
SeSystemTimePrivilege Administrators
SeTakeOwnershipPrivilege Administrators
SeTcbPrivilege No one
SeMachineAccountPrivilege No one
SeChangeNotifyPrivilege Everyone
SeBatchLogonRight No one
SeServiceLogonRight No one

Event Log Settings
The Application, System, and Security logs are configured to be up to 100MB each. They will overwrite events as needed, but only entries older than 30 days. Anonymous access to the logs is disabled.

Registry Values
The policy will also apply the following changes to the registry.

<table>
<thead>
<tr>
<th>KEY</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MACHINE\System\CurrentControlSet\Control\Print\Providers\LanMan Print Services\AddPrintDrivers</td>
<td>REG_DWORD</td>
<td>1</td>
</tr>
<tr>
<td>MACHINE\System\CurrentControlSet\Services\Rdr\Parameters\EnablePlainTextPassword</td>
<td>REG_DWORD</td>
<td>0</td>
</tr>
<tr>
<td>MACHINE\System\CurrentControlSet\Services\LanManServer\Parameters\AutoDisconnect</td>
<td>REG_DWORD</td>
<td>15</td>
</tr>
<tr>
<td>MACHINE\System\CurrentControlSet\Services\LanManServer\Parameters\AutoShareWks</td>
<td>REG_DWORD</td>
<td>0</td>
</tr>
<tr>
<td>MACHINE\System\CurrentControlSet\Services\LanManServer\Parameters\AutoShareServer</td>
<td>REG_DWORD</td>
<td>0</td>
</tr>
<tr>
<td>MACHINE\System\CurrentControlSet\Services\LanManServer\Parameters\EnableForcedLogOff</td>
<td>REG_DWORD</td>
<td>1</td>
</tr>
<tr>
<td>MACHINE\System\CurrentControlSet\Services\LanManServer\Parameters\RequireSecuritySignature</td>
<td>REG_DWORD</td>
<td>1</td>
</tr>
<tr>
<td>MACHINE\System\CurrentControlSet\Services\Netlogon\Parameters\RequireSignOrSeal</td>
<td>REG_DWORD</td>
<td>1</td>
</tr>
<tr>
<td>MACHINE\System\CurrentControlSet\Services\Netlogon\Parameters\EnableSecureChannel</td>
<td>REG_DWORD</td>
<td>1</td>
</tr>
<tr>
<td>MACHINE\System\CurrentControlSet\Services\Network\Parameters\SignSecureChannel</td>
<td>REG_DWORD</td>
<td>1</td>
</tr>
<tr>
<td>MACHINE\System\CurrentControlSet\Control\Lsa\RestrictAnonymous</td>
<td>REG_DWORD</td>
<td>1</td>
</tr>
<tr>
<td>MACHINE\System\CurrentControlSet\Control\Session Manager\ProtectionMode</td>
<td>REG_DWORD</td>
<td>1</td>
</tr>
<tr>
<td>MACHINE\System\CurrentControlSet\Control\Lsa\LmCompatibilityLevel</td>
<td>REG_DWORD</td>
<td>5</td>
</tr>
<tr>
<td>MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\LegalNoticeText</td>
<td>REG_SZ</td>
<td>This is a private system. Unauthorized use is prohibited.</td>
</tr>
<tr>
<td>MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\LegalNoticeCaption</td>
<td>REG_SZ</td>
<td>Hardened by HP Consulting</td>
</tr>
<tr>
<td>MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\Don\DisplayLastUserName</td>
<td>REG_SZ</td>
<td>1</td>
</tr>
<tr>
<td>MACHINE\System\CurrentControlSet\Control\Lsa\CrashOnAuditFail</td>
<td>REG_DWORD</td>
<td>1</td>
</tr>
</tbody>
</table>
Some of the changes above are not essential to the bastion host, since we don’t have any SMB services running on the system, but it’s still good practice to apply them. And the script does it all anyway.

**File System and Registry Access Control Lists**

The ACLs applied to the file system and the registry are identical to what Microsoft ships as the “Highly secure workstation” template in SCE. For details check the bastion.inf file with the SCE snap-in in MMC.

**Administrator Account**

The bastion.inf policy renames the Administrator account to “root.” This should be changed to something unique for your environment. Make sure to have a strong password on the Administrator account as well.

**Remove Unused and Potentially Dangerous Components**

If an attacker gains access to the bastion host, it is crucial that the attacker doesn’t get extra help to establish a back door or gain access to other systems. Therefore it’s good practice to remove unused binaries from the bastion host. The downside of doing this is that it may slow down the administrators as well. Use your judgment here.

**To remove DOS, Win16, OS/2, and POSIX subsystems:**

**KEY**

<table>
<thead>
<tr>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MACHINE\System\CurrentControlSet\Control\Session Manager\SubSystems\Optional</td>
<td>REG_BINARY</td>
</tr>
<tr>
<td>MACHINE\System\CurrentControlSet\Control\Session Manager\SubSystems\Os2</td>
<td>N/A</td>
</tr>
<tr>
<td>MACHINE\System\CurrentControlSet\Control\Session Manager\SubSystems\Posix</td>
<td>N/A</td>
</tr>
<tr>
<td>MACHINE\System\CurrentControlSet\Control\WOW</td>
<td>N/A</td>
</tr>
</tbody>
</table>

Delete the following files:

- %SystemRoot%\system32\ntvda.exe
- %SystemRoot%\system32\krnl386.exe
- %SystemRoot%\system32\psxdll.dll
- %SystemRoot%\system32\psxss.exe
- %SystemRoot%\system32\posix.exe
- %SystemRoot%\system32\os2.exe
- %SystemRoot%\system32\os2ss.exe
- %SystemRoot%\system32\os2srv.exe
- %SystemRoot%\system32\os2 (directory)

Note that some Win32 applications still have 16-bit installation programs (e.g., Firewall-1 3.0). Removing the Win16 or DOS subsystem will obviously break these programs. The system will claim it’s unable to find the executable you are trying to run.
Other potentially dangerous tools are:

- %SystemRoot%\system32\nbtstat.exe
- %SystemRoot%\system32\tracert.exe
- %SystemRoot%\system32\telnet.exe
- %SystemRoot%\system32\tftp.exe
- %SystemRoot%\system32\rsh.exe
- %SystemRoot%\system32\rsh.exe
- %SystemRoot%\system32\rsh.exe
- %SystemRoot%\system32\rsh.exe
- %SystemRoot%\system32\rsh.exe
- %SystemRoot%\system32\rsh.exe

You might even consider removing the actual files for the unused services and drivers from the system, but this might get you in trouble with Microsoft Support if you need to contact them. Also, the next service pack you apply will put them back anyway.

**Open Ports**

Though it's possible to make Windows NT stop listening on all ports, many applications rely on RPC loop-back communication, especially those from Microsoft. For example, Internet Information Server 4.0 breaks if you disable the RPC client or server. However, if you do not need RPC you can disable it by removing the following keys in the registry:

<table>
<thead>
<tr>
<th>KEY</th>
<th>Type</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>MACHINE\Software\Microsoft\RPC\ClientProtocols\ncacn_ip_udp</td>
<td>N/A</td>
<td>REMOVE THIS KEY</td>
</tr>
<tr>
<td>MACHINE\Software\Microsoft\RPC\ClientProtocols\ncacn_ip_tcp</td>
<td>N/A</td>
<td>REMOVE THIS KEY</td>
</tr>
<tr>
<td>MACHINE\Software\Microsoft\RPC\ServerProtocols\ncacn_ip_udp</td>
<td>N/A</td>
<td>REMOVE THIS KEY</td>
</tr>
<tr>
<td>MACHINE\Software\Microsoft\RPC\ServerProtocols\ncacn_ip_tcp</td>
<td>N/A</td>
<td>REMOVE THIS KEY</td>
</tr>
</tbody>
</table>

This will leave you with no open ports whatsoever on your bastion host:

```
C:\>netstat -an
Active Connections
    Proto  Local Address   Foreign Address  State
C:\>
```

If you do need RPC, the RPC end-point mapper service (RpcSs.exe) will open up some ports.

**Output of netstat on my test system:**

```
C:\>netstat -an
Active Connections
    Proto  Local Address   Foreign Address  State
TCP    0.0.0.0:135      0.0.0.0:0        LISTENING
TCP    0.0.0.0:135      0.0.0.0:0        LISTENING
TCP    0.0.0.0:1027     0.0.0.0:0        LISTENING
TCP    0.0.0.0:1028     0.0.0.0:0        LISTENING
TCP    127.0.0.1:1025   0.0.0.0:0        LISTENING
TCP    127.0.0.1:1025   127.0.0.1:1028   ESTABLISHED
TCP    127.0.0.1:1026   0.0.0.0:0        LISTENING
TCP    127.0.0.1:1028   127.0.0.1:1025   ESTABLISHED
UDP    0.0.0.0:135      *:*              LISTENING
C:\>
```

We will have to live with this. The TCP/IP security filters should deny any connection attempts made to those ports.
Test of TCP/IP Security Filters

Let's try the TCP/IP security filters. First I configured the filters to allow only tcp/80 and udp/1111. Then I fired up listeners with netcat (http://www.i0pht.com/~weid/netcat/) on tcp/80,81 and udp/1110,1111. To test I used netcat to try to connect to the server on the listener ports.

The tcpdump output below shows the behavior of the filter function with SP4.

**UDP packets to port 1110 (blocked) shows no output on the netcat listener.**
22:54:14.041112 arp who-has 10.0.0.43 tell 10.0.0.5
22:54:14.041171 arp reply 10.0.0.43 is-at 0:10:5a:e6:cf:7d
22:54:14.044210 10.0.0.5.1252 > 10.0.0.43.1110: udp 10
22:54:16.090514 10.0.0.5.1252 > 10.0.0.43.1110: udp 11

**UDP packets to port 1111 (unblocked) shows output on the netcat listener.**
22:58:30.045340 10.0.0.5.1254 > 10.0.0.43.1111: udp 10
22:58:32.087053 10.0.0.5.1254 > 10.0.0.43.1111: udp 11

**UDP packets to port 1111 (unblocked) with no netcat listener sends ICMP udp port unreachable.**
23:00:39.497178 10.0.0.43 > 10.0.0.5: icmp: 10.0.0.43 udp port 1111 unreachable
23:00:39.725978 10.0.0.5.1255 > 10.0.0.43.1111: udp 2
23:00:39.726038 10.0.0.43 > 10.0.0.5: icmp: 10.0.0.43 udp port 1111 unreachable
23:00:39.979497 10.0.0.5.1255 > 10.0.0.43.1111: udp 5

**TCP connect to port 80 (unblocked) shows output on the netcat listener.**
23:03:05.220808 10.0.0.5.1264 > 10.0.0.43.http: S 52482:52482(0) win 8192 <mss 1460> (DF) [tos 0x10]
23:03:05.220922 10.0.0.5.1264.http > 10.0.0.43.1264: P 17(6) ack 1 win 8760 (DF) [tos 0x10]
23:03:05.221044 10.0.0.5.1264 > 10.0.0.43.http: P 17(6) ack 1 win 8760 (DF) [tos 0x10]
23:03:07.289221 10.0.0.5.1264.http > 10.0.0.43.1264: P 7:8(1) ack 1 win 8760 (DF) [tos 0x10]
23:03:07.395725 10.0.0.5.1264.http > 10.0.0.43.1264: P 7:8(1) ack 1 win 8760 (DF) [tos 0x10]
23:03:11.04311010.0.0.5.1264 > 10.0.0.43.http: P 7:8(1) ack 1 win 8760 (DF) [tos 0x10]
23:03:11.960993 10.0.0.5.1264 > 10.0.0.43.http: R 52490:52490(0) win 0 (DF) [tos 0x10]

**TCP connect to port 81 (blocked) shows no output on the netcat listener. NT sends RST.**
23:23:43.669792 10.0.0.5.1286 > 10.0.0.43.81: S 52552:52552(0) win 8192 <mss 1460> (DF) [tos 0x10]
23:23:43.669857 10.0.0.5.1286 > 10.0.0.43.81: R 0:0(0) ack 52553 win 0
23:23:44.168936 10.0.0.5.1286 > 10.0.0.43.81: S 52552:52552(0) win 8192 <mss 1460> (DF) [tos 0x10]
23:23:44.168995 10.0.0.5.1286 > 10.0.0.43.81: R 0:0(0) ack 1 win 0
23:23:44.696639 10.0.0.5.1286 > 10.0.0.43.81: S 52552:52552(0) win 8192 <mss 1460> (DF) [tos 0x10]
23:23:44.6969710.0.0.5.1286 > 10.0.0.43.81: R 0:0(0) ack 1 win 0
23:23:45.170337 10.0.0.5.1286 > 10.0.0.43.81: S 52552:52552(0) win 8192 <mss 1460> (DF) [tos 0x10]
23:23:45.170392 10.0.0.5.1286 > 10.0.0.43.81: R 0:0(0) ack 1 win 0

The TCP/IP security filters work well on Windows NT 4.0 SP4.

If the filters are enabled, NT will ignore UDP packets, and TCP connection attempts will be reset on the denied ports.

Secure the Application

The last step is to make a security review of the application that is going to run on the system. This might include NTFS ACLs/Auditing and checking with application vendors for known holes and workarounds or patches.

Summary

Now your system is reasonably well secured. The only way of breaking into it over the network (as far as I can tell) is by exploiting a vulnerability in the applications running on the host (or possibly the MS IP-stack) to run arbitrary code that opens up the system.

We've basically rendered our system inoperable from a management perspective.

Windows NT does not provide us with remote logging. NT-based remote-administra-
tion tools like the Event Viewer and Server Manager are based on NETBIOS, and the problem with NETBIOS is that it's considered a no-go in perimeter networks. This is because everything runs in NETBIOS (SMB/CIFS, management, and other applications based on named pipes), which means you cannot limit traffic to a host in router access control lists in a granular way. Hence we have to find other – preferably standardized – ways of administering and monitoring the Windows NT host.

**Relevant MS Knowledge Base Articles**

Microsoft Support Knowledge Base is available at <http://support.microsoft.com/support/search>.

Use "Search for a specific article ID number" and type in the PSS ID number.

<table>
<thead>
<tr>
<th>PSS ID Number</th>
<th>Name of article</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q93362</td>
<td>C2 Evaluation and Certification for Windows NT</td>
</tr>
<tr>
<td>Q101063</td>
<td>Windows NT Logon Welcome, Displaying Warning Message</td>
</tr>
<tr>
<td>Q114463</td>
<td>Hiding the Last Logged On Username in the Logon Dialog</td>
</tr>
<tr>
<td>Q114817</td>
<td>No Shutdown Button in Windows NT Server Welcome Screen</td>
</tr>
<tr>
<td>Q140058</td>
<td>How To Prevent Auditable Activities When Security Log Is Full</td>
</tr>
<tr>
<td>Q142641</td>
<td>Internet Server Unavailable Because of Malicious SYN Attacks</td>
</tr>
<tr>
<td>Q143164</td>
<td>INF: How to Protect Windows NT Desktops in Public Areas</td>
</tr>
<tr>
<td>Q143474</td>
<td>Restricting Information Available to Anonymous Logon Users</td>
</tr>
<tr>
<td>Q143475</td>
<td>Windows NT System Key Permits Strong Encryption of the SAM</td>
</tr>
<tr>
<td>Q146906</td>
<td>How To Secure Performance Data in Windows NT</td>
</tr>
<tr>
<td>Q147706</td>
<td>How to Disable LM Authentication on Windows NT</td>
</tr>
<tr>
<td>Q151082</td>
<td>HOWTO: Password Change Filtering &amp; Notification in Windows NT</td>
</tr>
<tr>
<td>Q153094</td>
<td>Restoring Default Permissions to Windows NT System Files</td>
</tr>
<tr>
<td>Q155363</td>
<td>HOWTO: Regulate Network Access to the Windows NT Registry</td>
</tr>
<tr>
<td>Q161372</td>
<td>How to Enable SMB Signing in Windows NT</td>
</tr>
<tr>
<td>Q161990</td>
<td>How to Enable Strong Password Functionality in Windows NT</td>
</tr>
<tr>
<td>Q166992</td>
<td>Standard Security Practices for Windows NT</td>
</tr>
<tr>
<td>Q172925</td>
<td>INFO: Security Issues with Objects in ASP and ISAPI Extensions</td>
</tr>
<tr>
<td>Q172931</td>
<td>Cached Logon Information</td>
</tr>
<tr>
<td>Q174840</td>
<td>Disabling Buttons in the Windows NT Security Dialog Box</td>
</tr>
<tr>
<td>Q176820</td>
<td>Differences Between 128-bit and 40-bit versions of SP3 &amp; SP4</td>
</tr>
<tr>
<td>Q187506</td>
<td>List of NTFS Permissions Required for IIS Site to Work</td>
</tr>
<tr>
<td>Q195227</td>
<td>SP4 Security Configuration Manager Available for Download</td>
</tr>
<tr>
<td>Q214752</td>
<td>Adding Custom Registry Settings to Security Configuration Editor</td>
</tr>
<tr>
<td>Q217336</td>
<td>TCP/IP Source Routing Feature Cannot Be Disabled</td>
</tr>
<tr>
<td>Q218473</td>
<td>Restricting Changes to Base System Objects</td>
</tr>
</tbody>
</table>

**REFERENCES**


**Other Resources**


Learning with Source Code UNIX

My sister’s high-school-age son is applying to the computer science department of Carnegie Mellon University. When I expressed my surprise, she said, “Oh, Evan is very good with computers.” I proceeded to dig myself into trouble by asking what Evan does with computers.

As I suspected, he is an end user of applications. I said that if I were on the admissions board, I’d be looking for something more substantial to demonstrate an interest in the field. I suggested that if he were really interested in computers, he could install a copy of Linux on his computer, start writing programs, and learn Java. Then he would have something to show for his time in front of the screen.

To learn computer systems and programming, it is essential to acquire some starting knowledge, design and implement a solution to a problem, and then have your work reviewed by an expert. Ideally, the evaluation step is an ongoing, interactive process with one or more experienced mentors. A supplemental method is having Source Code UNIX in your corner. The running operating system, its utilities, and its thousands of ported applications, backed up with manuals and other printed documentation, provide a powerful surrogate for a real instructor.

In this month’s article, I’ll

- elaborate on the extensive learning material available.
- discuss the importance of coding style and conformity.
- provide some personal examples of how I’ve learned from and leveraged Source Code UNIX systems.

Learning Material

Consider mainstream, shrink-wrapped software. Load it and use it. You can only surmise what is happening inside. Need altered behavior? You’re out of luck. In contrast, ported applications are built from the source code. (See my October 1998 article, <http://boulderlabs.com/4.ports.html>, for a tour.) So if you need altered behavior, it may be relatively simple to achieve. You may need to develop a program that has a number of similarities to existing UNIX code. Why not learn how others have solved various aspects of your problem and leverage the experience from their working base?

Books can provide valuable in-depth design discussion for even the most thoroughly documented code. My favorite example is The Design and Implementation of the 4.4BSD Operating System, by McKusick, Bostic, Karels, and Quarterman (Addison-Wesley, 1996). For those curious about the internals of BSD variants, this is the gospel, and it is best summed up in the book’s dedication:

This book is dedicated to the BSD community. Without the contributions of that community’s members, there would be nothing about which to write.

FreeBSD, NetBSD, OpenBSD, and BSDI all derive their base code from 4.4BSD, which is now about five years old. Fortunately, McKusick, Bostic, Karels, Leffler, and Greenman have signed a contract with Addison-Wesley to produce a follow-on version for The Design and Implementation of the FreeBSD Operating System, due out about mid-2001.

The state of computer science was greatly advanced when Gary Wright and the late W. Richard Stevens published TCP/IP Illustrated Volume 2 (Addison-Wesley, 1995). This
Programming is one of those areas where unorthodox style is generally not appreciated because, invariably, other people will need to look at and understand your code.

book explains the workings of the 4.4BSD networking code. The data structures, algorithms, and thought processes behind much of the work are explained in detail. Given that TCP/IP is now the universal network protocol, this book is indispensable.

Other resources for understanding source code include Web sites, tutorials, newsgroups, Frequently Asked Questions (FAQs), mailing lists, and search engines. General system-level documentation that helps explain source code can be found with the search engines. Some particularly valuable Web sites:

  <http://sunsite.unc.edu/mdw/HOWTO/Installation-HOWTO.html>
  <http://www.dejanews.com>
  <http://www.freebsd.org/handbook>
  <http://www.freebsd.org/search.html>
  <http://www.freebsd.org/tutorials>
  <http://www.linux.org/help/howto.html>

If you want to be into the action, subscribe to mailing lists such as <freebsd-security@FreeBSD.ORG> for daily or even hourly activity <http://www.freebsd.org/support.html#mailing-list>.

By the way, before bothering busy people on particular lists with your questions, be polite by checking if your question has already been answered. Learn how to use <http://www.dejanews.com> or archival search engines such as <http://www.freebsd.org/search/search.html> for mailing lists and newsgroups.

As an outside interest and hobby, I work with maps, GPS, and astronomy. The huge body of knowledge and source code available on these topics can be found with a Web search engine. Subscribe to newsgroups such as <sci.geo.satellite-nav> to be connected to the group of contributors. Niels Elgaard Larsen has implemented software that places GPS track points on maps (<http://www.diku.dk/users/elgaard/eps>). Its GNU General Public License allows me access to the Java source code, and I can make the modifications I need for my project.

Interactions with individuals often are the best way to learn. Make the effort to attend conferences in areas that interest you. Go to the Birds-of-a-Feather (BOF) sessions to meet with workers and disciples. From those sessions, you'll be able to establish one-to-one relationships that can continue with email and telephone calls.

Coding Style and Conformity

Programming is one of those areas where unorthodox style is generally not appreciated because, invariably, other people will need to look at and understand your code. The problem is that the reader would need to put himself into an unnatural frame of reference to comprehend your phrases, and his basic assumptions regarding indentation or other common practices cannot be used. Imagine how much harder it would be for the home remodeler to accomplish her work if she could not rely on conventions about stud spacing in walls and electrical practices.

Steve Bourne, the original author of /bin/sh, used the C preprocessor to give an Algol feel to his 1979 code. Constructs such as:

```
#define IF if{
#define THEN }
#define ELSE } else {
#define FI ;
```

led to implementation code looking like this:
 Granted, it is cute and interesting, but I claim that he did the community a disservice with that style. As a reader, I am constantly forced to look up the meanings of his constructs. For example, C statements are semicolon-terminated, but Bourne's code (e.g., the DO...OD construct) confuses this principle.

How did I come to this opinion? By looking at hundreds of thousands of lines of code over years. You develop a feel for what is good style, and you easily become annoyed by "individuals" who want to express themselves. The best styles are those that don't seem to have any style at all, like the national TV news anchor who seems to have no accent at all. You should be able to look at a body of code and not find any surprises with indentation, braces, or idioms.

I believe the best way to learn good programming and good style is to design and implement a solution yourself first. Then get feedback and comments from others. You'll eventually notice a consensus. Kernighan and Pike's _The Practice of Programming_ (Addison-Wesley, 1999) is a gem for improving one's code. The authors show various solutions to problems in various languages and analyze the strengths and weaknesses of each. The beauty of their work is that they lead you along a normal solution path and show how simplicity, clarity, and generality can be gained along the way.

For those wishing for a historical perspective on an operating system design and style, John Lions, in 1977, published two books: A Commentary on the UNIX Operating System and its companion source-code listing for his course at the University of New South Wales. After years of suppression (as trade secrets) by various owners of the UNIX code, the books were rereleased (Peer to Peer Communications, 1996). Greg Rose, one of John's students, wrote:

John introduced a course in Operating Systems, and decided to study the Unix operating system. One of his motivations in doing this was to introduce the students to code which was well written by other people -- at the time, this was not a common practice, although it is now well accepted.

**Personal Examples**

The disadvantage of learning from books is that the problems tackled are seldom the ones you are faced with. That's why running a Source Code UNIX system is important -- you're likely to find some kernel facility or user application that largely overlaps with your problem. For example: in 1991, when designing my passive solar house, I wanted software to tell me the exact solar sky for my location at any time of the day throughout the year.

Table 1 shows the output from my program. You see that for January 1, at 12:00 the sun rises to only 27 degrees elevation and is pointing almost due south (179 degrees). As expected, this date has the fewest hours of sunlight, with the sun sweeping the lowest arc in the sky. (Of course, if I printed daily activity, December 22 would show as the shortest day of the year.)
Whenever possible, add to the body [of knowledge] by making your own software clear, robust, and available under some kind of general public license.

The code (<http://boulderlabs.com/dailySun.c>) is leveraged from a friend's spherical-navigation code, but I could have easily worked with Bill Randle's public calentool package. My sunrise, sunset program (<http://boulderlabs.com/riseset.tz>), which uses calentool code, presents everything I want to know about both the sun and the moon patterns, including Julian days, local sidereal time, and declination of the earth. I developed a curiosity about things like the equation of time and found an excellent discussion on the Web at <http://susdesign.com/sunangle>. Further, in a Java FAQ, I once saw that extensive libraries were implemented for calendars and date calculations, so I grabbed the Java source code from <http://java.sun.com> and studied the fascinating code and comments in Date.java, GregorianCalendar.java, and TimeZone.java.

I often record radio talk shows on my computer because it's easy to schedule (crontab), and it's easy to gain random access to the content when I later play it back. Most computer systems come with a GUI player, but for my needs that kind of an interface is clumsy. I found some audio source code and in a couple of hours added the features I wanted for command-line control. Simply, I wanted periodic printing of the time-code and file byte offset and an easy way to skip and maneuver within the file.

I urge you to take advantage of the knowledge embedded in Source Code UNIX. Whenever possible, add to the body by making your own software clear, robust, and available under some kind of general public license.

To end, I would like to highlight a huge event and honor three heroes in the history of UNIX.

Up through about 1991, all UNIX users had to be under some kind of a license arrangement to access the source code. This was in spite of the fact that most of the Bell Labs UNIX code base over time had been replaced with better, more functional software from the huge body of public contributors. Keith Bostic, Mike Karels, and Kirk McKusick at Berkeley realized that most of the BSD UNIX system could be released to the public without the traditional AT&T/USL/Novell license, because it was publicly developed. They boldly proceeded to freely redistribute the system, resulting in USL initiating a lawsuit for an injunction to stop the software release. In 1994 the pioneers from Berkeley prevailed, and now anyone can have the 4.4BSD system or its derivatives, FreeBSD, OpenBSD, and NetBSD. For a great story and more details see <http://www.oreilly.com/catalog/opensource/book/kirkmck.html>. 
politeness in computing

Does Saying “Please Log In” Give Anyone and Everyone the Right to Be in Your System?

Welcome to the first of what I hope will be a series addressing legal issues facing system administrators and the technical community in general. Future columns will deal with free speech, privacy, and censorship; intellectual property; export issues; policies and procedures and how they relate to system administrator liability; computer crime; and other issues or situations that you raise. If you have questions about a particular issue facing you, or comments on a particular column, please feel free to send them to me.

This column addresses one of the great debates in system administration — whether you can prosecute someone for cracking your system if it says “Please log in” at the prompt. After all, if it says “Please log in” and someone does that, whether authorized or not, that person has only done what you asked, right?

Dealing with the law is a lot like dealing with computer systems — law has its own language, areas of specialization with specific rules (which can sometimes interact in very strange, unexpected, and counterintuitive ways), and processes and procedures. Just as with technology, once you understand how the law works, you can apply that knowledge and understanding to new situations.

Like computer systems, law is built on the structure of history. In law, that structure is the laws that have been passed by Congress or the states and the decisions made by courts — some dating as far back as colonial or Roman times.

When a U.S. court is faced with a particular issue, the first question the judge (or judges) will ask is whether or not there is a U.S. federal or state law addressing the issue. If there is such a law, then the judge will look to see if a higher court that is directly in line above that court has interpreted how the law applies to the issue. For example, if you are in a U.S. district court, then the judge will look at decisions made by the circuit court that is directly above that district court, as well as to the U.S. Supreme Court. If one of those two courts has ruled on the issue, then that is considered a binding precedent to which the lower court will defer.

If there is no similar decision from a higher court directly in line above that lower court, then the lower court will look to decisions made by other higher courts and other courts at the same level as the deciding court. For example, a district court will look for rulings from U.S. circuit courts and other U.S. district courts.

If there is no specific law on the subject, the judge will follow the same procedure as above, looking at decisions of other courts. This is the analysis in which the decisions from English, colonial, and even Roman courts can come into play. A great deal of U.S. property law, for example, is based on what is called the “common law” established by English courts before the U.S. declared its independence. This is important because if there is no binding precedent, as is often the case with the technology arena, courts will frequently look to analogous areas of the law, public-policy considerations, and common sense to determine an appropriate course.

At this point, you may be wondering when I’m going to answer the question I’ve posed; but understanding how the different components of a system interact is the key to understanding the answer.
The court held that the intention requirement in the act required only that the unauthorized user intend to gain access to the system, not that the user intend to cause damage.

To analyze this question, we first look to see if there is any federal or state law addressing it. The Federal Computer Fraud and Abuse Act (CFAA)[2] states:

Whoever ... (A) knowingly causes the transmission of a program, information, code, or command, and as a result of such conduct, intentionally causes damage without authorization, to a protected computer; (B) intentionally accesses a protected computer without authorization, and as a result of such conduct, recklessly causes damage; or (C) intentionally accesses a protected computer without authorization, and as a result of such conduct, causes damage; ... shall be punished as provided in subsection (c) of this section.[3]

Section 1030 of the CFAA defines a “protected computer” as a computer (A) exclusively for the use of a financial institution or the United States Government, or, in the case of a computer not exclusively for such use, used by or for a financial institution or the United States Government and the conduct constituting the offense affects that use by or for the financial institution or the Government; or (B) which is used in interstate or foreign commerce or communications and defines “damage” as any impairment to the integrity or availability of data, a program, a system, or information, that (A) causes loss aggregating at least $5,000 in value during any 1 year period to one or more individuals;[4] (B) modifies or impairs, or potentially modifies or impairs, the medical examination, diagnosis, treatment, or care of one or more individuals; (C) causes physical injury to any person; or (D) threatens public health or safety.

For the purposes of this article, importantly, the Act does not discuss whether the login prompt of a system has warnings about accessing the system or just says “Please log in.” The law merely says “without authorization.” So, the question remains: does saying “Please log in” automatically authorize anyone to be in your system?

Since the relevant federal law covers virtually every computer but does not specifically address the issue, I will focus on court decisions interpreting the Act rather than look for state statutes.[5] Since this is a hypothetical case, and the federal district or state is indefinite, I will look at decisions from all federal courts.

In the case of U.S. v. Sablan,[6] Sablan had recently been fired from the Bank of Hawaii’s Agana, Guam, branch for circumventing security procedures in retrieving files. Sablan left a bar where she had been drinking with a friend and entered the closed bank through an unlocked loading-dock door. She went to her former work site (using a key she had kept) and used an old password to log into the bank’s mainframe. Sablan contended that she then called up several computer files and logged off. The government asserted that Sablan changed several of the files and deleted others. Under either party’s version of the story, Sablan’s actions severely damaged several bank files.

Sablan was convicted of violating §1030 of the CFAA at trial; on appeal, Sablan argued that she did not intend to damage the bank’s files, and that §1030 required that she have such an intention. The court held that the intention requirement in the act required only that the unauthorized user intend to gain access to the system, not that the user intend to cause damage. Nowhere in the analysis of the case did the court focus on whether the bank’s system had a warning on it or the invitation “Please log in.”

In the case of U.S. v. Czubinski,[7] Czubinski was an employee of the IRS who, as part of his job, routinely accessed information about taxpayers from one of the IRS data-
bases. Using a valid userid and password, Czubinski was able to access income-tax return information for virtually any taxpayer. IRS Rules of Conduct specified that employees could not use any IRS computer system for other than official purposes. Czubinski, solely out of curiosity, conducted searches and browsed files online that were not related to his job. Czubinski was prosecuted for violating $1030. At trial, Czubinski did not argue that he was authorized to view the files, and the issue of whether the login prompt said "Please log in" or not was not raised. Czubinski was acquitted, however, because he merely looked at the files and did not do any damage, disclose the information that he found, or "obtain anything of value."[8]

Neither the law nor the courts appear to address directly the question of whether the prompt saying "Please log in" automatically authorizes a hacker (or other undesired user) to be in the system. Indirectly, however, the courts have provided two examples of an "unauthorized user," and these examples make intuitive sense. Both Sablan and Czubinski were unauthorized users because the owners of the computers said they were, and neither could have reasonably argued that they did not intend to get into a system they knew they were not supposed to access.

To see why this makes such intuitive sense, let’s look to an analogous area of the law, property law, to see if we can analogize someone unauthorized getting into a network to situations in the real world. Let’s say your system is your house, the login prompt is the front door, the userid (or other authentication device) is the equivalent of looking through the peephole to see who’s at the door, and the password is the key. Assuming your network has a prompt that says "Please log in," is there a real-world equivalent and should it make a difference to whether or not you can prosecute an unauthorized user?

**Scenario 1**

Suppose that you are home for the evening and lock the door to your house. Suddenly you hear the lock turn and a complete stranger who has picked the lock comes in. Is that person committing a crime even if you have a "Welcome" mat down on the front porch? Of course – she is guilty of breaking and entering and trespassing. The "Welcome" mat is not considered a blanket invitation to everyone to come into your house. By analogy, if someone comes to the door of your network and hacks in, that person is not welcome, even if the login prompt says "Please log in." The logic here is relatively simple – the "Welcome" mat (or "Please log in") invitation applies only to those people you actually want to be there. The invitation to "Please log in" is an invitation for the person at the door to prove his identity (userid or other authentication device) and use his "key" (password) to open the door.

**Scenario 2 (Modified Czubinski Scenario)**

Since Scenario 1 seems obvious, let’s make it a little more questionable. You still have the "Welcome" mat on your front porch. Suppose a friend needs to pick something up at your house. You give your friend a key to your house and tell your friend that what he needs will be on the dining room table. Your friend goes to your house, walks in, and, in addition to picking up what he was supposed to, your friend goes exploring through the house, sees some money in the bedroom, and takes it. Has your friend committed a crime, despite the fact that you gave him a key? Yes. You authorized your friend to enter your house to perform a specific activity. Your friend exceeded the rights that you had granted and took property he was not allowed to take.

In this case, as in Czubinski, you knew the identity of the person going into your house (the userid), he had the right key (password), and he was there with your permission.

Indirectly, however, the courts have provided two examples of an "unauthorized user," and these examples make intuitive sense.
But, your friend exceeded the rights that he had been granted. Where Czubinski merely looked, your friend took something of value. In property law as well as under §1030, your friend has committed a crime.

Scenario 2a
Say that in Scenario 2 your friend, instead of stealing money, goes exploring through your house. While in the bedroom, he knocks a valuable vase off the dresser, shattering the vase. Is your friend liable for the vase? Under property law and the logic of §1030, probably. Your friend was not authorized to be in the bedroom, and he intended to go somewhere that he was not authorized to go, so he is still liable for any damage he causes.

Scenario 3 (The Sablan Scenario)
Your significant other has a key to your house. When you break up, your ex keeps the key. (You still have the “Welcome” mat on your front porch.) After spending a night drinking, your ex comes over to your house, uses the key, and either intentionally (the prosecution’s story) or accidentally (Sablan’s story) breaks something. Under the logic of §1030, your ex intended to enter your house without authorization, and is liable for any damage she does while in there, regardless of whether it was intentional or not. Even if your ex does no damage, she is still trespassing, because you have revoked her authorization to be there.

Conclusion
Section 1030 of Title 18 of the U.S. Code criminalizes unauthorized access to a computer system and intentional damage to a computer system by an authorized user where the computer system is used in interstate or foreign commerce or communications. Neither the statute nor the cases specify whether saying “Please log in” at the prompt automatically authorizes any person to access your system. Because neither focuses on the issue, we might assume that the answer to the question seemed obvious at the time — that the system owner gets to define who is authorized and who is not. (But we all know what happens to those who assume, especially when it comes to politicians and technology.) However, in this case, the assumption is probably valid. Looking at the way another area of the law deals with analogous situations, and applying common sense, we can say that saying “Please log in” should not grant anyone the right to access your system, just as placing a “Welcome” mat outside your door does not give anyone the right to enter your house.
interviewing for sysadmins

In my previous article (login; April 1999), I wrote about preparing a resume of qualifications for job hunting. In this article, I'll discuss how to handle the interview and other employment screening processes.

The system administration field is different from other high-tech venues; the job hunting and interviewing processes differ, too. In particular, a small core of individuals with niche talents are given an inordinate amount of responsibility. In fields such as applications development, individuals are sometimes hired in groups. System administrators, by contrast, are typically hired individually to fill very specific roles. As a result, the screening and interviewing process for system administrators can reflect this search for individuality, personality, and disparate skill sets.

Get That Interview: HR, Recruiters, and Other Obstacles

For the most part the interview and the steps leading up to it are the same for both contract and regular full-time (permanent) employment. Contract work generally requires fewer interviews, and references are usually not checked. Offer letters for permanent hires take longer than getting the go for a contract position. One of the perennial snags for full-time offers is reference checking.

If you wind up talking with an HR person or a recruiter prior to meeting the hiring manager, plan on taking a different tack. When interviewing with a non-technical or non-hiring authority, figure you need to fluff up a bit to pass the exams. Too often, these folks don’t screen well technically: the recruiter is eager to get anything that has ink on it to the hiring manager, while the HR person typically focuses on interpersonal skills and buzzwords rather than core technical competencies. Always keep in mind, however, as with any sale, you are best positioned when speaking directly with the decision-maker. Keep the interchange with HR and recruiters as short and direct as possible. Get as much information as you can from them (job description, salary, benefits), but keep your sights set on the real boss.

Pre-interview Planning and Hard Questions

Beyond the technical questions that you might be asked, a smattering of other topics might come up during an interview. If you haven’t been faced with them before, it might be initially awkward to respond to them. I’ll cover some of them now to give you some background about your options and some time to think about how you feel about them.

Time Lines

Some of the more difficult circumstances for interviewing are: recent graduation from college, extended absence from the workplace, or transitions from jobs that have lasted more than six or seven years. In these situations, use good judgment, talk with others to get current information, apply due diligence in your research, and create and follow a plan. No interviewer should ask you personal questions, but you should still be prepared to address questions about the dates in your resume or on your application. Keep your responses short, professional, and future-oriented. Never bad-mouth anyone; inside or outside of an interview, it’s taken as a red flag that you may be difficult to work with. And, if you’re between positions, it’s best to be honest about it, if asked.

Drug Tests

Except for certain safety-sensitive positions, such as that of airline pilot, there are now no federal or state laws that require drug testing. Still, certain companies place a great deal of importance on maintaining the image of a drug-free work place. This kind of
There are limits to what former employers may say about you and your performance on a previous job. Because of added liability, certain employers will confirm only start and stop dates, title, and rate of pay.

Testing must be required of all applicants as a condition of employment and be performed by an independent laboratory. Further, you can expect to be tested periodically while on the job. Even if you don’t use drugs, you should consider the implications for your privacy and in other areas of employment.

**Psychological Profiles**
Be wary of psychological-profile tests. Administered by qualified people in the right circumstances, they can be enlightening. When the questions are delivered verbally, in an ad hoc fashion and in the context of an interview, they are a nuisance at best. This is especially likely in small- to medium-sized companies. You might think of asking how many thousands of people have been screened using that exact technique. If it is a one-sesy-twosey operation you might as well try your hand at techie tarot.

We actually had an acquaintance who was asked a series of questions that bordered on the ludicrous: “Do you see the pond out there?” the interviewer asked. “If you were a duck, what kind of duck would you be?” Our fearless candidate responded without hesitation, “Whaddya mean what kind of duck would I be, what have you guys been smoking, anyway?” Although the story ends with our friend being escorted to the door, I maintain that the response was more appropriate than the question.

**Security Clearances**
U.S. Government security clearances can be requested by, and checked by, only other U.S. Government agencies. There should be no reason to provide government security documents to a civilian party.

**Insurance**
Purchase of special insurance such as Errors and Omissions, business liability, or bonding should not be required for W2 regular full-time employees, or W2 employees of agencies. You may, however, be required to carry automobile insurance.

**Background Checks**
There are limits to what former employers may say about you and your performance on a previous job. Because of added liability, certain employers will confirm only start and stop dates, title, and rate of pay. Depending on the state, you can be asked to provide a driver’s license number, employment history, employment salary, and whether or not you have been convicted of a felony.

**NDAs, Noncompetes, and Intellectual Property Rights**
These documents, if required, may be presented prior to the interview along with an employment application.

If you find legal-speak on an employment application and you don’t understand it, don’t sign it. You have the right to consult with an attorney. Consider asking for a photocopy of anything that you do sign.

An important word on contracts in general: they did not come down the mountain on clay tablets. Although this may cause delays, if you vehemently disagree with something (in an NDA or your employment contract, for example), mark it up and see whether they agree to your changes.

While parts of your contract, such as salary and benefits, may not have wiggle room, other areas, including scope of duties, training, or review periods, may be wide open. Be aware that employment contracts can be and are often altered; it’s just that most people don’t want to confront an authority figure. It’s important that someone who makes such a request have a valid point. Also, senior staff can pull this off more easily than recent college graduates.
The nondisclosure agreement usually concerns dissemination of proprietary information or trade secrets by employees and former employees. An NDA may also require you to testify or provide information for your employer in a legal case. Generally the information provided in the first interview will be unrestricted enough not to require an NDA. Take your time and think about it.

Noncompete agreements concern an employee leaving a company and forming his or her own company using proprietary information gained while on the job. Noncompete agreements must be limited in geography and time. An employee has the “right to earn a livelihood,” which often makes noncompetes hard to enforce.

IP does stand for something else. Intellectual property rights include patents, designs, copyrights, and related “know-how” rights. If you suspect that a product or idea of yours may fall into this area, you should consult with an attorney. You can expect that a release of your rights will begin with a list of prior inventions. This is again a case where there is no harm in the delay caused by a thorough perusal of the document. The question often comes down to whether you are being employed as an inventor and developer or to perform more general duties. Most jobs for system administrators do not require this.

The Interview
Phone Interview
While searching for a job you may come across the opportunity for a “phone interview.” Assuming that you are interested in the job and jazzed to go, I would suggest that you skip the phone interview if possible. The advantage of jumping directly to the in-person interview is that much of the decision is weighted on personality, which cannot be displayed in a phone interview. There is also the matter of timing: a phone interview might put you out another week or more before you meet the employer for an in-person session. Finally, an employer who has time for only a phone interview might not be serious about actually hiring you.

The On-Site Interview
Now that you’ve finally got your foot all the way in the door, it’s time to put your best one forward. In fact, much of the acceptance of a candidate is based on personality. If you don’t make it past eye contact and a handshake, no amount of technical wizardry will pull you through.

Because so much of how we come across is “impression” as much as technical competence, it’s important to take some time to “look good.” Depending on your geographical area and the type of job you’re looking for, a suit may not be required, but it’s never in poor taste to dress well for an interview. At the very least, dressing well conveys that you care about your meeting. Similarly, be prompt for your interview. If by some misfortune you must be late, take the time to call and inform the person you’re going to meet of the necessary change in plans; always call in advance.

Set personal goals for the meeting. Do you really want the job? Perhaps you want to go to an interview as a trial run, to polish your skills. As you move along in your career, you’ll probably be asking more of these questions, and you’ll be more discriminating when it comes to selecting the best opportunity.

When you meet with the hiring manager, your resume may serve as the tool that guides the interview. If you didn’t need the resume to get the interview, for example in the case of a word-of-mouth-referral, my advice is “don’t confuse the process.” A resume, good or bad, can be a hindrance at times; presenting information that was not request-
Always ask for the job. I can’t emphasize this enough. Techies, especially, have a way of understating their interest in a position.

ed may result in an objection. The same would be true for providing employment references (or copies of your source code) before being asked. But once you have the interview you should be prepared to furnish all of these, when requested.

Be Prepared
How much do you know about this company? If you are interviewing from a word-of-mouth referral or from a competent agency, get the inside scoop on the company before going out there. In particular, how many people will you meet, how long will the interview run, and what exactly are they looking for? Will you be replacing someone, or is this a new position?

Last-Minute Advice
Interviewing can be an intimidating experience. After all, you probably want it to go well, and that’s enough to make anyone a bit nervous. Try to relax and use the opportunity to tell the interviewer about yourself and your qualifications. I find it helpful to view the questions as a prompt, allowing me to expand on the topic and convey all the pertinent information about my expertise that applies. Remember that the bottom line is to provide the hiring manager with the information needed to offer you the position; this is the time to explain how you’re qualified. If you find the interview turning into a battery of questions and one-word responses, you’re not doing your part to fill in the blanks.

Toward the end of the interview, you might be asked whether you have any questions. If you do, it’s appropriate to bring them up, in a factual way. Asking about the typical career path of the job is one thing; asking the manager how long until her or his job will be open is another!

Always ask for the job. I can’t emphasize this enough. Techies, especially, have a way of understating their interest in a position. If you can’t muster the strength to ask, “So when would you like me to start?” then beat around the bush a little – but do show interest, even if you think you did poorly or have doubts about the job.

Practice Makes Perfect
Consider that a hiring manager has a candidate pool of four to eight individuals to interview. If you’ve decided to change positions, don’t you feel should have the same latitude in selection? If you haven’t interviewed in a year or more, get out and try a few, kick some tires, and practice a bit. You’ll increase your self-confidence and get a better feel for the current market. Even if you’re already up to speed, it’s good to look at more than one opportunity so that when you do accept a job, you will be able to do so knowing that you have made an informed decision.

If you ever do feel you’ve “bombed” an interview, remember it’s not the end of the world, and it may not even be your fault. If you are meeting with hiring managers who are inexperienced, their first few interviews may be rough.

Following Up
I mentioned that references are typically held until you establish a mutual interest. You may be asked to include them in your application. By the time you fill out an application for full-time employment, you should be serious enough to offer references. Employers expect at least one management reference and two peers; three references will generally suffice. I advise having these preprinted to attach to the employment application. Always notify your references before using their names. If an offer is going out, drop your references an email to remind them.
It is considered professional to follow up an employment interview with a letter expressing interest in the position. Following up after a poor interview might even get you a second shot. It always helps to tell someone that you are interested in the position.

A follow-up contact is especially appropriate for a senior position for regular full-time employment with a large company, since it is construed as the beginning of a potential relationship. The letter should be short and to the point. With smaller and less formal companies a simple email may suffice. In this case, try to attach some information pertaining to the interview. Avoid at all costs using a formal response to cover up any errors you feel you may have made during the interview. Broaching the subject of your tardiness or technical questions you bombed will reinforce the negative. I generally don’t advise phone calls unless the manager requests it, but use your judgment, be yourself, and show interest.

**Conclusion**

Don’t be disappointed if that “best job in the world” falls through. Trust me – another, better opportunity will take its place. Use your common sense and be as patient as possible. I’ve worked with companies who take several months to crank out an offer letter for a permanent position. Set your goals before you start the process and stick with them. In today’s world, “you are where you work.” Be careful what you sign up for!
Programmers are used to moving easily up and down levels of abstraction. A directory is made up of files. Each file may have many records, each record many fields, each field many bytes. Expressions give rise to statements, grouped into functions, libraries, and then applications. The ability to operate over so many levels of abstraction is arguably one of the traits that makes us human.

Managers who can move up and down levels of abstraction will be able to use this skill easily and effectively in dealing with people, even those who are not programmers, provided they understand some simple principles. But first some terminology.

As we get more and more abstract, we deal with larger and larger chunks of data. As we get more concrete, we deal with smaller and smaller chunks of data. Imagine a hierarchy of ideas or concepts with the most abstract and all-embracing at the top and the most concrete at the bottom. When we “chunk up,” we move up the hierarchy of ideas. When we “chunk down,” we get more and more concrete.

So, starting with a person Joe, we might chunk up to see Joe as a male New Yorker, then as a New Yorker, then as an American, then as human, then as a living entity. Or we might chunk down from Joe and examine his face, and then his eyes, and then his left retina, then a single cell in the retina, and so on.

Chunking is a good concept for a manager to understand, because many communication difficulties involve mismatched chunk sizes. Joe may use smaller chunks than Bob and see Bob as vague and kind of sleep-inducing when he talks. Bob may see Joe as terribly boring and “caught up in detail.” In meetings, Joe may get “picky” and slow the meeting down.

As a manager, you probably need to process bigger chunks than your employees and smaller chunks than your manager. Ideally, your employees will learn that you don’t want to hear all the details of their jobs, and you will learn the same about your manager.

It is useful to be able to “chunk up” and “chunk down” when you need to improve communication. When talking to someone using bigger chunks, you can ask the question, “What, specifically?” to get more details. When talking to someone using smaller chunks, you can ask, “What is the intention of this?” or “What is this an instance of?” to encourage larger chunks.

There are many ways to chunk up and chunk down. Frequently, you can get a meeting or discussion back on track by chunking up and then chunking down a different way. We tend to differ less on the bigger-chunk items. Most people in a meeting could agree on such sweeping statements as “We want the company to succeed.” So when there is disagreement, chunking up to a place where people agree can help to defuse the tension and give everyone more context. You can then carefully chunk down, preserving agreement, to develop the details that you need to.

There are two very useful ways a manager can “chunk up” an employee. The employee’s job can be seen in the context of the team and the entire company. Something that may be undesirable or unpleasent to the employee may appear more tolerable when the employee understands its importance in the workings of the entire company. Another way of chunking up a job is to see it in the context of the employee’s career. Ask the employee where she wants to be in five or ten years. Sometimes a job that doesn’t hold a lot of attraction to an employee is a logical step on the path he really needs to travel to reach his career goals. By seeing the job that way, both you and the employee can change your attitudes toward it.

We will have more to say about dealing with unhappy employees in our next column.
musings

I am going to try something different for this column. I am going to pretend to be a journalist. I have no training in journalism per se, although I was given a two-day class in writing for magazines by UNIX World. For this performance, I will actually interview a source.

The person I have chosen to interview is Richard Diemer. Mr. Diemer is a tool-and-die tradesman who has worked for General Motors for 24 years. It was actually my wife who brought him to my attention, after he had shared some of his insights about working in a modern manufacturing plant and how that had affected his life.

Diemer works the swing shift, mid-afternoon to evening for those of you who have never experienced factory life. (I have.) His specialty, tool-and-die maker, was critical to the manufacturing of automobiles. Those graceful fenders, side panels, and hoods get stamped out of pieces of steel when the tool and die are precisely matched. While the design of the fenders and other parts are left to the designers and the engineers, it used to be that tradespersons like Diemer actually kept them working. A new (or worn) die might need a bit of tweaking, just a few millimeters shaved off, and the same few millimeters added to the tool so that the fender comes out perfect.

Today, Diemer watches the process from behind a window. The tools and dies are computer-designed, and he is rarely allowed to touch them. In the cause of efficiency, the smallest possible piece of steel is used for each part, instead of a slightly larger piece that might need some trimming afterward. In the past, if Diemer had noticed that the tool and die were not producing a complete fender (the steel extending to all the edges), he would have taken the stamping press offline while he and fellow tradespersons remachined the tool and die so that it worked correctly. But now he gets in trouble for stopping the process or even getting on the other side of the glass, up close to the computer-driven machinery.

Diemer has six years until he can retire. His job satisfaction is at an all-time low. Any needed machine maintenance is scheduled for breaks and lunchtime, cutting him off from camaraderie with his fellow workers. Once a master of his trade, he is reduced to tending the machines, and his suggestions, based upon over two decades of experience, are mostly ignored.

Sysadmin
You might wonder what this has to do with you. Programmers and system administrators, especially experienced ones, are at the top of the heap, able to demand great salaries and jump between jobs with the greatest of ease. Just like Diemer, who was also at the peak of his experience, until he was marginalized. If it weren't for the union, GM would have fired him, er, downsized him, long ago.

Max Southall <max@pfninfo.com>, the MIS director of Kelme USA Inc., wrote a thoughtful letter that appears in this issue's letters-to-editors section. Here is a snippet of what he wrote to me:

Pretty well the only practical solution I've actually seen implemented by those who've tried to stick by Windows is the hiring of additional and progressively cheaper bodies to try to keep it all going somehow. And concomitantly, the laying off of the fewer more expensive bodies, because, as they say, it doesn't matter how smart you are under the MS scenario, because it takes just as long to reboot yet again. . . . In any case, costs keep rising and the level of service sinking. . . .
We need solutions for managing large numbers of systems, adding users, installing software and patches, and changing configurations – and those tools must be used everywhere.

I just can’t stand the thought of system administration being reduced to carrying a CD fanny pack from user machine to machine, forever. And that’s what’s happened to some of my formerly UNIX colleagues.

I am not simply writing another diatribe about Microsoft. Today, sysadmins carry critical information in their heads about the working of systems and networks. Their experience is great; their tools include Perl and shell scripts. Like Diemer, they are great and talented craftspersons, or they would not be able to do their jobs.

Yet the old way of fixing each problem as if it were unique is doomed. There are not enough talented people to support a world of computers and networks, regardless of whether they are running some version of Windows or of UNIX. The age of fighting fires by creating one-off solutions is passing, and so might the high times for sysadmins. We need solutions for managing large numbers of systems, adding users, installing software and patches, and changing configurations – and those tools must be used everywhere.

USENIX and LISA conferences include solutions to these problems in the papers track. But so far, none of these solutions has worked either well enough or universally enough to be widely accepted. The open-source movement has the potential for creating solutions that will make the future of system administration one of creativity and pleasure, not one of rebooting systems and wearing a fanny pack of CDs.

There will still be a place for shell scripts and Perl. No one system will fit all sizes, and there will be interesting and well-paying work for everyone who is capable. But not if we remain stuck, thinking that the world of the tool-and-die tradesperson will be with us forever. Just ask Richard Diemer.

Perhaps the tools that will save us already exist. A single tool that could securely and reliably distribute files and adjust configurations could do the trick. In the early ’90s, I thought that perhaps Tivoli had the solution, but it was dreadfully slow, it was proprietary, and now it’s also terribly expensive. I have seen several tools, written up in USENIX and LISA proceedings, that seem to come very close. They needed polishing to make them usable by anyone and portable to anything that runs an IP stack.

I am not the person who will choose and promote such a tool. I humbly suggest this as a path that should be taken, as it behooves us not to make ourselves obsolete, replaced by inferior technology, as in Southall’s comments. What he wrote about is real. I had already interviewed Diemer before I received Southall’s email. I have seen the same things coming to pass, and worry about the future of my friends, even the future of civilization.

I believe that there are broader issues here than just system administration. I mean more than just Visual Basic versus real programming. Are we going to design our own future, or just let it happen to us? If we do that, it will be designed to maximize profits, not to the advantage of everyone.

Management
Okay, I am now stepping off my soapbox. I received a book to review about management techniques that I’d like to share with you. In the world we currently inhabit, one of the ways that you make more money as you grow older is to slip into a management position. This is largely a function of human resources, the group that keeps charts describing exactly how much salary each named position may be granted, regardless of reality or market forces.
Management is never easy. We certainly weren’t taught good techniques in school, and some of us chose to work with computers instead of people because it was easier. So when I started rereading this book and listening to the tape while I drove to the airport, I realized that this is the book that could help many people in their dealings with subordinates.

The book is by Adele Faber and Elaine Mazlish. You might not recognize the authors, as they won’t appear in the Business Week top-sellers list. The big focus of the book is learning how to listen and then basing your responses on what you have heard. Listening is in itself a lost art. Most often, people will be planning a comeback rather than really hearing what the other person has to say.

The book is entitled How to Talk So Kids Will Listen and Listen So Kids Will Talk. No kidding. I used this book when learning how to be a parent. I have two grown stepchildren, and when I compare my relationships to my adult children to those of other parents, I know that I have been successful. Other parents have wondered at my success but would not take the time to learn some simple techniques.

One of the techniques involves acknowledging feelings. It is as simple as this. One of your workers comes to you, and he is obviously angry. Instead of explaining to him why you can’t do anything about his problem, you say, “You’re angry.” By acknowledging the anger you get it out into the open; you permit the person his feelings, rather than denying them. As the anger subsides, you can begin to talk about possible solutions. When solutions are really impossible, the book has answers for that as well.

Of course, everything in this book is geared toward children. But we learned our own personal-relationship skills when we were children, and our parents were using the same techniques that were handed down to them by farmers and herdsmen from thousands of years ago. Primitive techniques, in other words. We have not been proactive about social skills any more than we have taken an active interest in shaping our own future.

Communication is another of the problems that Diemer has. His bosses proclaim that they listen to what the line workers have to say, but in reality they don’t listen. I had hoped that the reorganizations of the ’90s were about streamlining management so that the head of a corporation would be closer to those actually doing the work. Boy, was I naive. Reorganization turned out to be lip service (are you listening, George Bush?), not a way of improving communication but, rather, one of increasing profits (and the value of stock options, as well).

There is a cultural tendency for older people to ignore the ideas of younger ones. They are so inexperienced. And the younger ones want to ignore the sage advice of older folk, because they are so rigid in their ways. Or perhaps just afraid of being supplanted, just as they may have supplanted those who came before them. Communication, working as a real team, can make things better.

You can buy the Faber and Mazlish book from Amazon. Note that Amazon carries books that BarnesandNoble.com claims are out of print (for reasons that can only be censorship). My wife ordered a controversial book from Barnes and Noble, and we first got an email saying that they would have to search for the book. Later, we got an email saying it was out of print. Amazon delivered a new copy of this “out of print” book in less than a week.

Buy Amazon. You will be supporting USENIX sysadmins and fighting censorship by a large company that pretends to be very progressive.
While I am on the subject: don’t buy from any Mitsubishi companies until the Mitsubishi Corporation abandons its plans to build a saltworks in the Laguna San Ignacio gray whale nursery (<http://www.nrdc.org>). I think that Mitsubishi makes wonderful large-screen televisions, as well as many other products. But their plans to invade a peaceful cove in Baja California and turn it into an outlet for brine will destroy the last place on the North American West Coast used by gray whales during breeding. Sure, the lagoon makes a handy place to bulldoze drying ponds, but there is no way this activity can be ignored by the whales. I’d like to see some sensibility from Mitsubishi, instead of claims that this activity will be harmless.

On a final note, I finally installed VMware. Several people had written to me about how much they like having it, so I spent the time installing it again. This also involved loading an updated X server, and I am still having problems matching my display and video card, with the side effect of having color-map problems. Still, it is certainly weird to have NT running on a system running Linux. I can scan it, attack it from the network, and I don’t have to have two PCs wasting energy just so I can run both Linux and NT. I’ll have more to say about VMware once I get the kinks out.

And lots of people have written to me about StarOffice as well. Rather nice that Sun bought it, and I hope they make it better and less feature-ridden than its competitor.

---

by Prithvi Rao

Prithvi Rao is the co-founder of KiwiLabs, which specializes in software engineering methodology and Java/CORBA training. He has also worked on the development of the MACH OS and a real-time version of MACH. He is an adjunct faculty at Carnegie Mellon and teaches in the Heinz School of Public Policy and Management.

<prithvi@ux4.sp.cs.cmu.edu>

---

using java

Thread Groups

In a previous article I introduced the use of threads within Java. It is necessary to have a deeper understanding of this topic in order to write serious Java applications. This article presents an introduction to the ThreadGroup class. This class serves to organize threads. In a limited sense, it is analogous to the concept of databases, in which semantically similar pieces of information are grouped in a common repository. Naturally, another such example is a directory (analogous to the thread group) and an individual file (a thread).

**Thread Group Characteristics**

A thread group can contain a group of threads or can contain other thread groups that contain threads (similar to directories, which contain subdirectories, which contain files). The resulting thread group hierarchy is a tree structure.

One of the key characteristics of thread groups is that it is possible to affect the state of all threads in the hierarchy with a single call. For instance, it is possible to stop every thread in the thread group with a single call. This is one reason to use thread groups.

Consider the following code:
ThreadGroup A = new ThreadGroup('A');
ThreadGroup B = new ThreadGroup(A, 'B');
ThreadGroup C = new ThreadGroup(A, 'C');
ThreadGroup D = new ThreadGroup(C, 'D');
ThreadGroup E = new ThreadGroup('E');

The resulting hierarchy is as follows:
From main, A and E are descendants. ThreadGroup B is a descendant of A, and so is ThreadGroup C. Finally, ThreadGroup D is a descendant of C.

One way to improve the performance of an application utilizing thread groups is to use a single call to affect the state of threads in a hierarchy. If this were not possible, then each branch of the tree would have to be traversed, and this could be a time-consuming operation.

Another application for thread groups is in a multiprocessing environment. A given CPU may run threads that are in a given group (this needs operating-system support, however), and it is possible to assign different priorities to the different thread groups depending on the application. Generally, however, this requires significant operating-system support not usually found on nonrealtime systems.

Although at the time of this writing there are not too many examples of JVMs that have been ported to running on realtime operating systems, this is likely to happen, given that modern-day audio and video applications must meet strict time deadlines.

The main Thread Group
Thread groups are organized by name. Each thread group must have a unique name. All ThreadGroup constructors take a name as an argument. The default thread group in a Java application is main.

When you start an application, the thread group is main. Unless otherwise specified, all threads will be created as part of the main thread group. If you run applets within a browser, the name of the root thread group will depend on the browser.

Creating a new thread without specifying a thread group in the thread's constructor places the thread in the same thread group as the creator.

Thread Constructors
The following four thread constructors create the thread in the current thread group.

Thread()
Thread(Runnable)
Thread(String)
Thread(Runnable, String)

The following three thread constructors create the thread in a specific thread group.

Thread(ThreadGroup, Runnable)
Thread(ThreadGroup, String)
Thread(ThreadGroup, Runnable, String)

It is possible to learn to which thread group a particular thread belongs by using the getThreadGroup() method.

ThreadGroup Z = foo.getThreadGroup();

It is also possible to enumerate threads within a particular thread group. Consider the following:

ThreadGroup X = Thread.currentThread().getThreadGroup();
int numThreads = X.getActiveThreads();
It is possible to limit the priority of any thread that is inserted into a thread group.

**Limiting Priorities**

It is possible to limit the priority of any thread that is inserted into a thread group. The call to `setMaxPriority` will enforce an upper limit of the thread group as a whole. The following example demonstrates the use of `setMaxPriority`:

```java
Thread threads[] = new Thread[numThreads];
X.enumerate(threads);
for (int n = 0; n < numThreads; i++)
    actonthreads(threads[i]);
```

The threads `Y` and `Z` are usually created with the default priority of `NORMAL_PRIORITY`, which is equal to 5. In the example above, before `Z` was created the maximum priority was set to `MIN_PRIORITY+2`, which is now 3. So the creation of `Z` results in its priority value of 3.

**Thread Groups and Priorities**

All thread groups that are descendants of `ThreadGroup` will be affected by a call to `setMaxPriority`.

Attempting to set the priority of a thread higher than the priority of the thread group to which it belongs will result in `SecurityException` being thrown.

Once the maximum priority of a thread group has been lowered, it cannot be raised.

The maximum priority of the system `ThreadGroup` is `MAX_PRIORITY` (10). The maximum priority of the applet `ThreadGroup` is 6.

**Thread Groups and Security**

The `Thread` and `ThreadGroup` classes both have a method called `checkAccess()`, and they both call the Security Manager's `checkAccess()` method. The Security Manager checks to see whether the threads are permitted to gain access to certain operations. If there is a violation, the Security Manager throws an exception (`SecurityException`), otherwise returns. Consider the following:

```java
void checkAccess(Thread t)
    checks if the current thread is allowed to modify the state of the thread.

void checkAccess(ThreadGroup g)
    checks if the current ThreadGroup is allowed to modify the state of the ThreadGroup g.
```

Most of the methods in the `Thread` class and the `ThreadGroup` class call `SecurityManager` before performing the actual operation. For example, the implementation of `stop()` reveals the following:

```java
public void stop()
{
    SecurityManager V = System.getSecurityManager();
    if (V != null)
        V.checkAccess(this);
    stop(new ThreadDeath());
}
```

The Java Security Manager implements security on an “all-or-none” basis. In other words, there is no granularity to distinguish allowable operations. If the Security
Manager does not permit a thread to suspend another thread, then it also does not allow the same thread to set the other thread’s daemon status. The setDaemon() function changes the daemon status of the thread group. It does not alter the daemon status of any individual threads within the group. If a group is made a daemon group, then it will be destroyed automatically when all of the threads it contains are terminated.

The following is a list of ThreadGroup methods that call ThreadGroup’s checkAccess() method:

```
ThreadGroup(ThreadGroup parent, String name)
setDaemon
setMaxPriority
stop
suspend
resume
destroy
```

The following Thread methods call checkAccess():

```
stop
suspend
resume
setPriority
setName
setDaemon
```

Recall that a standalone Java application has no security manager, so threads can modify and inspect any other thread. Within an applet, a thread can manipulate another thread only if both threads are members of the applet’s ThreadGroup. A thread cannot manipulate a thread that is in another applet.

**Conclusion**

We have presented the use of thread groups and their interaction with the Security Manager in Java. The use of thread groups is critical to writing advanced applications in Java. However, the programmer must be aware of the differences in their interaction with the Security Manager depending on whether the threads run in a standalone program or as applets.

In future articles I will present applications using ThreadGroup to further demonstrate the use of this important Java class.

---

The Java Security Manager implements security on an “all-or-none” basis. In other words, there is no granularity to distinguish allowable operations.
Perl and SQL Databases: A Tasty TiDBIt

For years, programmers have used text files as databases. UNIX is rife with examples: the passwd and group files, for instance, as well as many others in /etc. Text files work well for small amounts of data—a dozen rows or perhaps a hundred or more—but become cumbersome at larger sizes. They’re slow to access, they can’t be written simultaneously by multiple users, and they’re tedious to edit. The lack of any inherent structuring in their contents also limits the usefulness of text files.

If you’re keeping a database in text files, and things aren’t working out, the obvious alternative is a “real” database, which nowadays means an SQL database. (There are a few intermediate alternatives, like DBM files, but not many problems fit that niche.) However, in years past, an SQL database wasn’t an attractive solution for an everyday problem. Database servers were expensive and not really designed with small-to-medium-sized chores in mind.

But all this has changed! If you are working on a standard UNIX platform, you can build and install any one of several open-source SQL database servers in an hour or two. Even better, you can talk to it directly with Perl through a straightforward “DBI” interface. Nowadays, using SQL databases from within your Perl scripts isn’t just possible—it’s a good idea.

DBI and DBD

The DBI module is a “database-independent interface” to many different SQL-based databases. Mainstream commercial products like Oracle, Informix, and Sybase are well supported. However, for the purpose of this column I’m going to focus on MySQL, which is a well-known, high-performance, open-source alternative.

Each different database has its own DBD (Database Driver) module. Oracle has DBD::Oracle, Sybase has DBD::Sybase, MySQL has DBD::mysql, and so on. Each DBD provides an interface between the corresponding database client library and the database-independent DBI module.

We’ll use two different DBDs in the examples that follow: DBD::MySQL and a simpler alternative called DBD::CSV, which applies the DBI interface to text files in CSV (Comma Separated Value) format. The examples assume that you know some basic SQL. If you don’t happen to know any SQL, there are many good books on the topic—one of my favorite introductory texts is The LAN Times Guide to SQL. You can also find some SQL tutorials on the Internet.

Installing MySQL

Obtain a copy of the MySQL source tarball from one of the mirrors pointed to by <http://www.mysql.com>. Unpack it, then run configure and make as directed in the installation instructions. You may want to install it underneath your home directory using the --prefix option to configure. You could also skip the build process (it takes half an hour or so on a moderately fast single-user machine) and use a binary tarball instead. Either way, when you have it built and installed, you have to initialize the grant tables with the mysql_install_db command:

```bash
% scripts/mysql_install_db
```
You can now start the MySQL server. Because we’re just playing around with it, let’s run it on a different port and socket for now:

```bash
% setenv MYSQL_TCP_PORT 4001
% setenv MYSQL_UNIX_PORT /tmp/mysql.login.sock
% scripts/safe_mysqld &
```

With the server running, create a database called “test_foo,” which we’ll use later:

```bash
% bin/mysqladmin -p create test_foo
Database "test_foo" created.
```

You can see how the server is doing by running the MySQL client. (If you do this later, you’ll need the environment variables set.) Try the status command:

```bash
% bin/mysql
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 12 to server version: 3.22.27
...
mysql> status
-- -- -- -- --
bin/mysql Ver 9.36 Distrib 3.22.27, for sun-solaris2.7 (sparc)
...
```

That’s all you have to do for now. This installation has no security, but that’s a problem you can resolve later if you decide to use MySQL for real.

### Installing DBI

The following (abbreviated) instructions assume you have full administrative control over a Perl installation. It doesn’t have to be your machine’s “main” installation. If you want, build Perl in your home directory or some other convenient location before proceeding.

First, fire up the CPAN shell from the appropriate copy of Perl:

```bash
% /wherever/my/local/bin/perl -MCPAN -e shell
```

You may have to configure CPAN if this is your first time using it. Make your life easier by setting the `prerequisites_policy` config variable to follow. Once in the CPAN shell, verify that you can find the DBI module:

```
cpan> i DBI
Bundle Bundle::DBI (TIMB/DBI-1.13.tar.gz)
Module DBI (TIMB/DBI-1.13.tar.gz)
```

If so, go ahead and build and install it:

```
cpan> make DBI
... output omitted ...
cpan> install DBI
... output omitted ...
```

So far, so good. Now, install some DBDs. First do DBD::CSV. (We’ll do the Text::CSV_XS module, which is a prerequisite, first.)

```
cpan> install Text::CSV_XS
cpan> install Bundle::DBD::CSV
```

Even if you haven’t got MySQL working you’ll be able to get a feel for DBI with DBD::CSV. Speaking of which, to build the MySQL DBD:

```
cpan> make Bundle::DBD::mysql
```

When you are asked which database to install support for, answer “1” for MySQL only (unless you also happen to have mSQL installed). When asked for the host and port, if
The `connect` method returns a database handle, which we store in the variable `$dbh`. Database handles represent active connections.

you are running MySQL on an alternative port as suggested above, respond with “localhost:4001” (or whatever value you used). Assuming the make went smoothly, test and install the MySQL DBD:

```
$ cpan> install Bundle::DBD::mysql
```

NOTE: The DBD bundles reinstall DBI, at least in some cases. This is normal, if seemingly boneheaded.

So It's Installed – Now What?
Let’s use DBI as the basis for a simple mail-filtering application. Our eventual goal will be to create a program that parses a mail message and returns a nonzero status if the message is known to come from an “approved” address, or a nonzero status otherwise. A program like this can be used by a delivery agent to accept or bounce incoming email, or at least to divert “unapproved” messages into a different folder. We’ll determine whether an originating address is approved by looking up the sender’s host in a database.

We’ll start with a very simple schema consisting of a single column, HOST, containing approved host names. To make this even simpler, let’s start with DBD::CSV. Here is a Perl program that will “connect” to the “database” (really it’s just a bunch of files) and create a table for us:

```perl
use DBI;
my $dbh = DBI->connect("DBI:CSV:f_dir=csv")
or die 'couldn’t connect';

$dbh->doq(
    CREATE TABLE APPROVED_HOST ( 
    HOST CHAR(128) 
    )
) or die;
```

The argument to the `connect` method is the “data source” (DSN) string. This tells DBI which driver to use (CSV in this case). It also supplies additional arguments that are passed into the driver itself. In this example, we’ve supplied the argument `f_dir=csv`, which instructs the CSV driver to create its text files in the subdirectory `csv`. If `connect` fails, it will return false, and we die because there is no particular point in continuing. The `connect` method returns a database handle, which we store in the variable `$dbh`. Database handles represent active connections.

The `do` method is one of several ways of executing SQL statements. It takes a string and passes it to the driver for execution. Again, it returns true or false indicating success or failure, respectively. Note that we’ve quoted the argument to `do` with the generalized single quote syntax `q()` – this isn’t strictly necessary, but it makes the code easier to read.

After this program runs, the `csv` directory will contain a file named APPROVED_HOST, named after the APPROVED_HOST table. It won’t contain anything other than a single line with the name of the table’s (single) column, HOST, but we’ll fix that in a moment.

Now, let’s write a program, called `approve`, to insert an approved host name in the table. This is also straightforward:

```perl
use DBI;
my $dbh = DBI->connect("DBI:CSV:f_dir=csv")
or die 'couldn’t connect';

my $host = shift or die "usage: approve host\n";
```
$dbh->do(q{INSERT INTO APPROVED_HOST VALUES (?)),
  undef, $host);

Use approve like this:

% approve foo.bar.com

Here we are using the multi-argument form of the do method. The second argument is a hashref of "attributes" that isn't often needed (just put undef in it). The remaining arguments are "bind values" that are bound to placeholders in the SQL argument. Each question mark in the first argument is a placeholder. When DBI executes the SQL statement, it replaces the placeholders in it with their corresponding bind values (SQL escaping them in the process). In this example, there is a single placeholder (the value in the INSERT statement) and a single bind value that gets plugged into it (the $host variable).

Our last example is a program called ok, which prints "yes" or "no" depending on whether or not its argument is an approved host:

use DBI;
my $dbh = DBI->connect('DBI:CSV:foo_dir=csv'
  or die 'couldn't connect';
my $host = shift or die 'usage: ok host
';
($h) = $dbh->selectrow_array(q{
  SELECT * FROM APPROVED_HOST WHERE HOST = ?
}, undef, $host);
if ($h) {
  print "yes\n";
} else {
  print "no\n";
}

The selectrow_array method is a convenient way to run an SQL query statement when you need only the first row of the result. The row is returned as a list.

Connecting with DBD::mysql

Let's rewrite the programs above to use DBD::mysql. We'll start with the program to create a table. The only change that's absolutely necessary is the connect method:

my $dbh = DBI->connect('DBI:mysql:database=test_addr;'
  'mysql_socket=/tmp/mysql.login.sock'
  or die 'couldn't connect';

The first part of the DSN string has changed from DBI:CSV to DBI:mysql. The rest of the DSN string is DBD-specific. The MySQL DBD allows quite a few different options. By default it connects to a MySQL server running on the local host through a UNIX socket. Because we started the server on a different (nonstandard) socket, we have to specify a value for mysql_socket. Setting the MYSQL_UNIX_PORT variable would also work, as would using a config file.

Other optional arguments for the connect method include user and password. We're using the defaults, which is fine for our test database.

Let's change the schema while we're at it. We'll make HOST a primary key, and add some DATETIMEs so that we can keep track of when approvals are created and expire them after a period of time.
The call to `disconnect` is a very good idea and avoids inconsistent operation and warning messages. To save space, though, I won’t always show it. Next, let’s look at a revised version of the `approve` program:

```perl
$dbh->do(q DROP TABLE IF EXISTS APPROVED_HOST);
$dbh->do(q(  CREATE TABLE APPROVED_HOST (    HOST VARCHAR(128) PRIMARY KEY,    APPROVED_DATE DATETIME NOT NULL,    EXPIRE_DATE DATETIME NOT NULL  ) ) or die;
$dbh->disconnect;
```

The call to `disconnect` is a very good idea and avoids inconsistent operation and warning messages. To save space, though, I won’t always show it. Next, let’s look at a revised version of the `approve` program:

```perl
use DBI;
use POSIX;
my $dbh = DBI->connect #... as before
my $host = shift or die "usage: approve host\n";
my $now_td = strftime("%Y-%m-%d", localtime);
my $later_td = strftime("%Y-%m-%d",
    localtime(time+24*60*60*180));
$dbh->do(q INSERT INTO APPROVED_HOST
    (HOST, APPROVED_DATE, EXPIRE_DATE) VALUES (?, ?, ?)
), undef, $host, $now_td, $later_td);
```

The `localtime` and `POSIX` `strftime` functions are handy when converting UNIX times to formats that can be understood by databases. I insert a “now” date as well as a date 180 days in the future. The dates are in “YYYY-MM-DD” format, which is readily understood by both humans and MySQL. Next, the `ok` program:

```perl
# use statements and connect omitted ...
my $host = shift or die "usage: ok host\n";
my $now_td = strftime("%Y-%m-%d", localtime);
($count) = $dbh->selectrow_array(q(  SELECT COUNT(*) FROM APPROVED_HOST  WHERE HOST = ? AND EXPIRE_DATE > ?
), undef, $host, $now_td);
if ($count) {
    print "yes\n";
} else {
    print "no\n";
}
```

This works like the previous version of `ok`, except that it also checks to see that the approval hasn’t expired, and it uses a count of the matching rows (there should be only one anyway). Next, let’s look at a program called `approved` that lists all the currently approved hosts:

```perl
# use statements and connect omitted ...
my $now_td = strftime("%Y-%m-%d", localtime);
$sth = $dbh->prepare(q(  SELECT HOST, EXPIRE_DATE FROM APPROVED_HOST  WHERE EXPIRE_DATE > ?  ORDER BY HOST ));
$sth->execute($now_td);
my ($host, $expire_date);
while (($host, $expire_date) = $sth->fetchrow_array) {
    printf "%30s expires $expire_date\n", $host;
}
This program is the first we've looked at that uses a query that will return multiple rows. There are several ways of working with such queries. In general, you first "prepare" the SQL statement into a statement handle. Then you execute the prepared statement and iterate over the rows in the result. The `prepare` method returns a statement handle object ($sth in this case). After calling the `execute` method on the statement handle, we read the resulting table with the `fetchrow_array` method. There are a number of alternative ways of handling query results -- see the DBI documentation for more information.

Our last program, which accomplishes the promised task of "approving" mail messages, requires that you have the `Mail::Internet` bundle:

```perl
use DBI;
use POSIX;
use Mail::Internet;
use Mail::Address;
my $dbh = DBI->connect("..."); # as before

my $noTd = strftime("%Y-%m-%d", localtime);
my $mail = Mail::Internet->new("STDIN") or die 'can't parse message';
my ($from_a) = Mail::Address->parse($mail->head->get('From'));
my ($host) = $from_a->host;
my ($count) = $dbh->selectrow_array("SELECT COUNT(*) FROM APPROVED_HOST WHERE HOST = ? AND EXPIRE_DATE > ?", undef, $host, $noTd);
$dbh->disconnect;
exit($count ? 0 : 1);
```

We read the message from standard input, then use a few lines of `Mail::Internet` voodoo to extract the host name from the From line. Then we look up the host name in the database and return an exit status of 0 or 1 depending on whether or not it is approved.

**Databases: Free and Easy!**

There are many more details to consider in a production version of this system -- error handling, for example. But that'll have to wait for a future column.

Meanwhile, I hope that with these examples I've shown you that nowadays SQL databases are both inexpensive (free!) and easy to use. If you need a safe, organized place to store some data -- no matter whether you have a little or a lot -- consider doing it with Perl and an SQL database.
java performance

Performance Issues with the Java Native Interface

The Java Native Interface (JNI) is a mechanism in Java that allows a Java program to call functions in other languages such as C++. A variety of issues come up with JNI use, including some performance ones, and it's instructive to step through an example and look at some of these.

Summing the Values in an Array

Suppose that we have a Java program that calls a C++ function to sum the values in an array, with -1 used as an array terminator value. The Java program looks like this:

```java
public class sum {
    static {
        System.loadLibrary("clib");
    }
    public static native int sum(int arr[]);
    public static void main(String args[]) {
        int x[] = new int[5];
        x[0] = 37;
        x[1] = 47;
        x[2] = 57;
        x[3] = 67;
        x[4] = -1;
        int y = sum(x);
        System.out.println(y);
    }
}
```

A method called from Java but defined in some other language has a "native" modifier in the declaration, and the method has no body (because the body will be supplied by the other language implementation).

Native methods are dynamically linked, and their implementations are found in a shared library or DLL. `System.loadLibrary()` is called to load the shared library. This loading is done when the Java program starts up (enclosing Java code in "static (...)" has this effect). The use of dynamic linking implies some performance overhead.

Once the Java program is defined, it is compiled by saying:

```
$ javac sum.java
```

If the program is run at this point, an `UnsatisfiedLinkError` results, because the shared library that defines `sum()` has not yet been created.

Building the Shared Library

The first step in building the library is to generate a header file that declares the C++ `sum()` function. One way of generating the file is:

```
$ javah -jni sum
```

using Java Development Kit 1.2 commands. The result is a file that looks like this:

```c
#include <jni.h>

JNINATIVEEXPORT jint JNICALL Java_sum_sum(JNIEnv*, jclass, jintArray);
```
This is the declaration of the C++ function we need to implement to sum the elements of the array.

An actual implementation of the `sum()` function is:

```c
#include <jni.h>
extern "C" {
JNIEXPORT jint JNICALL Java_sum_sum(JNIEnv* env, jclass, jintArray arr)
{
    jint sum = 0;
    jint* p = env->GetIntArrayElements(arr, 0);
    for (jsize i = 0; p[i] != -1; i++)
        sum += p[i];
    env->ReleaseIntArrayElements(arr, p, 0);
    return sum;
}
}
```

The `extern "C"` is a C++ notation that says that the enclosed function should have C name linkage rather than use C++-style external names. Once we've defined this function, we compile it and create a shared library containing it. For example, using C++Builder 4, we would say:

```
$ bcc32 -c -Ip:/javanew/include -Ip:/javanew/include/win32 clib.c
$ bcc32 -tWD clib.obj
```

to create `clib.dll`. We can then run the Java program:

```
$ java sum
```

and it will print out a value of 208.

### Details of How Sum() Works

If we go back to the C++ implementation of `sum()`, there are several points of interest. One basic issue is how Java and C/C++ differ in the way arrays are treated. In C/C++, an array is simply a contiguous region of storage. In Java an array is more complicated. It fits within the class-object hierarchy, so you can assign an array reference to an object reference. Java arrays have their length stored with them, which can be retrieved at any time by saying:

```java
int len = arr.length;
```

Array subscripts are checked at runtime for validity, with an exception thrown if they're out of range. Java has no pointers, and Java array values can be referenced only through subscripts or by using the reflection mechanism, so the Java runtime system is allowed flexibility in the way it stores arrays. For example, it's possible that the Java garbage collector might move an array to a different memory location. This is fine if you're using only Java, but it doesn't work at all if you've obtained a C-style pointer to a Java array and then it moves on you.

To solve this problem, a function such as `GetIntArrayElements()` may return an actual pointer to the Java array, if the Java garbage collector can guarantee that the array will not move, or else it will copy the array into a temporary location and return a pointer to the copy.

A final point about `sum()` is that a certain level of abstraction is implied by the JNI interface. For example, a Java array structure includes the actual elements, along with...
the length of the array, JNI does not provide access to the raw runtime array descriptor, but, rather, provides functions to obtain information about arrays. This abstraction is safer and more portable than using a lower-level interface.

**Exception Handling**

We said earlier that Java guarantees that array subscripts will be checked at runtime, but the implementation of `sum()` above does not honor this guarantee. What happens if the user fails to terminate the array value sequence with `-1`?

To fix this problem, we can rewrite the C++ code as:

```cpp
#include <jni.h>
extern "C" {
JNIEXPORT jint JNI_CALL
Java_sum_sum(JNIEnv* env, jclass, jintArray arr)
{
jint sum = 0;
jint* p = env->GetIntArrayElements(arr, 0);
ssize maxlen = env->GetArrayLength(arr);
jsize i = 0;
while (i < maxlen && p[i] != -1)
    sum += p[i++];
if (i == maxlen)
    jclass exc =
        env->FindClass("java/lang/ArrayIndexOutOfBoundsException");
if (exc != NULL)
    env->ThrowNew(exc, "thrown from C++");
env->ReleaseIntArrayElements(arr, p, 0);
return sum;
}
}
```

The subscript is checked before each array access. If the subscript overflows before `-1` is found, an exception is thrown and propagated back to the Java program. If the Java program is run and `sum()` is called with an invalid array, the result is:

```
java.lang.ArrayIndexOutOfBoundsException: thrown from C++
at sum.sum(Native Method)
at sum.main(sum.java:14)
```

**Summary**

The Java Native Interface is quite useful in accessing bodies of code written in other languages. We’ve illustrated some of the performance issues that come up with use of JNI. Additional material on JNI, including performance considerations, can be found in the book *The Java Native Interface* by Sheng Liang (Addison-Wesley, 1999).
This month I want to look at only four books: one truly important one and three others. The areas covered are open source, languages, and protocols.

OPEN, SESAME!
Beginning in 1992, Eric S. Raymond has jotted notes and comments that were (and are) Net-accessible. Since 1996, several of his essays (most notably “The Cathedral and the Bazaar”) have become required reading. If anything, the obloquy heaped on Raymond by the PR folks in Redmond, WA (e.g., in the “Halloween documents”) has made him more important.

O’Reilly has done us all a favor by collecting a number of Raymond’s pieces and making them readily accessible at a price everyone can afford.


These are the Common Sense or the Federalist Papers for the Open Source movement: they are the testimony of just why the BSDs and Linux, Perl and Python, Tcl and (even) Java are successful; these facilities have tens of thousands of programmers all over the world contributing to the excellence of programs and systems. They don’t have an encapsulated proprietary system that no one can debug.

When I was writing A Quarter Century of UNIX (Addison-Wesley, 1994), I realized that essential to the "UNIX philosophy" was something that was alien to commercial programming: the changes to the kernel, the applications, the programs were all written by one or two or three hackers — not by teams of programmers. Eric Allman wrote sendmail; Mike Lesk wrote the original uucp (and even the mid-1980s version, HoneyDanBer, was by Peter Hemenway, Dan Nowitz, and Brian Redman); Steve Johnson wrote yacc; Bill Joy wrote vi; etc. Brian Kernighan once told me that Awk was the toughest project he ever worked on "because there were three of us" (Aho, Weinberger, and Kernighan).

Of course, it’s all the Internet’s fault. Even with the USENIX tape-swap and uucp, stuff passed about more slowly. It’s the Net that enabled a Finnish student to send his work to nearly every corner of the world and currently enables thousands to contribute and feed stuff back to the Linux and GNU and Perl communities.

In some ways, “The Magic Cauldron” is my favorite essay of Raymond’s. Here he shows that he understands the underlying economic reasons for the success of open software. This understanding is based on the anthropological study of gift-exchanging and on the concepts of what happens in a gift culture, when “survival goods are abundant" and therefore the exchange is no longer interesting.

This is tied together with the notions inherent in the fact that software has two distinct values: use value and sales value. As Raymond says, use value is value as a tool; sales value is value as a saleable good. One of Raymond’s most interesting discussions is founded in this.

Food, equipment, books all retain value independent of the producer. If a farmer sells his farm, the food produced retains its value, etc. When a computer manufacturer (hardware or software) goes out of business or a line is discontinued, the price users are willing to pay plummet. The price users will pay is limited by the expected future value of vendor service.

Open-source software forces the vendor into a world of service-fee-domination and exposes "what a relatively weak prop the sale value
of the secret bits in closed-source software was all along."

The true advantage for all of us lies in the notion of high-quality software being built upon by the community, rather than being locked up in a vault or discontinued. (See my "20 Years Ago . . . " in this issue.)

Raymond believes that in 2000/2001 Linux will be "in effective control of servers, data centers, ISPs, and the Internet, while Microsoft maintains its grip on the desktop." Most likely that's correct. But with the advent of products like StarOffice and WordPerfect for Linux, there may well be inroads into the desktop market as well.

This is a fine, thought-provoking book that should be read by anyone interested in computing: open, academic, or proprietary.

TICKLE?

Nelson's Tcl/Tk: Programmer's Reference is a very fine example of just what a language reference should be. Following a mere 20 pages of Tcl and Tk basics, Nelson launches into a brief preface on syntax and then the Tcl/Tk commands from after through 10 pages on van. I found all the information both compact and appropriate. The remainder of the book contains an appendix on options and a rather good index.

I do have one nit, but it's not with Nelson, rather with Osborne/McGraw-Hill, the publisher. I may be getting old and my sight getting worse, but I find the sans-serif font used for arguments impossible: under eof (p. 121), for example, we find channelId and the "I" and "1" are indistinguishable. As "channelId" occurs frequently, I found this annoying.

The text in serif type is quite clear, but the confusion was unnecessary.

PROTOCOLS

Johnson's book on mail protocols is quite a respectable one, especially once you get past his paragraphs on history. The chapter on IMAP and on filtering were informative, though it's unclear to me why there's no mention of procmail. There's no mention of sendmail save as an MTA, either, and none at all of Eric Allman (though Ray Tomlinson and [in the appendix on languages] Larry Wall and Guido van Rossum are there).

Actually — though that appendix talks about C, C++, Java, Perl, Python, and Emacs — Ritchie, Thompson, Stroustrup, Gosling, and Stallman are notable in their absence.

The meat of this book is quite good, but the lacunae are quite striking.

On the other hand, the CD is just terrific! It contains all the RFCs actually mentioned in the book.

JAVA ALL OVER THE PLACE

Distributed Programming with Java has the weaknesses of Johnson's volume (paltry bibliography, no credit to most important authors/programmers), but it has its advantages, too. You may not be able to find James Gosling nor Bill Joy nor Ken Arnold nor Jim Waldo nor Ann Wollrath, but who cares who did the work, anyway?

Mahmoud has organized his text well, and the organization of the chapters is far better than average. But there are just about two pages on internetworking, with no mention of IPv6.

The introductory material on sockets programming is fine, as are others. But, all in all, for a book published in 2000, that's just not good enough.

Oh, yeah. Chris Nelson does mention John Ousterhout, as well as Brent Welch and many others. Perhaps some folks do care.

Craig Larman

Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design


Reviewed by Cuy Flint

Craig Larman's Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design may not be the best book on object-oriented design ever written, but it's the most useful book I've found for the practicing programmer. I've been studying and doing OO analysis and design for 10 years, and I found a new insight or a statement that crystallized my gut-level understanding on almost every page.

Larman explains the theory of object-oriented analysis and design (compared and contrasted to the structured-design methods that most of us are already familiar with). He discusses the mechanics of diagramming systems with UML and he offers the iterative-development process to follow when developing an OO project.

The book uses a very nuts-and-bolts approach to OO design. The text intermingles introducing the concepts of OO analysis and design with a sample OO design for a point-of-sale terminal. There are several tables of items to consider when defining objects and classes during the different phases of developing a project, many short examples of design details, and lots of practical advice.

The book is well-written. Larman expertly treads the fine line between wasted words and being too terse. The text and diagrams complement each other well and make the abstract concepts easy to apply. At over 500 pages, this book delivers a lot of information while still being short enough to comprehend.
A New Editor

by Nick Stoughton

USENIX Institutional Representative

<nick@usenix.org>

Introductions

by David Blackwood

Standards Reports Editor
<dave@usenix.org>

First of all, many thanks to outgoing editor Nick Stoughton for all his contributions to this column over the years. We are sure to hear from Nick as the author of many more articles, though, as he continues in his role of USENIX Institutional Representative.

As this is my first column as Standards Reports ("snitch") Editor, please allow me to introduce myself. Although I am only a one-time contributor to this column, I have been active in standards for over ten years. I have been a member of the Canadian POSIX Working Group (equivalent to the US WG15 Technical Advisory Group, or TAG) since 1989 and chair of the group and head of the Canadian delegation to WG15 since 1995. Professionally, I have been an in-house systems integrator with the Government of Canada and have worked with UNIX systems since 1985. I am also a sometime convenor of the Ottawa Carleton UNIX Users' Group.

This column welcomes dialog with you, the readers. Please send your comments to <dave@usenix.org>. Your contributions to this column in the form of articles are both welcome and requested. You may note that my title is editor, not author. I will be relying on those who are participants in or observers of standards activities relevant to advanced-computing-systems users to help keep the rest of us informed of what is happening. Whether the concern is the various POSIX committees, ISO, ITU, or IETF, all contributions are welcome.

There are two major issues facing standards-development organizations today. The first issue is one of declining participation. Typically, standards are developed by groups of volunteers working on their own (or their employers') time and expense. As organizations focus more on their core business, participation in standards development often falls by the wayside. I believe this is very shortsighted and happens all too often. The second is one of funding. Standards-development organizations have for many years funded their operations through the sale of printed standards. Today many people insist that for standards to be truly open and widely implemented they must be freely available on the Net. Standards-development bodies therefore need to find alternate sources of revenue if they are to survive. Without standards-development bodies to guide the process and publish the results, and without volunteers to do the work, there will be no more standards. Some may argue that this would be no great loss, but you do not have to look very far to see the value of standards in daily life. Take our telephone and electrical systems, both prime examples of the kind of interoperability that standards can enable. If you have thoughts on how to address these or other issues, write to me.

In June 1999, Compaq Computer Corporation announced that it was ceasing publication of the "Open Systems Standards Tracking Report." First published in 1989 by Digital Equipment Corporation as the POSIX Tracking Report, it was retitled in 1992 to reflect a change in focus. Its stated purpose was to "stimulate discussion, inform, eduscate, and raise the importance of standards-related issues." Its loss leaves a void I hope this column will be able to fill.

The IEEE Standards Association has recently approved two new Project Authorization Requests (PARs). The first is P1003.1s (C/PA) Standard for Information Technology –

At the same time it announced that 1003.1d (C/PA) Standard for Information Technology – Portable Operating System Interface (POSIX®) – Part 1: System Application Program Interface (API) – Amendment d: Additional Realtime Extensions [C Language] had been approved.

Complete details of the status of all POSIX projects are at <http://www.pasc.org/standing/sd11.html>.

For this month, I will leave you with a set of bookmarks where you can learn more about the POSIX standards and the various participants and players in the process.

American National Standards Institute (ANSI) <http://wwwansi.org/>>

British Standards Institution (BSI) <http://wwwbsiorg.uk/>>

Dansk Standard (DS) <http://wwwds.dk/>>

Deutsches Institut für Normung (DIN) <http://wwwdin.de/>>

Information Technology Standards Commission of Japan (ITSC) <http://wwwitscjp organizipjorg/eg/>>

The Institute of Electrical and Electronics Engineers, Inc. (IEEE) <http://wwwieeeorg/>>


Portable Application Standards Committee (PASC) <http://wwwpascorg/>>

International Electrotechnical Commission (IEC) <http://wwwiecorg/>>

International Organization for Standardization (ISO) <http://wwwisoorg/>>

Joint Technical Committee 1 (JTC1) <http://wwwjtc1org/>>

Sub-Committee 22 (SC22) <http://wwwdkuugdkjtc1sc22/>>

Working Group 15 (WG15) <http://wwwdkuugdkjtc1sc22wg15/>>

Nederlands Normalisatie-institut (NNI) <http://wwwnini.nl/>>

The Open Group (TOG) <http://wwwopengrouporg/>>

The Austin Common Standards Revision Group <http://wwwopengrouporgaustin/>>

Standards Council of Canada (SCC) <http://wwwscca/>>

POSIX Revision Project: Austin Group Status Update

Andrew Josey, Austin Group Chair <a josey@opengrouporg/>>

This is a brief status update after completion of Draft 2 of the Austin Group specifications – the joint project to revise the POSIX and Single UNIX Specifications.

Draft 2 was made available on October 29, completing almost four staff months of editorial work since the Montreal meeting in July. The page count for Draft 2 now totals 2,740. The Draft 2 specifications are available from the Austin Group Web site, <http://wwwopengrouporgaustin/loginhtml/>>.

This draft has attempted to resolve many of the style issues raised against Draft 1, and I am pleased to report that recent discussions with the editorial team at IEEE and ISO have indicated that the manual-page format, long a significant issue, is acceptable. This should be a major benefit for the end reader of the standard.
The Times, They Are a’Changin’

by Andrew Hume
President, USENIX Board
of Directors

Does it seem like the world is changing too fast to you? It seems trite, but every day brings fresh reminders of the increasing pace of life in general, and our computer world in particular. I'm still surprised when my 11-month-old daughter notices more hair than the day before. And I am in awe of the perceived value of Internet stocks, and that AOL was able to buy something real with its funny money. And although I thought I was inured to disk price sticker shock, I was (very pleasantly) surprised to see how much disk storage costs now.

I believe that cheap disk is far more an enabler of the Webification of everything than CPU speed, and is only expected in importance by consumer access speed. Three years ago, one of my projects bought commercial RAID 5 storage at $300K/TB. I just bought a 4U high rack-mounted RAID 5 box yielding 350GB at a rate of about $42K/TB. In six months, I expect the cost to dip below $30K/TB (and this is RAID 5 storage, not raw disk). Just amazing.

USENIX, too, is changing. Over the last few years, our membership has changed in both job distribution and interests. Over 50% of our membership self-identify as sys admins of one flavor or another. There is much more interest in Web-based computing (our USITS and NETA conferences are good examples) and in UNIX-style environments for commodity PC hardware (such as the var-

ious *BSD systems and Linux). In fact, this year we expect to form a second STG (SAGE is our first) focusing on Linux. (We’ll form more if you want one!)

As a result of these forces, the USENIX Board has been trying to map out a strategy for our Association for the next few years. An integral part of this process is clarifying what USENIX is and what it wants to be. The normal technique for this is articulating a mission and a vision for our organization. This is very much in progress and a report will be arriving soon in login. So far, there is consensus on some things: we work a lot on issues near the bleeding edge of technology, we are very interested in practical applications of theory and technology, and we interact a lot with academia, especially computer science education.

For something as important as this, we need input from you, our members. Are there things you consider essential to USENIX? Are there activities so abhorrent we should never do them? For example, should USENIX be concerned with technology in the community, such as funding experimental projects around computer technology in disadvantaged areas? Please let us know your thoughts about what USENIX should be and should do. You can email me (andrew@usenix.org), the Board (bd@usenix.org), write to the editors (login@usenix.org), or even post to comp.org.usenix.

P.S. In the upcoming elections, please consider the importance of having people on the Board who are academics or closely allied with the academic scene. Each year, we spend several hundred thousand dollars on research grants and other academic-related activities, and I consider it vital that we have a strong academic presence on the Board in order to best spend that money. We have been blessed with such a presence in the recent past with Margo Seltzer and currently Peter Honeyman; we need to continue that tradition of excellence.

USENIX MEMBER BENEFITS
As a member of the USENIX Association, you receive the following benefits.

FREE SUBSCRIPTION TO /login/, the Association's magazine, published eight to ten times a year, featuring technical articles, system administration articles, tips and techniques, practical columns on Tcl, Perl, Java, and operating systems, book and software reviews, summaries of sessions at USENIX conferences, and reports on various standards activities.

ACCESS TO /login/ ONLINE from October 1997 to last month.

ACCESS TO PAPERS from the USENIX Conferences starting with 1993, via the USENIX Online Library on the World Wide Web.

THE RIGHT TO VOTE on matters affecting the Association, its bylaws, election of its directors and officers.

OPTIONAL MEMBERSHIP in SAGE, the System Administrators Guild.

DISCOUNTS ON REGISTRATION FEES for all USENIX conferences.

DISCOUNTS on the purchase of proceedings and CD-ROMS from USENIX conferences.

SAVINGS (see <http://usenix.org/membership/membership.html> for details)

- 10% off all Academic Press Professional books
- 10% off BSDL Inc. "personal" products
- 10% off Morgan Kaufmann books
- 20% off New Riders/Cisco Press/MIT books
- 10% off OnWord Press publications
- 10% off The Open Group publications
- 20% off O’Reilly & Associates publications
- $10.00 off Prime Time Freeware publications and software
- 10% off Wiley Computer Publishing books

SPECIAL SUBSCRIPTION RATES (see <http://usenix.org/membership/membership.html> for details)

- $45 subscription to IEEE Concurrently (regularly $88)
- 15% off subscription to The Linux Journal
- $5 off subscription to The Perl Journal
- 20% off subscription to any Sage Science Press Journals

FOR MORE INFORMATION REGARDING MEMBERSHIP OR BENEFITS PLEASE CONTACT
office@usenix.org
Phone: 510 528 8649
New Projects Funded

by Gale Berkowitz
Deputy Executive Director
<mailto:gale@usenix.org>

The USENIX Association is pleased to announce the funding of two important projects that are relevant to the USENIX and SAGE communities: The Internet Software Consortium BINDv9 project, and the Electronic Frontier Foundation’s legal work. Updates on each of these projects will be regularly featured in login over the next several issues.

INTERNET SOFTWARE CONSORTIUM
USENIX is contributing $100,000 to the Internet Software Consortium (ISC) to complete its BIND (Berkeley Internet Name Domain system) version 9 enhancements. BIND is a crucial part of the Internet infrastructure, estimated to be used on between 85% and 99% of all domain name servers on the Internet and needs to remain freely available and open. The project, called “Deep Space BIND” (later renamed BINDv9), includes the following major features:

- full DNSSEC support
- TSIG, TKEY, EDNS0, EDNS1, Notify, IXFR, Negative Caching, A6, DNAME, Bitstring Labels, and Rollover DNS protocol enhancements
- multi-processor scalability
- multi-thread safety

ISC anticipates that the public beta version will be available by February 1, and the Final Release by May 1. For more information about the ISC BIND project, see <http://www.isc.org/products/BIND/>.

ELECTRONIC FRONTIER FOUNDATION
The USENIX Board of Directors approved funding for the Electronic Frontier Foundation (EFF) in the amount of $100,000. The mission of the EFF is to explore civil rights and civil responsibilities online. The funds are to be used to support the legal work in the Bernstein case. This federal legal battle has been going on since 1993 and seeks to protect the Constitutional right to publish encryption software. The case can affect many USENIX members directly and will directly affect the security and privacy of the general public’s interactions across the Internet.

For more information, see: <http://www.eff.org/berstein/>.

Board Meeting Summary

by Gale Berkowitz
& Ellie Young

Here is a summary of some of the actions taken at two of the regular meetings of the USENIX Board of Directors, held in August and November 1999.

Draft Budget 2000
A First Draft Budget for 2000 was presented and approved. The assumptions behind it were discussed, and it was decided to continue to budget conservatively for conference attendance. Any unspent funds in the 1999 Good Works budget will be rolled over to 2000.

Promotion
There was interest in conducting an expanded image marketing and public relations program for USENIX and SAGE. A marketing plan will be presented to the Board at a future meeting.

Proceedings on the Web
It was decided that 12 months after an event, the USENIX conference proceedings will be available on the Web to everyone.

Proposals for Funding

International Research and Development Programme (IRDP). A proposal jointly submitted by USENIX and the NLnet Foundation for $200,000 for an international research exchange program was approved. The program was launched in January 2000. Its goal is to build relationships internationally between research institutions. An advisory committee is being formed with representatives from both institutions. In the first year it will be funded with $100,000 grants from USENIX and the NLnet Foundation.

Support of the Bernstein Case. A proposal from the Electronic Frontier Foundation requesting $100,000 for the support of legal work to pursue a Federal case to protect the Constitutional right to publish encryption software was approved.

International Software Consortium (ISC) BINDv9. A proposal by the ISC to provide funds to complete the BINDv9 project in the amount of $100,000 was approved.

SOS Children’s Village Illinois. A proposal by the SOS Children’s Village Illinois for $40,000 for the purchase of computers, hardware and software for this non-profit foster care agency was approved. Computer instruction will be provided by SAGE members who live in the area.

Standards. A proposal by Nick Stoughton for $80,400 to continue standards activities in 2000 was approved. In the coming year, USENIX will continue to have a presence in the Open Group, POSIX and POSIX Revision groups, and will continue to provide updates to the membership of these activities.

USA Computing Olympiad (USACO). A proposal was approved to continue funding the USACO team at the International Olympiad in Informatics that will be held in Beijing, China, in 2000, in the amount of $52,000.

Software Patent Institute. A proposal from the SPI for a grant of $55,000 was approved. This funding will also allow SPI to improve its computer hardware and software, to con-
continue to process design and administration, and to provide more technical assistance.

Computers, Freedom and Privacy Conference Student Stipends. A proposal for USENIX to provide $20,000 for student stipends for travel and accommodations at this conference was approved.

Sponsorship of the Grace Hopper Women in Computing Conference. The proposal for Gold Sponsorship of the Grace Hopper Conference in the amount of $25,000 was approved. Funds will be used for travel expenses for students to attend the conference.

Fast Software Encryption Workshop Stipends. A request for $10,000 to sponsor the Fast Software Encryption 2000 Workshop was approved. These funds will be used for stipends.

Sponsorship of the African Network Infrastructure Meeting. A proposal from the Network Startup Resource Network for $20,000 for sponsorship of the African Network Infrastructure Meeting in Cape Town in May 2000 was approved. This meeting will promote international networking and train network engineers. The funds will be used to support travel, room and board for engineers coming from African countries.

Conference Registration Fees
It was agreed to increase registration fees for tutorials and technical sessions by $10 per day per event in 2000 (student fees are exempt), with the objective of reducing the projected budget deficit, and providing net connectivity and giveaways (i.e., t-shirts) at all conferences.

Conferences
Atlanta Linux Showcase and Conference (ALS). Young reported that negotiations for USENIX partnering with the volunteer groups (ALS Inc.) of this conference were progressing and an agreement and call for papers would be out soon.

Embedded Systems. It was decided to postpone the workshop that was scheduled for March 1999 by 3-5 months.

Java Virtual Machine Workshop. A proposal from Saul Wold to sponsor a workshop on this topic was approved.

Computing Research Association. USENIX and SAGE will submit two abstracts for two sessions at the upcoming Snowbird conference this summer. They would address system administration and open source topics.

Next Board of Directors Meeting
The next meeting will be held on February 17-18, 2000, in Austin, TX.

20 Years Ago in USENIX

by Peter H. Salus
USENIX Historian
<peter@pedant.com>

The USENIX Association held a conference in Boulder, CO, from January 29 through February 1, 1980. It was preceded by a meeting of STUG, the Software Tools User Group. It was also preceded by the appearance (after 18 months!) of the journal, vol. 5, #1.

STUG was largely concerned with its forthcoming distribution tape and its contents. But two other things proved really important.

First, recall that Kernighan and Plauger published *Software Tools* in 1976. Andy Tanenbaum introduced the LBL crowd (Hall, Scherrer, Sventek) to it, and they were enamored and set to writing tools on a virtual operating system. Also enamored were the students at Georgia Tech. They also worked on tools, using their PDP-11, but they were aiming at PrimeOS.

Second, at the end of the STUG session, Debbie Scherrer announced that the LBL group was looking for someone "to run our UNIX systems." In the audience was a young man attending his first USENIX meeting. He says he literally ran to the front of the room, "throwing chairs out of my way." The group took him out for Chinese lunch--interview, and so Mike O'Dell left the University of Oklahoma for Berkeley, where he became their UNIX guru and the ARPANET Liaison. (The LBL 11/70 was an early ARPANET host.)

I've my copy of "GIT-ICS-79/07: Georgia Tech Software Tools Subsystem User's Guide September 1979" sitting here as I write. It stands as a monument to Perry Flinn, Allan Akin, and Dan Forsythe, who wrote the contents:

Subsystem tutorial
PRIMOS File System Overview
Software tools text editor
User's Guide for the Command Interpreter
User's Guide to the Ratfor preprocessor
and
Software Tools Text Formatter User's Guide

Half a dozen years later, Dan Forsythe was one of the organizers of the Atlanta USENIX (June 1986); Debbie Scherrer served on the USENIX Board for many years, including terms as president and vice president; O'Dell also served on the board and was the founding editor-in-chief of *Computing Systems*.

Stay with me a while, you'll see what I'm getting at.

Al Arms was at Boulder, too, on behalf of Western Electric. He informed the 450 attendees that the Justice Department had said that the UNIX licensing agreements are "compatible" with the consent decree. He also announced a new small-systems license at $700/user to $9,400 for an (unspecified) larger number of users.

Bill Joy spoke about his work on implementing VAX/UNIX paging.

Lou Katz, president of USENIX, announced that he expected the distribution tapes to begin going out around April 1.

Tapes were featured in *login*, too:

**Fourth Software Distribution**
Submissions for the Fourth Software Distribution may be brought to the Boulder
meeting or mailed to arrive in New York before February 15, 1980. On that date we will start packaging the distribution with a target date for first mailings of April Fool’s Day.

The issue of *login*: also contained copies of the Articles of Association and the bylaws of the Association.

The entire issue of *login*: (as well as summary notes on the Boulder meeting by Ian Jackson [U. of Sydney]) appeared in the Dec.–Jan. issue of the AUUG Newsletter.

In 1955, during the transition from the IBM701 to the 704, a number of “operators” in California got together to share software and hardware fixes.

With IBM’s encouragement, this grew into SHARE.

Till the late 1970s, source code wasn’t a question; code came with your machine. Those brown 8-inch floppies from DEC!

With the advent of USENIX began the wholesale exchange of hardware and software bug fixes and — in 1976 — the swapping of tapes. (The First Distribution was May–June 1976, the Second in November.)

Just to give you a “taste,” the Second Distribution contained contributions from the RAND Corporation, the Naval Postgraduate School, UCSD, Yale, and the University of Illinois. The Third followed in May 1977 (when USENIX held its meeting in Urbana, IL, with 150 attendees).

It’s hard to express just how important these tapes were — they contained software from all over that became indispensable to the users. USENIX continued distributing tapes into the late ’80s, when distribution by ftp made them unnecessary.

However, here’s the nub: open source is far older than Linux or GNU. Accessible source and shared code have been with us as long as we have had real computers. In fact, the IBM 701 and 704 were large machines with thousands of diodes and triodes. And, at a point where the ARPANET had under 100 hosts, those USENIX distribution tapes were the way to get the stuff around.

The way you get versatile, robust code is by letting everybody poke at it. Open code is better code.

---

**Report of the Nominating Committee**

by Evi Nemeth
Chair, USENIX Nominating Committee

The USENIX nominating committee has beaten the bushes over the past few months searching for a superb slate of candidates for the 2000 biennial election of the USENIX Board of Directors. And we have found them. Our nominations are:

- Dan Geffner, Exhibitions
  Telephone: 408 335 9445
  FAX: 408 335 5327
  Email: <dfe@usenix.org>

- Daniel V. Klein, Tutorials
  Telephone: 412 421 2332
  Email: <dkv@usenix.org>

- Monica Ortiz, Marketing
  Telephone: 510 528-8649
  Email: <monic@usenix.org>

---

**USENIX BOARD OF DIRECTORS**

Communicate directly with the USENIX Board of Directors by writing to: <board@usenix.org>.

**PRESIDENT:**
Andrew Hume <andrew@usenix.org>

**VICE PRESIDENT:**
Greg Rose <ggr@usenix.org>

**SECRETARY:**
Peter Honeyman <honey@usenix.org>

**TREASURER:**
Dan Geffner <dg@usenix.org>

**DIRECTORS:**
Jon “maddog” Hall <maddog@usenix.org>
Pat Parseghian <pp@usenix.org>
Hal Pomerantz <hal@usenix.org>
Elizabeth Zwicky <zwicky@usenix.org>

**EXECUTIVE DIRECTOR:**
Ellie Young <ellie@usenix.org>

**CONFERENCES**
Judith F. DesHarnais
Registration/Logistics
Telephone: 714 588 8649
FAX: 714 588 9706
Email: <conference@usenix.org>
Two key positions on the USENIX Board are the President and Treasurer. The president must provide vision and guidance for the organization, as well as interface with the staff and chair the board meetings. The treasurer is responsible for not only keeping an eagle eye on finances but also for advising USENIX on investing its endowment funds.

Andrew Hume has served as USENIX President for the last four years Dan served as both Vice President and Treasurer (two years each). We are nominating Dan for President and Andrew for treasurer. Both have done superb jobs in their respective positions, and it is our hope that by broadening each of their focuses, the team can be even more effective than it has been in the past.

Dan Geer is well known as a visionary, and we believe that the role of President will give him the opportunity to do for USENIX what he has done for any of a number of different organizations. Andrew wrote the software that gives AT&T real time auditing capabilities of all our long distance phone bills, so we hope to take advantage of his expertise to maintain and grow the financial stability of the organization.

In short, both Andrew and Dan have enormous skill sets and by changing their positions, we hope to take even better advantage of them.

We are nominating Kirk McKusick to run for the position of Vice President. Kirk is a past President of the USENIX and represents both the academic community as well as the free software constituency. Kirk has a PhD in Computer Science and an MBA from the University of California, Berkeley; he was the Research Computer Scientist for Berkeley's Computing Systems Research Group (the people who brought you BSD); and is now an instructor at both UC Berkeley and UCLA. He has recently been quite active in the development and evolution of the FreeBSD track at the USENIX Technical Conference and is serving as its program chair at the USENIX Annual Conference in June 2000.

We are nominating Mike Jones for the position of Secretary. Mike has been an active participant in the USENIX community for the past decade. He earned his PhD at Carnegie Mellon University working on the Mach project and has been a researcher in Microsoft's Research Lab for the past several years. Mike publishes regularly in USENIX conferences, has served on a number of program committees (the Annual Technical Conference, OSDI and Windows/NT), and was instrumental in starting the USENIX Windows NT Symposia, and will be program co-chair for OSDI 2000. Mike brings a strong academic bent, as well as boundless energy and a commitment to USENIX.

We have nominated six candidates to run for the four positions of Director at large. The most important criteria for board members is their willingness and ability to get things done and work together productively. Your board does a tremendous amount of work for you and for the organization, and we need eight actively engaged members. For each of the candidates we are nominating for Director at large, we outline the constituency they represent and the particular strengths that led to their nomination. In the election materials that members will receive in March, the candidates themselves will issue statements describing their backgrounds and goals for serving as board members.

Alphabetically:

John Gilmore was an early employee of Sun and Cygnus and is a founder of the Electronic Frontier Foundation. He has been very active in the societal side of UNIX and the whole IT movement. John has been outspoken and effective at challenging things like the Computer Decency Act, the alleged safety of 40 bit keys for cryptographic use, the export controls legislation, etc. He brings to the board a wider view than previous board members.

Jon (Maddog) Hall is currently serving his first term as a USENIX board member. Maddog was a UNIX supporter at Digital for many years, and now is affiliated with VA Linux and Linux International. He is a strong representative of the Linux community and cares very deeply about the interaction and relationship between the USENIX and Linux communities. As a current board member, he
adds an element of depth to the slate of nominees.

Dirk Hohndel got started with UNIX as a sysadmin managing Suse Linux systems while he was a Computer Science student at Würzburg University. After finishing his Masters degree he went to a startup, on to Deutsche Bank and is now with Suse Linux in Germany. He may be best known to our community for his work on the XFree86 window system for PCs which he did in his spare time and still helps maintain. Dirk wants to strengthen the bond between the USENIX and Linux communities.

Darrell Long is a Professor of Computer Science at the University of California at Santa Cruz. He has been a member of the board’s scholastic committee for the past few years, served on several program committees, and publishes at USENIX conferences regularly. Darrell adds academic representation to the board. He is concerned with maintaining the high quality of our conferences.

Marcus Ranum is well known in the security and SAGE communities and a frequent USENIX tutorial speaker. Marcus was program chair of the Intrusion Detection workshop and has served on several program committees. Marcus has been running his own small company, Network Flight Recorder, that sells a security monitoring software package used by system administrators.

Avi Rubin is a young researcher at AT&T and an adjunct faculty member at New York University. He has been program chair for both the Security conference and for the General Conference. Avi became involved with USENIX as a student when he published his first paper here; now 6 years and many papers later, he is ready to start giving back to the organization. Avi is a finisher, gets things done, and will be a hard worker on the board.

We were fortunate to get a good mix of excellent, experienced folks and some really terrific new folks. The committee is aware that the slate contains no women. We approached several outstanding possible candidates, but other commitments prevented their acceptance.

The USENIX Nominating Committee,

Evi Nemeth, University of Colorado, Chair
Trent Hein, XOR Network Engineering
Steve Johnson, Transmeta Corp.
Dennis Ritchie, Bell Laboratories
Margo Seltzer, Harvard University
Ready! Set. No?

by Tina Darmohray

Tina Darmohray, co-editor of *login:*, is a consultant on Internet firewalls and network connections and frequently gives tutorials on those subjects. She was a founding member of SAGE.

tmd@usenix.org

My husband and I frequently receive compliments on our children’s behavior. A fellow father will comment on how well-behaved they are when they’re out at a building-supply center with my husband, or a couple will swing by our dinner table at a restaurant and compliment us on our family. Heck! Even the photographer has gone on about how easy it is to photograph our kids because they’re so cooperative. Now, do my husband and I think our kids are perfect? Not a chance!

So, why do we get such compliments? We’ve given this some thought. Our best guess is that it’s a combination of things: we’ve got reasonable kids to start with, we threaten them before we go into public places (joke), and we do our best not to set them up to fail. What do I mean by that last one? That means we choose family restaurants rather than five-star, for instance. We also make sure they’re rested, fed, comfortable, and have appropriate entertainment available. Otherwise, we just don’t go. In short, they behave because they’re in situations where they can.

I’m a firm believer that this same principle applies to adults in the workplace. A good manager doesn’t set employees up to fail. Setting folks up to fail is a lose-lose situation. Managers who don’t heed this rule have the scenario where the employee not only fails at the task but also gets the accompanying negative feelings of being a failure. Lose-lose situations should be avoided wherever possible.

Still, I see way too much of it in the workplace; some examples are below.

A security incident has occurred, so now security response and fixes are the focus of everyone in the company, from the CEO down. The security group are suddenly in the hot seat. They pull all-nighters to get things running again. Just as things return to normal, the message comes down from the top: “Rearchitect the site over the weekend to avoid this situation again.” Everyone knows that responding to this knee-jerk reaction isn’t possible or even advisable. Still, those in charge demand immediate action and a formal report on the tested “perfect solution” with an arbitrary deadline for presentation to the president and CEO by close of business on Friday.

What happens? The security group spend the rest of the week setting up tests of various vendor solutions and calling upon whatever applicable internal resources are necessary. No one is working smart, and everyone is feeling as though they’re in a never-ending fire drill. It turns out that this arbitrary deadline really can’t be met. The week ends, the report of “no solution” gets presented to the top, and the whole urgency of the exercise gets lost going forward. Meanwhile, the security group are burnt out, frustrated, and feeling like failures on all fronts.

An internal support employee is doing a good job supporting two internal groups. One customer group, unaware of the split-time nature of the support employee, requests more of his time from his manager. The employee’s manager increases the hours, committing, on paper, every available hour in the employee’s day. This leaves no “overhead” time for the employee to keep up on mail, fiddle with something new, read a trade magazine, and so on. Now there’s an inherent expectation that the employee can’t meet. He’ll either have to work extra hours or disappoint the customer. This dilemma sets him up to fail.

Let’s not omit the proverbial favorite of being assigned the responsibility without the authority. The senior network administrator is responsible for configuring the company’s router so that it protects the corporate computer assets. Such decisions ideally involve key management and technical personnel who create a corporate security policy. Once in place, such a policy can easily be implemented, including configuring the ACLs in the company’s Internet router. Without such a document, or consensus, the network administrator is left to deploy her “best guess.” In this situation she’s “damned if she does and damned if she doesn’t.” If the company is broken into, she’ll clearly get the heat, but if she implements what she guesses to be “right,” she’ll likely be criticized for that too. Being in a situation where you have the responsibility but not the authority is being set up to fail.

Managing projects or people under unrealistic or arbitrary deadlines is a recipe for failure. Not only do you miss the target of delivering good work in a timely fashion, but managers also suffer the collateral damage of discouraging their employees en route. I maintain that it’s far better to give the “bad” news up front by setting the expectations of the customer (the boss, the client, the group you’re working in conjunction with) realistically to begin with. This may mean that you have to push back on a deadline, let someone know that you can’t support them at the level they’d like immediately, or tell them that you need something from them in order for you to be successful going forward. No matter what it takes, shooting for a win-win situation where the employee succeeds and everyone feels good about it is worth the realism up front. If it seems dicey to begin with, remind yourself that folks often have a short memory, and if you deliver the goods in a consistent and timely manner, you’ll be well thought of in the end. Over time, you’ll build respect from your employees and confidence from your peers, managers, and clients.
From the SAGE President

by Barbara Dijker

Barbara Dijker is currently SAGE president. She’s been sysadminning for about 12 years and runs a couple of ISPs.

<barb@usenix.org>

A few years ago, the SAGE executive committee finally ratified formative documents. At the same time, we tried to address the idea of continuity and consistency in the SAGE executive committee by synchronizing terms and clearly (as one can predict future cases) outlining procedures for appointing officers and filling vacancies. Getting all this in writing has proven very useful. 1999 is the beginning of the first full elected term for the SAGE executive where the organizational documents have been in place.

While terms of SAGE executives are two years, SAGE officers are appointed by and within the executive committee for a term of one year. Normally, at mid-term, existing officers are simply reappointed. When someone decides to step down from an office, however, another officer is appointed.

Being on the SAGE executive committee represents a significant commitment. At the least, it involves meeting in person three to four times each year and participating in four to eight teleconferences between the in-person meetings. In addition, any project a SAGE executive takes on involves time delegating, coordinating, and ensuring follow-through. This is commonly referred to as “herding cats” – difficult indeed. Add to that the duties of particular offices. Since we’re all system administrators who are notoriously overworked, and being on the SAGE executive committee is a volunteer extracurricular activity, paid work and family can take priority. The SAGE organizational documents provided a mechanism for executives to tender resignations and have those vacancies filled so that the work can continue.

Mid-term this year, we had one officer step down from office (but not from the executive committee) and two executives resign from their term. After significant deliberation, the end result is the following new SAGE executive committee: Barb Dijker (president), Xev Gittler (vice president), Peg Schafer (treasurer), David Parter (secretary), Geoff Halprin, Hal Miller, and Bruce Alan Wynn.

My job as president will be much easier for the prior work done by Hal Miller. This executive committee can and has hit the ground running. As with any such group, progress is a team effort. We’re only as effective as the least of the group. With a full team on board again, we’re able to move full steam ahead.

Looking forward, we have quite a few significant projects in the works. The most important things we can do for our membership fall under the broad category of building credibility for and continuing the definition of the profession of system administration. We are doing this through education and certification, building our own ranks, and increasing awareness in the general public.

The SAGE occupational analysis survey, which took place in October and November, generated over 1,000 responses. This information will form the basis of future education and certification efforts. John Seehurst chaired a productive workshop on education at LISA which has formed the basis for future collaboration in this area. SAGE has been working with Sun Microsystems, who are investing significant resources in system administration education with Sun Network Academy (SNAP) and their College Resource & Instructor Support Program (CRISP). Also in the education area, we have a unique opportunity to address computer science faculty at the CRA (cra.org) conference next summer. Dave Parter will be coordinating a session there to show CS faculty why they should and how to implement system administration courses in CS curricula.

SAGE, the System Administrators Guild, is a Special Technical Group within USENIX. It is organized to advance the status of computer system administration as a profession, establish standards of professional excellence and recognize those who attain them, develop guidelines for improving the technical and managerial capabilities of members of the profession, and promote activities that advance the state of the art or the community.

All system administrators benefit from the advancement and growing credibility of the profession. Joining SAGE allows individuals and organizations to contribute to the community of system administrators and the professions as a whole.

SAGE membership includes USENIX membership. SAGE members receive all USENIX member benefits plus others exclusive to SAGE.

SAGE membership includes USENIX membership. SAGE members receive all USENIX member benefits plus others exclusive to SAGE.

SAGE members save when registering for USENIX conferences and conferences co-sponsored by SAGE.

SAGE publishes a series of practical booklets. SAGE members receive a free copy of each booklet during their membership term.

SAGE sponsors an annual survey of sysadmin salaries collated with job responsibilities. Results are available members online.

The SAGE Web site offers a members-only Jobs-Offered and Positions-Sought Job Center.

SAGE STG EXECUTIVE COMMITTEE

PRESIDENT:
Barbara L. Dijker <barb@usenix.org>

VICE PRESIDENT:
Xev Gittler <xev@usenix.org>

SECRETARY:
David Parter <parter@usenix.org>

TREASURER:
Peg Schafer <peg@usenix.org>

MEMBERS:
Geoff Halprin <geoff@usenix.org>
Hal Miller <hal@usenix.org>
Bruce Alan Wynn <wynn@usenix.org>
SAGE currently has about 5,500 members. Membership could top 7,000 in the year 2000. Compare this to about 2,000 who attend the LISA conference each year and only seven years of existence. Growth in SAGE membership has been strong: 36% this past year. In addition, SAGE has been growing in number of local groups and international affiliates. Last year, Australia was the only international affiliate. This year groups in WISE (Wales, Ireland, Scotland, England), Portugal, and the Netherlands have started or are forming. A centralized Web site is being created to provide common information and links. We are also working on making SAGE more visible to system administrators by partnering with vendors to include SAGE flyers in their products.

Increasing awareness in the general public is a tough problem. This all can be broadly considered marketing. There are essentially four aspects of this activity: active promotion (advertising and press releases), passive promotion (media references to SAGE), vendor relations, and member pride. Interestingly, passive promotion is harder than active. SAGE is working on doing more promotion, both active and passive, than in the past. Recently, Dr. Dobbs' Journal published an article on system administration careers that refers to SAGE. Small things like this can have a significant effect. We are also going to be working with appropriate parties to get SAGE to be the voice of our community for quoting in industry press. Vendors are tricky because they often (incorrectly) don't view system administrators as important targets. There are several reasons why this is changing, enough to be a separate article later. We hope to leverage this and establish valuable relationships.

There was a great deal of discussion about this topic at the SAGE community meeting at LISA. It was gratifying to finally hear the membership change from questions about why SAGE needs to exist and why are we doing this or that to questions about how can we make SAGE more visible to the rest of the world. This shift acknowledges that SAGE is moving in a positive direction and we need to flaunt it. We hope you wear your SAGE pin every day and your SAGE T-shirts frequently. We will be developing more and better ways to enable our thousands of members to help us in this effort.

So while there has been some "changing of the guard," we plan to not miss a beat. Momentum and energy within the organization are at what I see as an all-time high. Things set in motion many years ago are starting to click and fall into place. My job is to simply watch it happen and take all the credit. More seriously, this is a result of the concerted efforts of many individuals, many of whom are not on the executive committee and all of whom have other work of higher priority (like for pay). My job is really to make sure nothing falls in the cracks.

SAGE Update

by Gale Berkowitz
Deputy Executive Director
<gale@usenix.org>

A lot has been going on with SAGE these days. The SAGE Executive Committee recently met in Berkeley, CA, and the summary of that meeting will be available on the Web at <http://www.usenix.org/sage/people/excmemos/>. Here are updates on two of SAGE's important projects, the SAGE Certification Project and the SAGE System Administrator Profile.

SAGE CERTIFICATION PROJECT

As part of the SAGE Certification Project, SAGE conducted its Occupational Analysis survey. The survey, which ended in mid-November, had over 1,200 respondents. A preliminary report on the survey is being reviewed by the SAGE Certification Committee and the SAGE Executive Committee. The next steps include deciding on the direction and scope of the Certification Project.

SAGE SALARY SURVEY

The SAGE Salary Survey was redesigned this year, including many new questions and a new format. For the first time, the survey was conducted online. Results from the survey will be available to all SAGE members, and a summary will be posted on the Web.

SAGE SUPPORTING MEMBERS

Collective Technologies
Deer Run Associates
Electric Lightwave, Inc.
ESM Services, Inc.
GNAC, Inc.
Macmillan Computer Publishing, USA
Mentor Graphics Corp.
Microsoft Research
MindSource Software Engineers

Motorola Australia Software Centre
New Riders Press
O'Reilly & Associates
Remedy Corporation
RIPE NCC
SysAdmin Magazine
TransQuest Technologies, Inc.
UNIX Guru Universe (UGU)
14th Systems Administration Conference (LISA 2000)

December 3-8, 2000

New Orleans Marriott Hotel, New Orleans, Louisiana

Important Dates:
- Extended abstracts due: June 6, 2000
- Invited Talk proposals due: June 6, 2000
- Notification to authors: July 10, 2000
- Final papers/Invited Talks due: October 11, 2000

Conference Organizers:
- Program Chairs
  Remy Evard, Argonne National Laboratory
  Phil Scarf, GE-Finuc Automation
- Program Committee
  Jeff Allen, WbTV Networks
  David Blank-Edelman, Northeastern University
  Strata R. Chalup, VirtualNet Consulting
  Trey Harris, Mail.com
  Christine Hogan, Imperial College
  Doug Hughes, Auburn University
  Ruth Milner, NRAO
  Car Okita, Globalcenter
  John Orthoefer, GTE Internetworking
  David Parce, University of Wisconsin
  Josh Simon, Collective Technologies
- Invited Talks Coordinators
  Pat Wilson, Dartmouth College
  Tom Limoncelli, Lucent Technologies / Bell Labs
- Network Administration Symposium
  Coordinators
  Bill LeFebvre, Group Sys Consulting
  David Williamson, Global Networking and Computing
- Security Symposium Coordinator
  Simon Cooper, SGI
- The Guru Is In Coordinator
  Lee Damon, Qualys.com

The Only Conference By and For System Administrators

"A Depth and Breadth of Experience" If LISA isn't the number one event on every system professional's calendar, it should be. As the only technical conference designed and run by veteran system administrators, LISA 2000 will offer a unique depth and breadth of technical expertise for hard-pressed systems, network, and security administrators from sites of all sizes. The depth and breadth of experience of LISA attendees makes it an event unlike any other in the industry.

With the millennium hype behind us, LISA 2000 focuses on the future of the industry as the system administration workforce is stretched to capacity and the demand for diverse skills increases. At LISA 2000, the tutorials, taught by leaders in their fields, will bring you up to speed with new skills or introduce you to the most advanced features of your favorite languages and tools. Refereed papers and invited talks offer insights into all areas of system administration. "Guru" and "Birds-of-a-Feather" sessions provide opportunities for individual advice and discussion.

In addition, the technical program has expanded with a new symposium track, focusing on two very specialized topics: Network Administration and Security. This new symposium track will highlight the trends, solutions, and breakthroughs in networking, security, and intrusion detection.

And, as always, the "hallway track" will be running for informal discussions with your peers and other experts in all areas of system administration.

How You Can Participate
Beyond attending LISA, we also invite you to submit a proposal for any of these conference events:
- Teaching a Tutorial
- Organizing a Workshop
- Writing a paper for the Technical Program
- Delivering an Invited Talk
- Coordinating a Symposium Track
- Participating in a Symposium

Conference Topics
The Program Committee invites you to join the contributors to the LISA XIV conference. Submissions of refereed papers or other presentations which address any and all aspects of system administration are acceptable. Here is a partial list of timely paper topics for potential authors:

- Innovative system administration tools and techniques
- Tips and tricks: new uses for old tools
- Distributed or automated system administration
- High availability and disaster recovery
- Scaling support of "open source" systems for servers and desktops
- Designing, selecting, scaling, integrating and managing "enterprise" computing services.
- Security (all aspects)
- Authentication systems
- Applications of tools, techniques and methods from other disciplines
- Integration of new networking technologies, protocols and applications
- Integration of emerging technologies
- Performance analysis and monitoring

The "Soft Science" of System Administration
- Methodology, paradigms and models for system administration
- Analysis of "best practices" in systems administration
- Analysis and comparison of alternative systems for system administration
- Case studies
- Application of scientific methods to systems administration
- Metrics for systems administration

The System administrator's role in the organization

Tutorial Program
Gain mastery of complex techniques and technologies, and you'll get immediate payoff within your organization. You can choose from up to 40 full- and half-day classes over three days. Whether you are a novice or senior systems administrator, you will be able to find a tutorial to meet your needs. Tutorials cover important topics such as: performance tuning, administering Windows NT, Perl, TCP/IP troubleshooting, security, networking, network services, sendmail, Samba, legal issues, and professional development.

Submitting A Tutorial Program Proposal
To provide the best possible tutorial offerings, USENIX continually solicits proposals for new tutorials. If you are
interested in presenting a tutorial at this or other USENIX conferences, please contact the tutorial coordinator: Daniel V. Klein (Tel: 1.412.421.0285; Fax: 1.412.421.2332; Email: duk@usenix.org).

**Workshops**

For the past several years, the LISA conference has held limited-attendance workshops during the first few days of the week. These have included the popular "Advanced Topics Workshop", the "Global LISA" workshop, and, last year, a workshop on education issues. The format of these workshops vary, but can be thought of as focused Birds-of-a-Feather sessions that last for half a day or a full day.

Up to three workshops will be offered alongside the Tutorial Program at LISA 2000. If you are interested in organizing a LISA 2000 workshop, please submit a workshop proposal. The proposal should address the following questions/issues:

1. workshop organizers
2. topic/goals
3. format
4. projected attendance - limited (One workshop can be as large as 60 people; other workshops are limited to "roundtable" configuration)
5. target audience
6. how you will recruit participants
7. special needs, if any
8. anything else we should know

Please submit proposals via email to lisa2000chairs@usenix.org no later than May 1, 2000.

**Technical Sessions**

Three days of technical sessions feature parallel tracks of Refereed Papers, Invited Talks, the Symposium Track, and "The Guru is In" sessions for individual consultation with experts on specific topics. Refereed papers are published in the Proceedings (provided free to Technical Sessions attendees). Invited Talk and Symposium Track materials are made available online.

**Cash Prizes**

Cash prizes will be awarded at the conference for the best paper and for the best paper by a student. Prizes are for papers accepted to the refereed paper track.

**Submitting an Invited Talk Track Proposal**

If you have a topic of interest to systems administrators that is suitable for an invited talk, please submit a proposal to the Invited Talk coordinators.

Email your proposal to: itilisa@usenix.org.

**Invited Talk proposals due:**
June 6th, 2000

**Final copy due:**
October 11, 2000

**Submitting a Symposium Track Proposal**

If you have a topic of interest that would fit into one of the two symposia (Network Administration or Security), please send e-mail to the appropriate address below.

Each symposium will have a structure similar to the LISA conference as a whole. Refereed papers should be submitted like any other refereed paper to be reviewed by the committee as a whole. Ideas for longer talks, panels, and other presentations should be submitted to the respective symposia chair.

**Network Administration Symposium:**
lisanetadm@usenix.org

**Security Symposium:**
lisasecadm@usenix.org

**Work-in-Progress Reports**

Do you have interesting work you would like to share, or a cool idea that is not yet ready to be published? The USENIX audience provides valuable discussion and feedback. We are particularly interested in presentation of student work. To schedule your short report, send email to:
lisawps@usenix.org.

**Birds-of-a-Feather Sessions**

Birds-of-a-Feather sessions (BoFs) are very informal gatherings organized by attendees interested in a particular topic. BoFs are held Tuesday, Wednesday, and Thursday evenings. BoFs may be scheduled in advance by phoning the Conference Office at 1.949.588.8649 or via email to: conference@usenix.org. BoFs may also be scheduled at the conference.

**What to Submit to the Refereed Track**

A summary of two to four pages is required for the paper selection process. A summary should contain:

- A 1 paragraph abstract
- No more than 4 pages of text summarizing the final paper
- Any relevant charts or graphs
- Full papers are not acceptable at this stage; if you send a full paper, you must also include the summary. Include appropriate references to establish that you are familiar with related work, and, where possible, provide detailed data to establish that you have a working implementation or measurement tool.

Submissions will be judged on the quality of the written submission, and whether or not the work advances the state-of-the-art of system administration. Please consult the detailed author guidelines available on the conference Web site at: http://www.usenix.org/events/lisa2000.

Note that LISA, like most conferences and journals, requires that papers not be submitted simultaneously to more than one conference or publication, and that submitted papers not be previously or subsequently published elsewhere for a certain period of time. Papers accompanied by non-disclosure agreement forms are not acceptable and will be returned unread. All submissions are held in the highest confidence prior to publication in the conference proceedings, both as a matter of policy and as protected by the U.S. Copyright Act of 1976.

Every accepted paper must be presented at the conference by at least one author. Authors of an accepted paper must provide a final paper for publication in the conference proceedings. One author of each accepted paper receives complimentary technical session registration. Final papers are limited to 20 pages, including diagrams, figures and appendices. Complete instructions will be sent to authors of accepted papers.

To discuss potential submissions and for inquiries regarding the content of the conference program, contact any member of the program committee or the program chairs:

Remy Evard and Phil Scarr
Email: lisa2000chairs@usenix.org
Tel: (630) 252-5963 or (804) 978-5507

All submissions for LISA '99 will be electronic. Please use the web form at: http://www.usenix.org/events/lisa2000/cfp/submit.html

**Program and Registration Information**

Complete program and registration information will be available by September 2000 at the conference website at: http://www.usenix.org/events/lisa2000. The information will be in a printable PDF file.

If you would like to receive the program booklet in print, please email your request, including your postal address, to conference@usenix.org.
May You Live in Interesting Times

Will Rogers said, “All I know is what I read in the papers.”

PC Week had an interesting editorial in late December. It’s titled “Red Hat: Not Fitting?” They suggest that Red Hat is “listening more to Wall Street’s pandermers of quick wealth than to the customers who use the product.” Imagine that. Remember – we’re talking serious money here. Tens and hundreds of millions of dollars per member of the founding team.

PC Week notes that Red Hat Chair Bob Young asserts that Red Hat’s competition is not other Linux distributions but, rather, Microsoft. The editorial suggests Young should “wait in line behind Scott McNealy [Sun] and Larry Ellison [Oracle].” Sage advice that. The editorial makes the wry comment that “An anti-Microsoft animus has hardly been a formula for success in the past.” I reckon that’s not strictly true; both McNealy and Ellison have not exactly embraced the leader in the operating system market.

PC Week continues with a note that comments like “runs best on Red Hat” for products like the Mozilla browser and sendmail server would diminish both the Open Source movement and Red Hat. They conclude with comments that Linux needs more than one vendor, even if one vendor is large.

The Open Source (which I think really is heard by many as “free software”) movement is an interesting one. The money in sales for free software isn’t so hot, I fear. Even in volume, a large number times $0.00 isn’t too high. On the other hand, people do seem willing to pay $10–100 for distributions (e.g., of the Linux and “BSD CD-ROMs). It will be interesting to see how the bottom-line business model (vs. the paper wealth of stock model) turns out.

The industry has surely changed in just a few years, particularly in the world of financing for high-tech companies. The end-of-year stock market (and I’m writing this a few days before the Y2K excitement either pans out or fizzles) has continued throughout this year to put ever more glitz on high-tech stocks. My friend at Qualcomm is giddy over the more than 1500% rise in its stock price over the last year. Watching so much money move is truly an astounding game. Yahoo joined the DJIA. The finance people are watching and investing their hearts out.

I keep trying to find a way to create value and wealth. That particular proposition is an elusive and challenging one. I think we should all try to be careful not to disappoint the investment community and poison the well for our own financial futures. I’ll let you know when I figure out how to do that!
The Power of
UNIX on WINDOWS!

NOW ALL YOUR UNIX KNOWLEDGE IS JUST AS USEFUL ON WINDOWS
...AND SO IS YOUR UNIX CODE!

Are you a UNIX® pro working on Windows®?
Developing? Deploying? Administering?

NOW ON WINDOWS YOU CAN:

• Use a UNIX command-line environment: shells, commands, and utilities
• Run UNIX scripts
• Port business-critical UNIX programs
• Do cross-platform development—write once & deploy on both Windows and UNIX
• Web-enable existing applications
• Interoperate with UNIX using X, NFS and telnet
• Integrate mission-critical and workgroup applications onto a common desktop

Choose MKS Toolkit®, Toolkit Select, or NuTCRACKER® Professional. All these products have been proven to cut learning, development and support time—70%, 80%, even 90%. And there are thousands of customers and case studies to prove it.

TO ORDER YOUR 30-DAY RISK-FREE TRIAL OFFER,
CALL 800-637-8034 OR 703-803-3343 OR
VISIT US ON THE WEB AT WWW.MKS.COM/INTEROP

ORDER YOUR FREE PORTING GUIDE TODAY!

MKS, MKS logo, MKS Toolkit, NuTCRACKER and NuTCRACKER logo are registered trademarks of Mortice Kern Systems Inc. UNIX is a registered trademark in the U.S. and other countries, licensed exclusively through X/Open Company Limited. Windows, Win32, and Windows NT are either registered trademarks or trademarks of Microsoft Corporation in the U.S. and/or other countries.
CONTRIBUTIONS SOLICITED

Your are encouraged to contribute articles, book reviews, photographs, cartoons, and announcements to ;login:. Send them via email to <login@usenix.org> or through the postal system to the Association office.

The Association reserves the right to edit submitted material. Any reproduction of this magazine in part or in its entirety requires the permission of the Association and the author(s).

The closing dates for submissions to the next two issues of ;login: are April 5 and May 2, 2000.