

;login:

THE MAGAZINE OF USENIX & SAGE

October 2002 volume 27 • number 5



inside:

OPINION

Chalup: Vive la Révolution! Now Get Over It!

THE LAW

Nicholson: Software Licensing 101

PROGRAMMING

McCluskey: C99 and Compatibility

Turoff: Practical Perl

Flynt: Generating Ethernet Packets

SECURITY

Farrow: Musings

SYSADMIN

Haskins: ISPadmin

Skvarcek: Remote Monitoring with SNMP

Carlini: Cabling

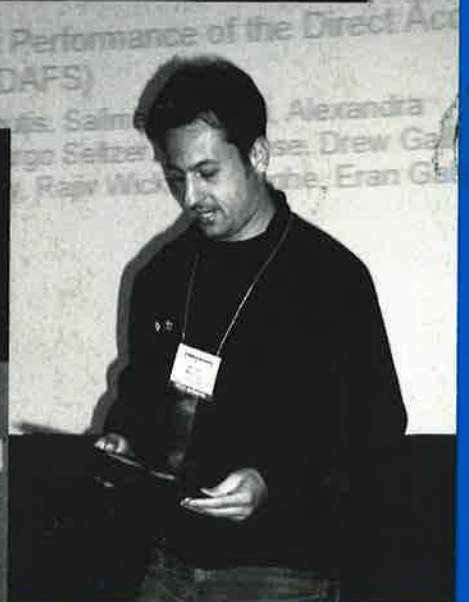
CONFERENCE REPORTS

USENIX 2002

2002 Linux Kernel Summit

Java™ VM 2002

and Book Reviews, Standards Reports,
and News from USENIX and SAGE



USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

16TH SYSTEMS ADMINISTRATION CONFERENCE (LISA '02)

Sponsored by USENIX & SAGE

NOVEMBER 3-8, 2002

PHILADELPHIA, PENNSYLVANIA, USA

<http://www.usenix.org/events/lisa02>

INTERNET MEASUREMENT WORKSHOP 2002

Sponsored by ACM SIGCOMM, co-sponsored by ACM SIGMETRICS and USENIX

NOVEMBER 6-8, 2002

MARSEILLE, FRANCE

<http://www.icir.org/vern/imw-2002/>

5TH SMART CARD RESEARCH AND ADVANCED APPLICATION CONFERENCE (CARDIS '02)

Sponsored by USENIX and the IFIP Working Group 8.8 (Smart Cards)

NOVEMBER 20-22, 2002

SAN JOSE, CALIFORNIA, USA

<http://www.usenix.org/events/cardis02>

2ND WORKSHOP ON INDUSTRIAL EXPERIENCES WITH SYSTEMS SOFTWARE (WIESS '02)

Sponsored by USENIX

Co-sponsored by ACM SIGOPS & IEEE TCOS

DECEMBER 8, 2002

BOSTON, MASSACHUSETTS, USA

<http://www.usenix.org/events/wiess02>

5TH SYMPOSIUM ON OPERATING SYSTEMS DESIGN AND IMPLEMENTATION (OSDI '02)

Sponsored by USENIX

Co-sponsored by ACM SIGOPS & IEEE TCOS

DECEMBER 9-11, 2002

BOSTON, MASSACHUSETTS, USA

<http://www.usenix.org/events/osdi02>

5TH NORDU/USENIX CONFERENCE (NORDU2003)

Sponsored by EurOpen.SE and USENIX

FEBRUARY 10-14, 2003, VÄSTERÅS, SWEDEN

Web site: <http://www.nordu.org/NordU2003/>

4TH USENIX SYMPOSIUM ON INTERNET TECHNOLOGIES AND SYSTEMS (USITS '03)

MARCH 26-28, 2003

SEATTLE, WASHINGTON, USA

<http://www.usenix.org/events/usits03>

Notification of acceptance: November 8, 2002

Camera-ready final papers due: January 17, 2003

2ND CONFERENCE ON FILE AND STORAGE TECHNOLOGIES (FAST '03)

MARCH 1-APRIL 2, 2003

SAN FRANCISCO, CALIFORNIA, USA

<http://www.usenix.org/events/fast03>

Notification of acceptance: November 1, 2002

Camera-ready final papers due: January 6, 2003

THE FIRST INTERNATIONAL CONFERENCE ON MOBILE SYSTEMS, APPLICATIONS, AND SERVICES (MOBISYS '03)

Jointly sponsored by ACM SIGMOBILE and USENIX in cooperation with ACM SIGOPS

MAY 5-8, 2003

SAN FRANCISCO, CA, USA

<http://www.usenix.org/events/mobisys03/>

Notification of acceptance: December 18, 2002

Camera-ready final papers due: March 4, 2003

2003 USENIX ANNUAL TECHNICAL CONFERENCE

JUNE 9-14, 2003

SAN ANTONIO, TX, USA

Paper submissions due: November 18, 2002

12TH USENIX SECURITY SYMPOSIUM

AUGUST 4-8, 2003

WASHINGTON, DC

Paper submissions due: January 27, 2003

contents

2 **MOTD** BY ROB KOLSTAD

3 **APROPOS**

BY TINA DARMOHRAY

4 **LETTERS TO THE EDITORS**

;login: Vol. 27 #5, October 2002

;login: is the official magazine of the USENIX Association and SAGE.

;login: (ISSN 1044-6397) is published bimonthly, by the USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710.

\$50 of each member's annual dues is for an annual subscription to *;login:*. Subscriptions for nonmembers are \$60 per year.

Periodicals postage paid at Berkeley, CA, and additional offices.

POSTMASTER: Send address changes to *;login:*, USENIX Association, 2560 Ninth Street, Suite 215, Berkeley, CA 94710.

©2002 USENIX Association. USENIX is a registered trademark of the USENIX Association. Many of the designations used by manufacturers and sellers to distinguish their products are claimed as trademarks. Where those designations appear in this publication, and USENIX is aware of a trademark claim, the designations have been printed in caps or initial caps.

OPINION

6 **Vive la Révolution! Now Get Over It!** BY STRATA R. CHALUP

THE LAW

9 **Software Licensing 101** BY JOHN NICHOLSON

PROGRAMMING

20 **C99 and Compatibility** BY GLEN MCCLUSKEY

23 **Practical Perl** BY ADAM TUROFF

26 **Generating Ethernet Packets** BY CLIF FLYNT

SECURITY

31 **Musings, Or What Did I do on My Summer Vacation** BY RIK FARROW

SYSADMIN

35 **ISPadmin** BY ROBERT HASKINS

41 **Remote Monitoring with SNMP** BY JOZEF SKVARCEK

50 **Cabling: Just the Tip of the Iceberg** BY JAMES CARLINI

BOOK REVIEWS

52 **The Bookworm** BY PETER H. SALUS

53 **Incident Response** by Eugene Schultz and Russell Shumway – REVIEWED BY ANTON CHUVAKIN

54 **Honeypots: Tracking Hackers** by Lance Spitzner – REVIEWED BY ANTON CHUVAKIN

STANDARDS REPORTS

55 **Austin Group (POSIX) Status Update (May 14, 2002)** BY ANDREW JOSEY

55 **Committee Maintenance Procedures for the Approved Standard (POSIX) (revised May 21, 2002)** BY ANDREW JOSEY

SAGE NEWS

59 **SAGE Elections** BY BRYAN ANDREGG

USENIX NEWS

60 **Twenty-five Years Ago in ;login:** BY PETER H. SALUS

CONFERENCE REPORTS

62 **2002 USENIX Annual Technical Conference, Monterey, CA, June 10-15, 2002**

87 **2002 Linux Kernel Developers Summit, Ottawa, Canada, June 24-25, 2002**

91 **2nd Java™ Virtual Machine Research and Technology Symposium, San Francisco, CA, August 1-2, 2002**

Cover: Best Paper award winners at USENIX: clockwise from top left: Alexandra (Sasha) Fedorova and Rajiv Wickremesinghe; Ludmila Cherkasova and Wenting Tang; Kostas Magoutis; Sotiria Lampoudi and David M. Beazley. See pages 62ff.

motd

by Rob Kolstad

Rob Kolstad is currently Executive Director of SAGE, the System Administrators Guild. Rob has edited *;login:* for over ten years.



kolstad@sage.org

Leverage

I keep thinking that I will soon be wizened and know everything I need to know. I avoid prejudices and try to judge actions and results more than hype.

I try to partake of technology services so that I can respond intelligently about Web sites, software services, initiatives, ideas, and the like. I pay my bills online. I read my email. I installed a spam filter (what a treat; I am using SpamAssassin and it's so much better than not having it). I have online access throughout my house via strategically placed laptops and other computers. I have both UNIX(-like) and Windows running on various platforms both fixed and portable.

I had the opportunity last week to extend my knowledge of the use of Windows. I had a document that was required to be formatted "nicely" in order that I might distribute it to SAGE members who might wish to see it. For better or worse, the document contained roughly 40 tables and a dozen postscript graphs. Troff is not regarded as the world's best table formatter, so I embarked on implementing something a bit snazzier that would inspire my readers to enjoy the document and its myriad figures.

I had implemented the original document in HTML but found I needed a PDF document. We know that HTML → PDF converters are not renowned for their beauty. I tried to get the document typeset (by someone else) in Quark but initially failed on a quick success due to scheduling and other non-technical reasons. So, I reasoned, Quark can read MSWord files, I'll just format the document in MSWord and then it will be trivially convertible to Quark.

I spent a few minutes writing a short script to extract the document from the HTML so that it could be imported into MSWord and formatted (for the text) and converted to a table (for the tables). After accidentally typing the wrong filename, I learned that MSWord will read HTML directly and do a credible job of laying it out – including tables! I figured my troubles were over.

That was before my ultimate epiphany about MSWord (and, as it turns out, so many of the "tools" under Windows). I found that while MSWord has styles and templates for global formatting conventions, it is difficult to get tables to fall into a consistent format. Each table required personal attention. Even moving the tables to be justified on the right margin was also not possible without clicking on each table and moving it. In fact, I ultimately spent about 25 hours trying to get the document to be passably presentable.

What was the epiphany? Windows has the wrong kind of leverage. I think of leverage as using devices (e.g., computers) or people to increase one's reach or one's ability to solve problems. The Windows tools I was using were insidious. They accepted incremental improvement but never enabled me to improve by an increment of any appreciable size. I could inch closer to my desired solution, but only a little at a time – a snail's pace. "I think I'd like the tables on the right margin," required multiple clicks on each table. Argh!

Why is it insidious? Because it was always possible to move (often to move closer to the solution). At no time did I fail to have some (potentially untried) option or some idea of how to move forward. I have every reason to believe that people who format one table every two months find extraordinary power in the ability to do so with MSWord.

And, to be fair, MSWord works for dozens (hundreds?) of languages. The folks at the international programming contests regularly translate the problems into 65 languages – all using MSWord. This is an incredible feat!

But I hate it. No leverage! More skill or practice only incrementally improves my ability to format tables (in this case). The same seems to be true for any repetitive operation that one would like Windows to perform (e.g., "Please convert these 22 PostScript files to PDF" for our LISA attendees). That's 22 sets of choose-a-file, convert, confirm, and iterate. Ick.

So, I am now familiar with the state of the art. It was an epiphany for me.

apropos

I'll Scratch Your Back . . .

A few years ago I wrote an article entitled "Ready, Set, No?" in which I outlined things that I thought a good manager does for – and doesn't do to – their employees. Apparently it struck a chord with many readers, since I continue to receive comments about it.

Initially, I was surprised at the response, but it just underscores that most of us want a few very similar things from our managers. We want our managers to set the tone for the work environment. They should provide leadership, but also mentoring and learning opportunities to let the employees know that they are valued today, and will be in the future as well. A savvy manager will figure out what matters most to each employee and, wherever possible, make things like work assignments, office arrangements, and face-time hours accommodate those preferences. Managers shouldn't set their employees up to fail with unrealistic workloads, deadlines, or expectations. And when politics, heavy workloads, or other office snafus inevitably rear their ugly head, managers who support their employees, run interference for them, or simply push back a little, will gain employees' undying loyalty and their willingness to go the extra mile – after all, "s/he did it for me."

What about the individual contributor? Are we all created equal? Or can our performance in the workplace vary, and if so, what makes a good worker? I asked a few friends of mine about this and found that, just like managers, there are soft skills that make individuals better to work with and land them consistently on the top of the heap among peers and managers alike.

Whenever the opportunity presents itself to make your boss or co-workers look good, do so! This has multiple payoffs. You'll be rewarded for your individual contribution, but you'll also be recognized for the part you play in giving credit to others. There is usually ample praise to go around, so make yourself look good by making others look good too.

If you want your technical prowess to really shine, share it with others! Those employees who inform others about newsworthy items, teach about and document what they're working on, and mentor peers are much more highly valued than those who don't. If you have a clue, don't keep it to yourself! Teaching and mentoring are forms of leadership and are often rewarded come advancement time.

If you see the wreck coming, telegraph the information ahead of time. No one likes to be the bearer of bad news, but if you feel your manager is going to be blindsided, it's better to give an in-private "heads-up" than let them be publicly caught off guard. No matter how bad the news, they'll appreciate hearing it and appreciate you for giving it.

There's more to most business decisions than simply technical input. Managers are tasked with considering a larger picture, such as risk assessment, profitability, and return on investment. Employees should provide their managers with all the technical information necessary for these big-picture decisions and be willing to entertain give-and-take discussions that involve non-technical considerations. Employees who demonstrate the ability to see the whole picture will be called upon more often for advice, and will ultimately be in a better position to have it adopted.

We expect a lot of our managers, but it's not a one-way street. Make your manager look good, share a clue, and get behind the big picture. The sooner you get on board with these soft skills, the sooner you'll become that all-around employee who has their manager's ear.

by Tina
Darmohray

Tina Darmohray, co-editor of *login:*, is a computer security and networking consultant. She was a founding member of SAGE. She is currently a Director of USENIX.

<tmd@usenix.org>



letters to the editors

;login:

EDITORIAL STAFF

EDITORS:

Tina Darmohray tmd@usenix.org
Rob Kolstad kolstad@usenix.org

STANDARDS REPORT EDITOR:

David Blackwood dave@usenix.org

MANAGING EDITOR:

Alain Hénon ah@usenix.org

COPY EDITOR:

Steve Gilmartin

TYPESETTER:

Festina Lente

MEMBERSHIP, PUBLICATIONS, AND CONFERENCES

USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710
Phone: 510 528 8649
FAX: 510 548 5738

Email: office@usenix.org
login@usenix.org
conference@usenix.org
WWW: <http://www.usenix.org>
<http://www.sage.org>

To John Nicholson:

I thoroughly enjoyed your article, "Politeness in Computing," which appeared in the February 2000 edition of *;login:*. In particular I appreciate the commonsense analogies you have drawn to a home with a welcome mat. You have a gift for explaining difficult technical/legal issues. Have you written any similar articles discussing the legalities of "honeypots"? I am the InfoSec security specialist at my firm. Along with my superiors and the legal department, we are having a lively debate after reading an article by William Jackson in *Government Computer News*, http://www.gcn.com/vol1_no1/daily-updates/19506-1.html.

He basically cites a speech by one Richard P. Salgado, a trial attorney in the [Justice] Department's Computer Crime and Intellectual Property section, in which Mr. Salgado preaches caution. One of our attorneys feels that the entrapment defense is a "pure fantasy" and I am inclined to agree, but others are more cautious. The downside of being cautious is that it tends to make my job of protecting information assets more difficult.

I eagerly await any input you may have.

Regards,

Dave Warde

John Nicholson replies:

Dave –

I'm glad you liked the article, and, just as importantly, my editors are glad you liked the article. It's funny that you should ask about honeypots, because that was the issue that got me into the writing business in the first place. There was an argument a couple of years ago on one of the Security Focus lists about whether honeypots are entrapment or not, and one of the *;login:* editors saw

my response and asked if I would be willing to write a column.

To address the issues raised in the article:

Federal wiretap laws prohibit interception of electronic communications, including traffic monitoring across a network. There are exceptions for network protection, but Salgado said that is an "uneasy fit" for honeypots, because they are set up with the expectation of being attacked.

This isn't entirely correct. If you are the owner of the network, you can monitor what happens on it. You can doubly protect yourself by putting a banner on your login page that says that any use of the network is subject to monitoring, but the key thing that courts have looked at with regard to such monitoring is whether the person had a legitimate expectation of privacy in the communication. I think a judge would have a tough time accepting an argument that someone attacking your network had a legitimate expectation of privacy in his/her attack.

Even if you were only allowed to monitor your network for defensive purposes, I think the honeypot could arguably qualify as a defensive tool. For example, I have a limited budget for physical security at my home. I recognize that there are a number of ways that someone could break in, and I take steps to secure or prevent those. However, if someone is determined to break in, I must recognize that they will find a way. To deal with that possibility, I try to recognize where an intruder might be able to break in, and I have cameras in those areas. If I could only afford a certain number of cameras, I might make one path a little easier or more attractive than the others so that the intruder would take that path and thereby pass in front of the camera, allowing me to gather evidence of the crime. The intruder has already commit-

letters to the editors

ted the crime by being inside the house; the camera simply collects the evidence. By placing a honeypot and monitoring it, you are simply putting an intrusion detector on a place where unauthorized individuals are likely to go, if they are already committing the crime of being inside your network without authorization.

An operator might be held liable for damages if a compromised honeypot is used to launch an attack against a third party. “We don’t know” if such liability would hold up in court, Salgado said.

This is theoretically possible, and I actually wrote another article for *login*: on this subject called, “You’ve Been Cracked . . . And Now You’re Sued.” But if you’re setting up a honeypot, you ought to be sophisticated enough to isolate it and prevent outbound attacks on other networks (or at least either notify those networks that they are being attacked or shut down the attack as soon as it starts). There’s really no excuse for setting up a honeypot and then allowing it to be used as a zombie.

A hacker charged with illegal activities involving a honeypot could argue entrapment, which Salgado said is a difficult defense. He said it might not apply to so-called passive honeypots.

Salgado is correct that entrapment is a very difficult defense. The article doesn’t point out, however, that the defense of entrapment is also only available to someone who is being prosecuted as the result of activity by a government agent (like the DOJ, FBI, or some state or local law enforcement agency). If your company (or client), as a non-governmental entity, sets up a honeypot and a cracker gets prosecuted because of it, the defense of entrapment is not available. See the legal definition of entrapment at <http://dictionary.lp.findlaw.com/scripts/results.pl?co=dictionary.lp.findlaw.com&topic=64/64a96fc79d0ff3a77e4ddea401c7688>.

Furthermore, as Salgado also notes, because a honeypot is a purely passive thing, even if you were a government agent, you are not really inducing or encouraging a potential cracker to go attack it. If you are a government agent and set up a honeypot and then anonymously went to hacker sites and talked about this fantastic server with all kinds of really cool stuff on it and how easy it was to own, etc., etc., then you might be setting yourself up for the defense of entrapment.

Hope this helps. Feel free to write back with any comments or questions.

John.Nicholson@shawpittman.com

Correction

Robert Faust’s article in the last issue (Vol. 27, No. 4, August 2002) contains an error: the last equation on page 41 should read:

$$\text{Received signal strength (non-normalized \%)} = .83(C - 35\log_{10}\sqrt{(A - x)^2 + (B - y)^2}) + 83.891$$

vive la révolution! now get over it!

by **Strata R. Chalup**

President, VirtualNet. Starting as a Unisys 68K admin in 1983, Strata Chalup is now an IT project manager but allegedly has retained human qualities. Her mixed home network (Linux, Solaris, Windows) provides endless opportunities to stay current with hands-on tech.

strata@virtual.net



Let's consider the small-appliance manufacturing industry. I sincerely doubt that anyone goes into that industry expecting to comfortably retire a few decades early by dint of hard work and picking the right company. I also doubt anyone there expects to be lionized in glossy self-referential magazines or considered an "industry elder statesman" for the supreme achievement of starting a can-opener or toaster company and managing not to go out of business for several years.

We need to abandon the fiction of the high-tech industry as "the future." It is not the future. It is merely part of the future, yet another tool in the growing plethora of tools developed by civilization. The revolution is over. Technology won. As technologists, we won. We just may not have realized it yet.

After Enlightenment, Carry Water, Chop Wood

Now you turn to me with incredulity and say, "How can you say that we 'won' the computer revolution?! For Turing's sake, more people than ever are using Windows, the Internet is full of spam, Congress is passing all these ridiculous laws about stuff they clearly don't grok, and SAGE is still chafing under the iron yoke of USENIX! Okay, maybe that last one isn't the big deal some people make it out to be, but what about the other stuff?!"

To this question I must pose another: what happens when you win a revolution? There are several large-scale effects, of which two are most noteworthy. Firstly, the patriotic militia now must lay down its arms and go back to being shopkeepers, farmers, and draymen. Think of the American Revolutionary War (or Rebellion, with a cordial nod to our neighbors across the Pond). The minutemen who survived went back to what was left of their shops or farms and got back to work. Not very glamorous.

Secondly, and perhaps more applicably, when society-at-large accepts the validity of a general concept, that acceptance is far from complete and universal; it is often without full comprehension and always comes with some highly vocal dissenters. In sports terms, players take the ball and run with it – but probably out of bounds and often along a playing field that looks nothing like what the game's originators imagined. General societal adoption of technology takes "our ball" and goes off to play with it according to larger rules: in particular, with adaptation of technology into more specialized tool sets and re-purposing of technology and technological infrastructure for previously unfeasible uses – including those which we, as de facto technocrats, may feel lack a certain elegance or relevance.

Which Came First: The Goose or the Golden Egg?

To make matters more confusing, our technological revolution has largely become conflated with a minor economic "revolution" – in the old-fashioned sense of "a turn of the wheel." The economic boom cycle came to be considered part and parcel of the technological progression. The success of the technology boom cycle was the widespread normalization of an unprecedented level of technological sophistication. Today's high school kids tweaking their Doom skins have more power to do graphics visualization than the entire MIT astrophysics department had in 1981. Wow. The guy on the street doesn't think twice about this anymore. Double wow. To us it seems to matter that the kid may be running Windows instead of Linux, but in the larger context, the victory is that he or she is using a sophisticated tool and not thinking about

Technology has everyday value only when it is *applied*.

OS or hardware issues at all. Let's not forget the forest as we argue about which trees are better, and which types of woodlot management are best.

Did you know that not only your box-o-hardware but your OS and your applications are now commodity items? Why do you think Microsoft wants to turn on system auditing with mandatory updates in Windows 2000 (SP 3, read the EULA) or application licensing in XP? Commodity, commodity, commodity. Applications are just conduits for file formats, which in turn are just a convenient box with handles around *content*. "Ptht!" you say. "Content is king? Been there, done that, no market in it." Not quite true – there's no content market with pie-in-the-sky valuations that will catapult a company to stardom. What there is, instead, is a vast and insatiable long-term income stream. The big companies are spending millions twisting our legislative structure to support this income stream, this multi-billion-dollar market for those who aggregate intellectual property rights and copyrights to content.

Day-to-day technology is rapidly becoming merely a conduit to deliver content to those who require it. This mirrors the business world, where technological applications have been largely harnessed to make existing business processes easier. Everyone who has used it has a Meeting Maker™ horror story, yet for the corporation overall it still makes a positive difference. The success of the PDA is largely rooted in synchronization, and could not come to fruition until there was enough electronic infrastructure carrying out business processes that there was something on the other end to synchronize *to*.

I Will Gladly Pay You Tuesday for a Hamburger Today

This brings us to another key point. What many of us in the technology field tend to forget is that technology has everyday value only when it is *applied*. Further, Moore's Law gives technology something of the nature of a perishable good or even a service. This may seem counterintuitive, given the tangible nature of a rackmounted box stuffed with hard drives and application servers. The economics of perishable goods and of services are rather well understood. Empty hotel rooms are not bankable. The high prices with which we are familiar, and upon which empires were built, came largely from relative scarcity of technology applied in the right place at the right time. When you just *had* to have a 100GB RAID 5 file system to support your e-commerce application, and the server box only had so many disk drive slots, a 10GB 9600 rpm drive was priceless. Especially when time to market was perceived as *the* most critical factor, and a dozen other dot-com companies were trying to pry the same pallet of in-stock drives out of your local vendor's warehouse. What if there weren't a dozen other companies trying to buy the same drives? Or fast-forward six months to a year, when the same dollars buy you a 30GB drive. Those 10GB drives sitting on the shelf still cost the same to produce but their dollar value has dropped. Yet we consistently see business plans that insist that the price should have nothing to do with the current situation, and that the "value" of the drives is \$whatever.

This is an understandable, yet common, mistake. I had hopes that an MBA and a little bit of history would have proven more of a deterrent. Most of us think of technology prices as intrinsic rather than situational. The cost of manufacturing technically does not include research and development, despite our vendor's sales team's assurances to the contrary. Hence the thriving clone markets and the profitability therein. They're not building empires, but they're paying salaries and staying in business. It's a brave new world to the people who entered the workforce in the late '80s, but it's "back to reality" for the rest of us. In my most cynical moments I think that the industry will

Plenty of successful companies are making can openers. They don't have stock with valuations reflecting obscene P/E ratios.

resist normalization until everyone who remembers the glory days of IBM and DEC retires from positions of responsibility. We may have to broaden that to include Sun, Microsoft, and Oracle. The time of empires has passed. When the market was smaller, you could corner it. Who has cornered the market on butter? On best-selling novels? On gasoline?

It's the End of the World As We Know It . . .

Much of the technology sector is waiting for investors to "come to their senses." Sorry, they already have! Except for random islands of scarcity produced by the market equivalent of "lift pressure," the high flying days are over. Plenty of successful companies are making can openers. They don't have stock with valuations reflecting obscene P/E ratios. After all, they're making can openers. Welcome to the commodity marketplace.

In many ways the burst of the dot-com bubble was like Watergate. Its worst aspect was not the individual investments or reputations destroyed, but the damage to the societal perceptions of the institutions behind them. There are still many things that we can do with the new technologies, but the man-on-the-street perception in much of our society today is that *all* technology is over-hyped and that there's nothing out there but Web site pyramid schemes and pop-up ads.

There are still good technology stocks out there. They're the kind that will help fund your retirement in 40 years, not the kind that will buy you a house in five years. The familiar proverb is missing a key modifier – it's not that "there's no such thing as a free lunch"; it's "there's no such thing as a *consistently* free lunch." If you're used to having Peets or Starbucks in the break room, "Office Caterer House Blend" seems like a deep pit rather than a return from the mountains to sea level.

. . . And I Feel Fine

Do you remember life before Post-it™ Notes? A lot of us can't imagine being without them. Their invention didn't "make" 3M, nor did they become 3M's flagship product against which all other products were measured. Their inventor didn't leave 3M and try to start his own company to propel himself into stardom and riches. There is still a lot of potential out there. It's just *traditional* potential, which can feel a lot like "nothing" in contrast with the excitement of the past couple of decades.

So where is this all going? Is the high-tech sector dead? Should we all just take up dog walking or open a hot dog stand? I think that the end of the Age of Empires in high tech has a lot of potential for us as a profession. Work can still be fun, which is good because you won't be retiring at 35. Opportunities are out there, and while they aren't as glamorous, they may not demand as high a personal price in unpaid overtime and ulcers.

There are still next-level paradigm jumps out there. They may not make you rich, but who cares? As the late Michael Dertouzos said at a futurism forum in 1999, "We haven't really invented the bulldozer yet. We're just out there with platinum, diamond-encrusted shovels." I think that there's the potential for a New Internet that looks a lot like the Old Internet, a place where innovation was exciting and people did it for fun, and innovators weren't punished with a four-digit surprise bandwidth bill from their ISP. New advances might make hardware hobbyism possible on a level that would remind old-timers of the heady days of the '70s when "home computing" meant breathing solder fumes. Details, details, you say. But that's the next column. See you next issue!

software licensing

101

We've all seen them – pages and pages of two-column text, generally in 8-point type. In real life, we might get a CD in a jewel case with lots of small text printed on it and a warning that says that opening the box means that you accept the license. When we're online, we get an easy "I Accept" box that doesn't require reading all the gibberish before letting us install the software that we want. But have you ever actually read one? What are you agreeing to? Software vendors have lawyers who spend hundreds, if not thousands, of hours drafting these licenses to protect the "rights" of the seller. And in the case of Microsoft, Adobe, AutoDesk, Borland, and several others, they also have the Business Software Alliance (<http://www.bsa.org>) to collect reports of and prosecute unauthorized software use.¹ The only thing protecting your company is your understanding of the terms of the license and your (or your lawyer's) ability to negotiate with the vendor.

The purpose of this article is to give you some insight into what a software license should cover, provide an analysis of some sample language that vendors have used, and give you some tips for negotiating with vendors.

What Is a "License"?

The word "license" is somewhat misleading. A software license is a contract. According to WhatIs.com, "An End User License Agreement (EULA) is a legal contract between a software application author or publisher and the user of that application. The EULA, often referred to as the 'software license,' is similar to a rental agreement; the user agrees to pay for the privilege of using the software, and promises the software author or publisher to comply with all restrictions stated in the EULA."² So, in exchange for being allowed to use the software, you agree to comply with the restrictions imposed by the vendor. If you break the rules, you have breached the contract, and there may be specific remedies written into the license or you may have even violated some laws (e.g., copyright laws,³ the Digital Millennium Copyright Act,⁴ the No Electronic Theft [or NET] Act,⁵ or others).

License Structure and Analysis

A software license tells you what you can and can't do with the software and what the vendor can and can't do. What the license says (and doesn't say) is just as important to you as it is to the vendor. In general, a software license should cover:

- Software included under the license
- Scope of use for the software and any restrictions on its use, as well as any different restrictions on the use of the documentation
- Duration of the license
- Related services (e.g., consulting, enhancement, help-desk support) that will be provided, and the terms under which those services will be provided
- Pricing and payment terms
- Confidentiality provisions
- Warranties and indemnities
- Limitations on liability
- Termination of the license and/or the services
- Other legal terms that are relevant (e.g., rights of publicity, choice of law)

by John Nicholson

John Nicholson is an attorney in the Technology Group of the firm of Shaw Pittman in Washington, D.C. He focuses on technology outsourcing, application development and system implementation, and other technology issues.



John.Nicholson@ShawPittman.com

Frequently, software licenses drafted by vendors will try to limit software use to employees of the company purchasing the software.

If the software is going to be used internationally, there may be specific terms that need to be discussed. For example, the US government restricts certain software from being exported to certain countries or at all, and you might have to apply for a special permit; or there are certain international treaties that may come into play regarding the sale of goods.

SOFTWARE INCLUDED UNDER THE LICENSE

The first question to ask when looking at a software license is whether it covers all of the software (and all of the functionality) that you think you are getting. It's important to make sure that you understand how the software covered by the license will provide *all* of the functionality that you think you are getting, and that it will continue to provide the same functionality in the future. Although this may seem obvious, there are certain provisions related to updates and upgrades that can affect the availability of functionality down the road. (See the section on updates and upgrades, below.) When you are discussing the software covered by the license, be sure to review in detail how the software will provide the functionality that you think you are getting. If the software cannot currently provide some of the functionality that you need, see whether there is a workaround, whether the vendor plans to provide such functionality in the future, or whether you will need to develop that functionality in cooperation with the vendor, using a consultant or on your own.

SCOPE

The scope of a license can vary dramatically, depending on the restrictions that the vendor wants to place on your use of the software. The license may specify:

WHO CAN USE THE SOFTWARE

Frequently, software licenses drafted by vendors will try to limit software use to employees of the company purchasing the software. This would mean that neither your contractors nor any consultants could use the software, which could be an inconvenience for you (if you complied with the restriction) or could result in a breach of the license (if you didn't). If you are only using object code, there isn't really much reason for this restriction. If you have source code, it would be reasonable for the vendor to insist that contractors and consultants sign a Non-Disclosure Agreement before being allowed access to the source code. And there might even be a short list of direct competitors that the vendor would want to prevent from seeing the source code under any circumstances.

FOR WHOSE BENEFIT

Software vendors want to sell as many copies of their software as possible, so they try to limit the number of people who can benefit from each installed version of the software. You, on the other hand, want to buy as few copies of the software as possible, so you want to be able to use it for your own company, your subsidiaries, other affiliates, your customers, your suppliers, etc. As long as you and the vendor come to an understanding regarding how the software will be used prior to executing the license, the vendor can take the scope of your use into account in its pricing.

FOR WHAT PURPOSES

Just as vendors want to limit access to the software by other companies, vendors want to limit access across business lines, if possible. By understanding how your company will want to use the software, you can reach a reasonable agreement with the vendor in this area.

IN WHAT LOCATIONS

If you plan to use the software in multiple countries, some vendors will try to argue that they need to get approval from regional segments of their organization in order to negotiate a deal. How the vendor organizes its own accounting is not your problem, and if a vendor wants to be a global player, it should act like one. (Note: this is just an extension of the “no authority” strategy discussed in the “Negotiation Tips” section later in the article.) It is possible, however, that there may be countries in which, for some legitimate reason (generally legal or regulatory), a vendor is unable to license you to use the software.

ON WHAT EQUIPMENT

Certain licenses are restricted to a particular-sized processor. Be certain that the license you are getting is sufficient for the use you intend now and throughout the term of the license. You will not put yourself (or your successor) in a good negotiating position if your business needs to renegotiate the license in the middle of the term because you did not adequately anticipate growth levels. If the pricing for the license is based on the processor (or other equipment) that you will be using, and there is a possibility that you might need to upgrade or expand your equipment during the term, negotiate the pricing for using the software on the upgraded/expanded equipment before signing the initial deal. Remember, your negotiating leverage is at a maximum before you sign.

If the license is based on number of users or number of machines accessing the software, you should consider whether devices like PDAs would qualify. For example, one vendor’s standard license prices the software by “client,” which it defines as “an application that invokes, typically via a network protocol, the software functions provided by one or more servers.” Under that language, a query from an application installed on a PDA could qualify as a “client,” which could dramatically increase the price of the software.

PERMITTED COPYING

Vendors frequently include language such as, “Customer shall not make any copies of the Software, except for a single archival copy solely for internal purposes.” This means that, for example, you would have to separately license development, test, and production instances of the software, and that you might also need to separately license any copies to be used for training. Since you must have at least a test region in addition to your production region, the license should include copies for you to use in those additional regions without an additional charge. Furthermore, if you have multiple locations, each running the software locally, you might want to have a backup copy at each location so that you can re-install via the local network. This is also a reasonable request.

Another thing to look out for in software licenses is whether the defined term “software” includes the documentation. Frequently, it does, and the impact of that definition combined with the language quoted in the previous paragraph would mean that you could not copy any of the documentation associated with the software in order to make training materials or procedures manuals. Training materials and other documentation provided by the vendor are likely to be generic rather than customized to your particular implementation. To customize your documentation, you might want to make sure that the license includes language such as, “Customer may copy, share, distribute, modify and create derivative works from the user manuals and any related documentation solely for Customer’s internal business purposes.”

Certain licenses are restricted to a particular-sized processor.

WHETHER THE LICENSE IS TRANSFERABLE OR NOT

Frequently, a license will specify either that (1) the customer may not assign the license without the vendor's consent or (2) neither party may assign the license without the other party's consent. If your company is acquired or wishes to assign the license to another company (including a parent, affiliate, or subsidiary) for some other valid business reason, this provision could allow the vendor to refuse to grant its consent unless you satisfy conditions specified by the vendor. Language like this should be revised so that you can transfer the license to a parent, affiliate, or subsidiary without being required to get the consent of the vendor. At the least, you should get the vendor to agree that it will not unreasonably withhold any consent.

DURATION/TERMINATION OF THE LICENSE

The license should specify the term of the license. Large system operating software is generally priced by the month or the year. In the US, a perpetual license is legally acceptable, but outside the US you should discuss with a lawyer using a term of 20 years or less in order to avoid certain legal issues related to perpetual contracts.

In addition to knowing when the license will expire, you should examine very carefully the ways that the license can be terminated. A piece of software can be critical to your company's operations. Unless you don't pay for the license or you knowingly disclose the vendor's legitimately confidential information, the vendor should not be able to terminate the license. There are legal (i.e., financial) remedies that can compensate the vendor for virtually anything else that your company could do without threatening your company's ability to use a potentially mission-critical piece of software. On the other hand, you should be able to terminate the license and associated maintenance as long as you provide a reasonable amount of notice to the vendor.

Frequently, vendors will agree to allow you to terminate a contract early provided that you give them notice and pay a termination-for-convenience fee.⁶ These fees range from a reasonable recovery of costs by the vendor (such as any investments that the vendor had to make to provide your particular services or to develop a particular function) to bordering-on-absurd attempts to include all of the revenues that the vendor would have received during the term of the contract. If you terminate the contract early, it isn't unreasonable for a vendor to recover costs that the vendor expected to recover during the term. The vendor shouldn't suffer harm because you elected to terminate a contract early. However, once you terminate the contract, the vendor won't be providing services or incurring costs, so there is no reason for the vendor to receive the total revenues that the vendor would have received for providing the services during the remainder of the term.

Nor is there much ground for the vendor to argue that it should receive the profits that it would have received during the remainder of the term. Vendors will try to use the argument that they have a right to lost profits because they shouldn't suffer harm because you decided to terminate the contract early. If the contract had never happened, the vendor would not have received the profits that they are trying to claim, so allowing the vendor to include lost profits in a termination-for-convenience charge puts the vendor in a better position than if the deal had never happened. The vendor should be allowed to recover costs, but not lost profits.

PAYMENT, ACCEPTANCE TESTING, AND WARRANTIES

In general, the vendor wants its money for the software right away (particularly these days). When negotiating, vendors look for payment on delivery (if not sooner) and in

a lump sum. For example, one vendor's standard license agreement specifies, "Within 10 days after the Effective Date of this Agreement, the applicable Order Form and payment to [Vendor] of the associated license fees, Support Fees or other fees set forth in the Order Form, [Vendor] shall provide access to the [Vendor] FTP site to enable licensee to download the license Products and Documentation." This means that you have to pay for everything up front and only then will you be able to download the software and documentation. For something really simple and basic, this might not be too bad.

But for more complicated software that will require some implementation, you should be able to delay at least some of the payment until the software is successfully implemented (i.e., after acceptance testing). Vendors will argue that acceptance testing isn't necessary because (1) the product is proven in the market and (2) you're getting a warranty. However, the warranties frequently are effective for 90 days after *delivery* of the software. Most implementations take longer than 90 days, so unless the implementation is either simple or on a fast track, your warranty is effectively useless if it is based on the delivery date. Your warranty should begin running after the software has been successfully installed and tested (including integration testing), and you should be able to return the software for a full refund of the license fees if it isn't accepted.

Vendors want to be able to recognize the revenue from the sale of the software as soon as possible, and allowing you to return the software can interfere with the vendor's ability to recognize the revenue. Revenue recognition affects the vendor's earnings and the salesperson's incentive compensation. There are very specific rules regarding when revenue can be recognized.⁷ If revenue recognition becomes an issue in your deal, you should have someone from your accounting/finance group who is experienced in this area working with your team.

For ongoing services or installment payments, you should look at the payment terms specified in the license. For example, one vendor's standard terms states:

All billed charges are due ten (10) days from the invoice date. Licensee agrees that a copy of an invoice received by facsimile machine shall be binding on Licensee and have the same effect as an original. All balances ten (10) days past due will be subject to a ten (10) percent annual finance charge, and [Vendor] may elect to suspend technical support, software updates and enhancements, withhold shipment of computer supplies, and/or activate time sensitive devices . . . until [Vendor] receives said past due payments.

Not only is "Net 10" a short period of time for your company to make its payments, but the multiple remedies to this vendor of being able to charge interest, withhold support or other services, *and* potentially disable the software for failure to pay a bill within 20 days of the date the invoice was printed (not received by your company) is unreasonable. Because the license specifies an interest charge for late payments, the vendor is already being compensated for a late payment.

REMOTE DEACTIVATION OF SOFTWARE

As mentioned in the language quoted in the previous section, vendors sometimes include in licenses the right to switch off the software or deny you access to date-sensitive activation codes if you haven't paid for something or if the vendor thinks that you are in breach of the license. For example, one vendor's standard form license states:

You should rarely, if ever, accept a license that allows the vendor to deactivate any part of your software with time-sensitive disabling devices.

Licensee acknowledges that some or all of the [software] may contain time sensitive devices which may be activated automatically, by [Vendor] or otherwise upon a material breach of this Agreement by Licensee, including without limit Licensee's breach of the payment terms . . . and Licensee's breach of the confidentiality provisions . . . or the expiration or termination for any reason of this Agreement. Upon activation, such time sensitive devices may alter or prevent the functionality of the [software]. Licensee acknowledges and agrees that such time sensitive devices are necessary to protect [Vendor's] intellectual property rights, that [Vendor] shall have no liability whatsoever for any outcomes caused by activation of such time sensitive devices and that Licensee shall be liable for all costs associated with the activation of such time sensitive devices, as well as costs associated with resuming normal use of the [software].

You should rarely, if ever, accept a license that allows the vendor to deactivate any part of your software with time-sensitive disabling devices. If the vendor can deactivate your software, you will likely give in to the vendor's demands rather than lose your functionality. Under the license, and under the law in general (at least in the US), the vendor will have legal remedies for any breach; the ability to remotely disable the software represents a self-help measure that gives the vendor unreasonable leverage over your company.

Maintenance and Support

GENERAL

The standard vendor position regarding maintenance and support is that it begins after the warranty period (if any), is payable in advance, is non-refundable, and is subject to annual increases that are not capped in any way. From your perspective, you want the maintenance and support to be optional, since you might not need it or might be able to get it from someone else; you want to be able to pay in installments; and you want the price to be fixed during the term of the license, or if not fixed, then at least have the increases specified in advance and capped, to ensure that you understand the maximum amount of costs that you will have to pay.

NEW RELEASES, UPDATES, AND UPGRADES

Frequently, vendors will include new "releases," "updates," or "maintenance patches" in the cost of maintenance but will expect you to pay for "upgrades" to new "versions." Each vendor uses these terms differently and, frequently, in confusing ways. You should understand how the vendor numbers its software, and agree in advance to what constitutes something covered by maintenance (e.g., going from version 3.1.1 to 3.1.2) versus something for which you'll have to pay (e.g., going from version 3 to 4). It's reasonable for you to pay for significant new functionality, assuming that you want it and can use it, but it seems unfair for a vendor to require you to pay for the privilege of going through an upgrade to receive new functionality that you may not want or, in the alternative, risk losing your maintenance.

To minimize their costs of providing support, vendors try to limit the number of releases that they have to support at any one time. They do this by requiring you to upgrade to within a certain release level (e.g., "n-1") or by specifying that support for a previous version will be discontinued some period of time after the next version is released. On the other hand, upgrades are unpleasant and expensive for you. The more frequently you have to upgrade, particularly where an upgrade provides some change in the user experience, the less happy your users will be.

One vendor's standard license states:

[Vendor] periodically issues software updates that may require additional processing capacity (i.e., CPU memory or additional disk capacity) for Licensee's equipment. Licensee understands this requirement and agrees to purchase equipment as needed to remain current with [Vendor's] software releases. Licensee agrees to install each update and enhancement as soon as reasonably possible but in no event later than ninety (90) days after receipt. In the event that Licensee fails to so install any update or enhancement, then any warranty or obligation of [Vendor] with respect to the affected Program shall be invalidated.

The impact of the quoted language means that if the vendor releases an upgrade that required more processing capacity than your existing hardware had available, then within 90 days of the new release being made available, you *must* (1) upgrade your hardware to the specifications for the upgrade and (2) install the upgrade. Think of the impact that this could have on your budget and your upgrade schedule for your hardware.

Vendors frequently will not commit to a specific number of releases or the frequency with which they will produce new releases. It's not unreasonable to insist on a maximum number of required upgrades in a given year, nor is it unreasonable to expect vendors to keep up with industry-standard hardware/OS upgrades and industry-specific regulations.

The Legal Stuff You Usually Ignore

There are whole sections of contracts that business people regularly ignore because they feel like those sections cover stuff that is only interesting to lawyers. Although the lawyers are usually the only ones who argue about these issues, the business people really ought to pay more attention to them. Things like confidentiality, warranties, limitations of liability, and other terms that are always at the back of the contract can have a significant operational and financial impact; frequently, the vendors put things back there assuming that the business people won't be paying close attention.

CONFIDENTIALITY PROVISIONS

Some vendors' standard licenses define every aspect of their software and documentation to be confidential information. While it is not unreasonable for a vendor's source code or detailed technical documentation to be confidential, making the object code or user manuals confidential could create a problem. For example, one vendor's standard form contract states:

The parties agree . . . that they shall not use, except as otherwise expressly permitted hereunder, or disclose to any third person, including any of its affiliates, or to any employee of the receiving party without a need to know, either during or after the term of this Agreement, any Confidential Information.

If the definition of "Confidential Information" were to include the object code or user documentation (which this vendor's license does), this would mean that except for employees of your company who actually have a need to see the software, no one else could even see the screen while the software was being used without it being a violation of this term.

Some vendors' standard agreements also have confidentiality provisions that are "one-way": they protect the vendor's information but not yours. If your company is hiring

It's not unreasonable to insist on a maximum number of required upgrades in a given year.

the vendor to implement the software or provide any other services, the vendor will be learning a great deal about how your company operates and would be under no obligation to keep any of it a secret.

WARRANTIES AND INDEMNITIES

High on the list of things that non-lawyers traditionally ignore are the warranties and indemnities. In addition to the warranty that there are no disabling devices included in the software, the big one from your perspective is that the vendor should warrant that the software and documentation do not violate any patents or otherwise infringe on any other intellectual property rights. If someone claims that the software or documentation does violate their intellectual property rights, the vendor should indemnify your company for all costs and expenses associated with that claim.

LIMITATION OF LIABILITY

Next time you drop film off to be developed, look at the disclaimer on the envelope that says that if the developer screws up your pictures or even just completely loses the film, all they have to do is give you a new roll of film. This is a limitation of liability. A limitation of liability provision is a contractual term that specifies the maximum amount of damages for which a party can be liable under the contract.

In most standard vendor licenses, the limitation of liability generally only limits the liability of the vendor and limits it to an amount equal to something like the cost of the software or one year's maintenance. This means, in general, that no matter how badly the vendor screws up or how much it costs your company to fix the problem, the vendor only has to pay your company a maximum of what is specified in this clause.

Whether the cost of the software or the price of a year's maintenance is a reasonable limitation on damages is something that you and your business people can evaluate and negotiate. This provision should also be mutual, so that your company's damages are also capped.

There are certain standard exclusions to this limitation that are generally included in each contract without too much argument from either side. If you're really interested, your general counsel can tell you more.

OTHER LEGAL TERMS

There are frequently a few other sections at the back of any agreement covering things like "choice of law" and "right of publicity" and other terms that sound trivial, and sometimes they are. However, if you are not careful, your agreement could be governed, for example, by the law of Virginia, which has passed a very vendor-friendly version of the Uniform Computer Information Transactions Act (UCITA).⁸ New York is a state that has not passed a version of UCITA and is generally considered to have a well-developed body of commercial law.

Vendors like to publicize their sales. Some companies like to keep very close control over how their name and trademarks are used. Frequently, vendor licenses include the right for the vendor to issue a press release announcing that you are a customer and to use your company's name in their customer list. Your company may have a policy regarding publicity, and if you don't pay attention to the provisions at the end of the license, you could end up agreeing to a contract that violates your company's policy.

Modifications and Customizations

What happens if the software you are buying doesn't do *exactly* what you need it to? The first question is whether you're going to do something about it or just live with the lack of functionality. If you want to do something about it, then you will either need to modify or customize the software. "Modifications" to the software are modules developed by you or a third party that use pre-defined APIs in the software but do not require any changes to the underlying source code. "Customizations" are changes to the underlying source code, made by you, the vendor, or a third party.

Some vendors will encourage you to make modifications to their software, or they may offer to either assist you with developing them or develop them for you. Be careful, though, because some vendors' licenses give them the rights to any modifications you develop.

If you plan to develop customizations, you will need access to the source code for the software. This is not usually provided as part of a standard license. Some companies provide access to source code as part of a "developer's license" that is more expensive than the standard object code license.

Customizations can create risks, though, and the biggest is that a customization will take you out of the upgrade path (this is sometimes true with modifications, too). The best way to make sure that a customization or modification will not lock you into a particular version is to have the vendor develop the customization for you, but negotiating a development and implementation deal is a subject for another time.

Negotiation Tips

The most important thing in any technology procurement is to understand what business need the technology solution must satisfy and what that is worth to the company. Once you understand why you are looking for in a technology solution, the negotiation can begin.

As with any deal, negotiating software licenses is all about who has leverage, and, until you sign the deal, you have the most leverage you are going to have (in some cases, this is still very little, but once you sign the deal you have even less). Frequently, customers are reluctant to negotiate, believing that if they don't make things difficult for the vendor at the negotiating table, the vendor will take that into account during the relationship. It's a nice thought, but this almost never happens. By that logic, you would agree to pay the sticker price for a new car in the hopes that you would get better service from the dealer in the long run. It's important to realize that, in general, a vendor's standard initial position is a position that is very favorable to the vendor. It's simply the opening position in the negotiation. One of the most important aspects of a negotiation is to know what is reasonable and why. Hopefully, this article has given you some insight into that.

Both comparison shopping and getting references are reasonable things to do in any procurement. If there are multiple packages that can provide the functionality that you need, a competitive procurement increases the likelihood that you will get a better deal from whichever vendor you ultimately pick. Also, discussing your needs and a vendor's offering with one vendor will give you ideas for additional questions to ask the other vendor(s). You should also ask the vendor for references, including at least one customer that has stopped using the vendor's software recently. If the vendor has a users group, contact them, as well, and ask them for references similar to your company's

profile and for a former user of the software. References can give you valuable information about issues with the vendor's software, how well the vendor supports the software, and the vendor's plans for the future.

When it comes time to negotiate, most vendors will try to send someone who does not have negotiation authority. He or she will agree to take your positions back to the vendor, particularly to the vendor's legal department, for review. It's the same principle that car dealers use – the salesman actually has no authority and acts like he is on your side, and it's just his manager who's being inflexible and unreasonable. Try to make sure that the person you are dealing with has the authority to agree to changes in the terms of the license. If they don't, then you're wasting your time negotiating with them.

To the extent possible, you should avoid using the vendor's standard form license. Any document can be written any number of ways, and software vendors have had lots of practice at drafting agreements that are interpreted in their favor. If you do have to use the vendor's form, have an experienced attorney look at it to identify areas of concern. Vendors are very good at writing provisions that sound very reasonable but that can have serious consequences to the consumer. You may not be able to do anything about them in negotiations, but at least you will understand the risks of a particular deal.

The common arguments that vendors will make when resisting changes to their form are that the changes will adversely affect their pricing model (i.e., they'll have to charge you more), that they will affect the vendor's ability to recognize the revenue, or that the vendor will have to make special accommodations to manage your contract if you get terms that are different from everyone else. The only one of these that holds any water from the customer perspective is that some provisions of a software license allocate risk (for example, a warranty pushes the risk of a failure onto the vendor), and if you change the allocation of the risk, you may actually legitimately change the assumptions underlying the vendor's pricing. However, vendors tend to use this as an argument much more often than it is merited. If the vendor argues that a change will affect the vendor's ability to recognize the revenue from the sale of the software, ask to see the opinion of the vendor's accountant.

Conclusion

If all of the above wasn't enough to make you want to look through your software licenses with a pair of lawyers and a fine-tooth comb, here's a recent quote from a Microsoft EULA for the patch that fixed a security problem in Windows Media Player:

You agree that in order to protect the integrity of content and software protected by digital rights management ("Secure Content"), Microsoft may provide security related updates to the OS Components that will be automatically downloaded onto your computer. These security related updates may disable your ability to copy and/or play Secure Content and use other software on your computer. If we provide such a security update, we will use reasonable efforts to post notices on a web site explaining the update.⁹

By downloading and installing that security patch, which you really needed to do, you would give Microsoft the authority to automatically dump software onto your machine, and the only thing they would have to do would be to make a reasonable effort to post a notice about it somewhere on a Web site.¹⁰

Your software is letting your company do its business, and the licenses are what control how you use that software. Understanding the reasons why your company wants to use a particular software and the terms and conditions governing its use are critical issues if you are going to be involved in negotiating or administering software licenses. At the very least, read the licenses of the software for which you are responsible so that you can have a clear understanding of what you are and aren't allowed to do.

Notes

1. This article provides general information and represents the author's views. It does not constitute legal advice and should not be used or taken as legal advice relating to any specific situation.
2. <http://whatis.techtarget.com/under EULA> (as of Sept, 3, 2002).
3. US Code Title 17.
4. PL 105-34 (1998).
5. PL 106-113, 113 Stat 1501, 1501A-521 (1999).
6. Note that there is a difference between "termination for convenience" and "termination for cause." Termination for cause is when the vendor has breached the contract in some way and you are firing the vendor. Termination for convenience means that you have simply decided that you don't want to use the software or the services any more.
7. American Institute of Certified Public Accountants Statement of Position 97-2.
8. VA Code §§ 59.1-501.1 through 59.1-502.1. Maryland is the only other state so far that has passed a version of UCITA, but it is being considered in a number of other states. Three states, Iowa, North Carolina, and West Virginia, have enacted "bomb shelter" anti-UCITA statutes to protect their citizens from the effects of UCITA provisions in shrink-wrap or click-wrap licenses.
9. As quoted in Thomas C. Greene, "MS Security Patch EULA Gives Billg Admin Privileges on Your Box," posted at <http://www.theregister.co.uk/content/4/25956.html>, as of 7/01/02.
10. For a humorous take on this, see J.D. "Illiad" Frazer's comic strip, "User Friendly," for July 6 and 8, 2002, at <http://www.userfriendly.org>.

C99 and compatibility

by **Glen McCluskey**

Glen McCluskey is a consultant with 20 years of experience and has focused on programming languages since 1988. He specializes in Java and C++ performance, testing, and technical documentation areas.



glenm@glenmcl.com

In previous columns, we've looked at some of the new features in C99, the standards update to C. In this presentation we'll discuss compatibility and look at issues with mixing C89 (the previous C standard) and C99 code. We'll also look at compatibility between C99 and C++.

C99 and C89

Let's start by stating what is probably obvious: if you use new C99 features in your C programming, you should not expect your programs to compile with an older C89 compiler. Here's an example:

```
#include <stdio.h>

struct A {
    int x;
    int y;
};

struct A a = {
    .y = 37,
    .x = 47
};

int main()
{
    printf("%d\n", a.x);
}
```

This program uses the designator feature of C99. When I compile the program as C89, the result is:

```
"e1a.c", line 9: error: expected an expression
    .y = 37,
    ^
"e1a.c", line 10: error: expected an expression
    .x = 47
    ^
2 errors detected in the compilation of "e1a.c".
```

Another example is the use of interspersed declarations and statements:

```
#include <stdio.h>

int main()
{
    int x;
    x = 37;
    int y;
    y = 47;
    printf("%d %d\n", x, y);
}
```

This feature was borrowed from C++ and added to C99.

You can't use C99 features with a C89 compiler, but what about going the other way? What happens if you try to compile a C89 program with a C99 compiler?

For example, consider the following program:

```
#include <stdio.h>

int main()
{
    static x = 37;
    x = g(x);
    printf("%d\n", x);
}

int g(int x)
{
    return x + 10;
}
```

This usage is legal C89, but not C99. The declaration:

```
static x = 37;
```

leaves off the type (int), and the statement

```
x = g(x);
```

calls an undeclared function. The C99 standard tightened up both of these areas. Requiring that a function be declared before use catches a certain class of errors, such as passing a wrong argument type.

Another example of valid C89 usage that is invalid C99 concerns the use of keywords. For example, restrict is a C99 keyword, so this program is no longer legal:

```
#include <stdio.h>

int restrict = 37;

int main()
{
    printf("%d\n", restrict);
}
```

Other new keywords include `inline`, `_Bool`, `_Complex`, and `_Imaginary`. There are also many new library functions, which may conflict with existing functions in C89 programs.

A third example is failure to specify a return value:

```
int f()
{
    return;
}

int main()
{
}
```

This is valid C89 but invalid C99.

Beyond a few areas like this, C89 programs should work with a C99 compiler.

C99 and C++

The C++ language was designed on a C base, and in the early days there was emphasis on trying to keep C++ compatible with C, so that C programs could be compiled as C++ code. Since that time, C and C++ have both diverged and converged, and compatibility between them is a complicated issue.

The first point is similar to what we said above about using C99 features with a C89 compiler. There are a great many C++ features that have no equivalent in C99. One example is the C++ template feature:

```
#include <cstdio>

using namespace std;

template <class T> T min(T a, T b)
{
    return a < b ? a : b;
}

int main()
{
    int x = min(37, 47);
    printf("%d\n", x);
}
```

This usage has long been part of C++ but is unknown in C. Another example is function overloading:

```
#include <cstdio>

using namespace std;

void f(int i)
{
    printf("f(int) called\n");
}
```

```
void f(double d)
{
    printf("f(double) called\n");
}

int main()
{
    f(37);
}
```

The specific `f()` to call is determined based on the argument type. Again, there's no C equivalent to this feature.

Just as there are C++ features not known to C, there are C99 features not part of C++. For example, C99 mandates a long long type:

```
#include <stdio.h>

long long x = 0xffffffffffffffffull;

int main()
{
    printf("%llu\n", x);
}
```

Many C++ compilers allow this feature, but if I compile the code using strict conformance compiler options, the result is:

```
"e4a.c", line 3: error: the type "long long" is nonstandard
long long x = 0xffffffffffffffffull;
    ^
"e4a.c", line 3: error: the type "long long" is nonstandard
long long x = 0xffffffffffffffffull;
    ^
2 errors detected in the compilation of "e4a.c".
```

Another example is the C99 predefined identifier feature, used to obtain the name of a function at compile time:

```
#include <stdio.h>

void f()
{
    printf("%s\n", __func__);
}

int main()
{
    f();
}
```

C and C++ have diverged over the years, but they've also converged in some areas. For example, the following code uses a declaration within a for loop, and is now both legal C (C99) and C++:

```
#include <stdio.h>

int main()
{
```

```

    for (int i = 1; i <= 10; i++)
        printf("%d\n", i);
}

```

Likewise, this code uses `//`-style comments, an idea C99 borrowed from C++:

```

// This is an example of C++-style comments.

int main()
{
}

```

Another area of incompatibility between C and C++ concerns features that are part of both languages, but which have different semantics. For example, both C and C++ support wide character types, but in C, `wchar_t` is a typedef defined in a header file, whereas in C++ it is a keyword. Based on this difference, the following code is valid C99, but not C++:

```

int wchar_t = 37;

int printf(const char*, ...);

int main()
{
    printf("%d\n", wchar_t);
}

```

No header file is included in this program, so it's perfectly okay to use `wchar_t` as an identifier, assuming this is a C99 program. If it's a C++ program, then `wchar_t` is a keyword, and the program is invalid.

An additional example of different semantics concerns file statics:

```

#include <stdio.h>

static int x = 37;

int main()
{
    printf("%d\n", x);
}

```

This code is legal C and C++, but the C++ usage is deprecated, that is, there is a possibility that the code will not be valid at some future time. The preferred C++ usage is unnamed namespaces:

```

#include <cstdio>

using namespace std;

namespace {
    int x = 37;
}

int main()

```

```

{
    printf("%d\n", x);
}

```

Whether this approach is really better than file statics obviously depends on your particular biases.

Conclusions

Suppose that you are concerned about compatibility in a practical way. You might have a large body of C89 code that you are thinking of migrating to C99. Or you might have some C code that you want to compile as C++. What should you do?

It seems likely that current C compilers will be upgraded to incorporate C99 features, and C99 is mostly compatible with existing C89 code. C99 provides some attractive new features that you might want to use. But if you care about compatibility with C++, it's not at all clear if and when C++ will incorporate C99 features. And it seems very unlikely that C will ever adopt many of the distinctive C++ features such as templates.

If you have a body of C code that you compile with a C++ compiler, some of the C99 features will help with compatibility: for example, support for C++-style comments and for mixing declarations and code.

Beyond these basic observations, there is really no alternative to sitting down and identifying the underlying differences between C and C++ and specifying some coding standards for use in your project. For example, if you want to use wide characters in your C application and compile the result with a C++ compiler, then you need to know that C treats `wchar_t` as a typedef'd type defined in a header, whereas C++ treats it as a keyword.

One Web page that discusses C/C++ differences can be found at <http://david.tribble.com/text/cdiffs.htm>.

practical perl

by Adam Turoff

Adam is a consultant who specializes in using Perl to manage big data. He is a long-time Perl Monger, a technical editor for *The Perl Review*, and a frequent presenter at Perl conferences.



ziggy@panix.com

Lightweight Databases

Many programs need to store some kind of state information between sessions. What's the best way to maintain this data? It depends on the application, of course. This month, I investigate solutions that are easy to use and that fit somewhere in between text files and relational database servers.

Managing Persistent Data

A few months ago, I started writing a program to monitor the online catalogs of several technical publishers periodically. Every time my program would visit a Web site, it would look for recently added book titles. Some of these publishers maintain extensive online catalogs, and as a good Web citizen, I certainly don't want to overload their servers with requests for information my program has already seen. Furthermore, I wanted to highlight new titles to see what I might be interested in reading.

Obviously, my program would need to store some state on disk describing what links had been seen before. However, there are literally dozens of ways to accomplish this, and it's not entirely obvious which one is best.

The lazy programmer's solution would be to write out some flat text files on exit that would be read in the next time my program runs. In this case, flat files would be adequate, if the saved data were relatively simple, such as a list of links stored one per line. However, if I need to save more data (ISBN, title, author, etc.), then other issues arise. For example, I would need to synchronize the functions to read and write to my datafile, to make sure they use the exact same format for both input and output. This might become a little tricky if I need to upgrade my program to store even more information later on.

Another perfectly valid approach is to start out using a relational database engine, such as MySQL, PostgreSQL, or Oracle. This is generally a good choice, except in this situation, using a

relational database server seemed like an overengineered solution. I didn't want to get bogged down with details of setting up databases, users, or passwords – I just wanted to write a simple Web crawler and save some data once my program finished. One issue I particularly wanted to avoid was having a program that magically breaks whenever a database server is moved, or when a database user or password changes.

In the end, I found that this simple little program fit into a sweet spot – somewhere between quick-and-dirty flat text files and full relational database servers. Many little programs I've written (most of them just quick hacks) fall into this category. One benefit of using Perl is that I'm not stuck using an inappropriate technology for my problem, whether that technology is overengineered or underengineered.

Of course, in Perl there is more than one way to solve this problem. In this article I want to examine two main types of solutions: the venerable DBM file, and lightweight relational databases.

Persistence Through DBM Files

The classic solution to this data storage problem is the venerable DBM file. DBM files come in many different forms, and all of them can be used to store simple key/value pairs. In this way, DBM files behave much like Perl hash variables, except that the keys and values can be saved and restored for later use.

Using a DBM file is quite simple and requires only a few lines of code to start:

```
#!/usr/bin/perl -w

use AnyDBM_File;
use Fcntl;

my %urls;

tie %urls, "AnyDBM_File", "url_data", O_RDWR | O_CREAT, 0640;

## ... %urls is transparently connected to the file "url_data.db" ...
```

First, we need to load a DBM library. In this case, I loaded the `AnyDBM_File` library through the `use` statement on line 3. I then create a new hash variable, `%urls`, on line 6, and connect that hash to the DBM file `url_data.db` with the `tie` statement on line 8. (The default DBM file implementation on my machine adds the `.db` suffix automatically.) From this point forward, any keys or values that are added or modified to the `%urls` hash in my program will also be stored on disk in the file `url_data.db`. The connection will be broken either when my program finishes or when I execute the statement `untie %urls;`

All of the magic occurs in the `tie` statement. This tells Perl to associate the variable `%urls` with the package `AnyDBM_File`.

The other parameters are sent to `AnyDBM_File` to describe the file we wish to use. Here, the parameters are a portion of the filename we'll be using (`url_data`), the flags used to open the file (`O_RDWR | O_CREAT`, values that come from the `Fcntl` module), and the permissions mode (`0640`, or read/write for the owner, read only for the group).

For my program to monitor online publisher catalogs, I could add a new key to this hash for each URL I process, with the value being the day it was processed. By checking to see if an entry already exists for a particular URL, I can easily identify which URLs are new:

```
foreach (@links) {
    next if defined $urls{$_}; ## We've seen this URL before

    print "New URL: $_\n";
    $urls{$_} = localtime();
}
```

And that's it. The first time I run my little link checker, I'll see a whole slew of URLs fly by. Starting with the second time I run my program, I'll only see the URLs that have been added since the previous run.

Flavors of DBM Files

Using the `AnyDBM_File` module is guaranteed to work whenever a new DBM file is created, but it does have some problems. Depending on the configuration of your system, Perl will support some of the various implementations of DBM files, including `NDBM`, `Berkeley DB`, `GDBM`, and `SDBM`. By using `AnyDBM`, you tell Perl that you don't care which one to use, any one of them is fine. The main problem with `AnyDBM` is that it is not guaranteed to use the same implementation on two different machines, nor is it guaranteed to open any random DBM file you happen to have. It is quite possible that `AnyDBM_File` will load the `NDBM_File` module when you want to open a file created by `DB_File` or `GDBM_File`. This operation will fail because the naming conventions or the file formats differ.

Therefore, it is better to choose a specific type of DBM file module instead of the `AnyDBM_File`:

- `NDBM_File` uses the native `NDBM` library on your system, if one exists.
- `DB_File` uses the `Berkeley DB 1.85` library, if present.
- `GDBM_File` uses the `GNU GDBM` library, if present.
- `SDBM_File` is Perl's own DBM library and is always available.

Each of these DBM libraries has its own advantages and disadvantages; see the documentation for `AnyDBM_File` for more information (`man AnyDBM_File` or `perldoc AnyDBM_File`). I use either `DB_File` or `GDBM_File`, because they tend to be avail-

able on most Perl installations. `SDBM_File` will always work, and it exists as a DBM implementation of last resort.

Remember that DBM files store simple key/value pairs. If you are programming multi-level data structures, such as a hash of hashes or a hash of lists, then regular DBM files will not store all of your data properly. For these kinds of data structures, look into Joshua Chamas' "multi-level DBM" module, `MLDBM`, available on the `Comprehensive Perl Archive Network (CPAN)`.

Limits to DBM Files

Perl's support for DBM files makes it easy to add persistent data structures to a program with just a few lines of code. The main disadvantage is that you need to manage all of the data yourself, using Perl hashes. This may be a useful technique in the small, but tends not to scale very well as requirements grow.

Suppose I wanted to create a report program to count URLs, grouped by the day they were first encountered. Using DBM files, that code might look something like this program:

```
#!/usr/bin/perl -w

use strict;

use DB_File;
use Fcntl;

## Load in the cache of URL => date values
my %url_dates;
tie %url_dates, "DB_File", "url_dates", O_RDWR | O_CREAT, 0640;

## Count books (hash entries), grouped by the day they were found
my %count_by_day;
foreach (values %url_dates) {
    ## Strip out the time component of the date
    ## "Sun Aug 11 13:18:59 2002" -> "Sun Aug 11 2002"
    my $date = $_;
    $date =~ s/\d{2}:\d{2}:\d{2} //;

    $count_by_day{$date}++;
}

## Print out the results (unsorted)
my ($date, $count);
while (($date, $count) = each %count_by_day) {
    print "$date:$count\n";
}
```

This small program re-uses the existing DBM file created by my Web-crawling program that finds new links. Note that this "little" program is 28 lines long (with whitespace and comments). More interesting reports, like one that counts books that contain the word "Perl" in the title, would require more data and might actually be significantly more involved. Now, imagine that two, three, or more of these reports become useful. All of a sudden, the quick-and-dirty solution is starting to run out of

steam, since each new report might require a few dozen lines of new code.

It's clear that DBM files, while useful in some circumstances, aren't always the best or the simplest solution available.

Lightweight Relational Databases

As the requirements for my quick little book-catalog program slowly grow, it's clear that a SQL database is the most appropriate solution, especially if I intend to perform multiple queries on this data. Remember that the problems I intentionally want to avoid are some of the administrative details of setting up databases and passwords with a database engine like MySQL or PostgreSQL. That is, I want my program to "just work," and not be impacted if I happen to move my MySQL server to another computer, convert to PostgreSQL, or change a username or password. Additionally, I want my program to "just work" if I move it to another computer, without requiring that a particular database engine be installed to run this little hack.

Again, we're using Perl, so there's more than one way to do it.

Two ready-to-use modules are available on CPAN that meet my requirements. The first is Jeff Zucker's `DBD::CSV` module, and the second is Matt Sergeant's `DBD::SQLite` module. Both of these are database drivers that work with Perl's DBI module, Perl's generic interface to many different database engines. `DBD::CSV` simulates a relational database by using text files with comma-separated values for each table in the database. `DBD::SQLite` contains a full-fledged relational database engine written in C that's embedded in the database driver module itself. Neither of these modules require setup, configuration, or a server process to manage the database. They just work.

If you're already familiar with using DBI to connect to MySQL or other relational databases, there is nothing new to learn here. Furthermore, should you need to upgrade from a CSV or a SQLite database, all you need to do is change the DBI connection string, and possibly some of your SQL statements – the rest of your Perl programs remain unchanged.

I used to recommend and use `DBD::CSV` when I wanted to create a lightweight relational database. Once Matt released his `DBD::SQLite` module, I started using that instead, since it contains a more robust database engine. This is mostly due to the hard work of Richard Hipp, who created SQLite as a full-featured, embeddable relational database, complete with indexes, transactions, and multiuser access.

Creating a Perl program that uses SQLite is straightforward, assuming you're already familiar with DBI and SQL (another issue entirely):

```
#!/usr/bin/perl -w

use strict;
use DBI;

my $dbname = "url_dates.db";
my $dbh = DBI->connect("dbi:SQLite:dbname=$dbname");

## ... use this SQLite database just like any other DBI database ...
```

One interesting feature of SQLite is that its columns are generally typeless. The column types that are declared in a `CREATE TABLE` statement are ignored (with the exception of integer primary keys), so there is no need to worry about losing data when storing a 30-character string in a column declared to be of type `CHAR(25)`, or getting an error when storing a string value in an `INTEGER` column.

Using a relational database makes reporting much easier. For example, a program to count all URLs in the database, grouped by date, would be much simpler than the DBM version seen above:

```
#!/usr/bin/perl -w

use strict;
use DBI;

my $dbh = DBI->connect("dbi:SQLite:dbname=url_dates.db");

my $stmt = $dbh->prepare("SELECT day, COUNT(day)
                        FROM urls GROUP BY day");

$stmt->execute();
while (my @row = $stmt->fetchrow_array()) {
    print join(" ", @row), "\n";
}
```

This program is half the size of my previous report program, and all of the logic for this report is contained in the SQL statement on line 4. The `while` loop at the bottom is reasonably generic and can be abstracted out into a separate sub. It would also be relatively easy to add another SQL query to count the number of books that contain "Perl" in the title – something that would have required more than one extra line of code in the DBM version of the program.

Conclusion

Maintaining persistent data is a common task in Perl programs, and there are easily dozens of ways to do it. For the truly simple tasks, Perl makes simple DBM files available easily and transparently. For more complicated tasks, the easiest solution tends to involve using the DBI, along with a suitable database engine, whether that's something big and powerful, or something small and easy to set up.

generating ethernet packets

by Clif Flynt

Clif Flynt is president of Noumena Corp., which offers training and consulting services for Tcl/Tk and Internet applications. He is the author of *Tcl/Tk for Real Programmers* and the *TclTutor* instruction package. He has been programming computers since 1970 and a Tcl advocate since 1994.



clif@cflynt.com

This article is the first of a series on building network and firewall testing and validation tools using Tcl, open source packages, and some special-purpose hardware. This time I will describe building and testing a Tcl extension for generating Ethernet packets. Subsequent articles will expand on techniques for using this and other extensions.

When I'm building a firewall system I always worry about what I might have missed. Did I install the new security patches in the right places, define the rules correctly, leave no holes?

There are online services like <http://scan.sygatetech.com/> that will scan my system for common flaws, but that requires that I put the system on the Net to test it.

SATAN or SAINT, for example, will check the system for a lot of holes, but they don't test all the firewall rules.

So, I decided to write my own firewall test framework and add new tests as I find and need them.

The first thing I needed for this toolkit was some way to send arbitrary IP packets, to confirm that things like packets on the outside interface with inside addresses are blocked, malformed packets are discarded, and so on.

A little Net searching found a few tools that would almost do what I want. These tools include the Tcl extensions `psh` from Sun and `pkt` from USC, and the programs `mgen` from the Naval Research Laboratory and `sendip` from Project Purple.

After a little bit of looking, I decided to work with the `libdnet` library, written by Dug Song (<http://libdnet.sourceforge.net/>). The advantage of this package is that it has the low-level support I need, is currently supported, and has adequate documentation.

The disadvantage is that it's a C library, not a Tcl extension, but that's easily changed.

Tcl was designed to be easily extended. With just a few hours' labor you can pick up a random library and generate the interface code to use it as a Tcl extension. However, this does require some knowledge of how Tcl extensions are constructed. Doing this the first time can take closer to eight hours.

If you don't feel like spending that much time and learning Tcl internals, you can use the SWIG (SoftWare Interface Generator) program to create the interface code for you (<http://www.swig.org/>).

SWIG was developed by David Beazley (beazley@cs.uchicago.edu) to make his life easier while he was developing software at Los Alamos. Even in its early forms the program was very useful.

I downloaded the version 1.3.13 for this work. SWIG's built-in support for structures and complex data types is constantly improving. Some details described in this article may be different on the version of SWIG you are using.

SWIG works by examining a definition file that describes the functions and data structures in a library and generating some C code to allow those functions to be loaded into a Perl, Python, Tcl/Tk, Ruby, Guile, or MzScheme interpreter.

Generating a definition file is fairly simple. The basic format is just a list of function declarations.

For example, if you have a file named `fibon.c` that contains this Fibonacci function:

```
int fib (int i) {
    if (i <= 1) {return 1;}
    return fib(i-2) + fib(i-1);
}
```

it could be turned into a Tcl extension with this one-line definition file:

```
$> cat fibon.i
int fib(int i);
```

and this SWIG command line:

```
swig -tcl -module fib -prefix fib -namespace -v fibon.i
```

The `-module fib` argument defines the name for this module. The module name can be defined on the command line (as done here) or in the definition file, with the line `%module fib`.

The `-prefix fib` argument sets a value that will be used to prevent command name collisions. When used with the `-namespace` argument, SWIG will generate code to create the new commands in the `fib` namespace. Placing the extension commands

in a namespace has become the preferred style for Tcl extensions.

Running this SWIG command will create a wrapper file named `fibon_wrap.c`, which can be compiled into a shared library with a command line resembling this:

```
gcc -shared -L/usr/lib fibon_wrap.c fibon.o -o libfib.so
```

Once this is done, you can load the `libfib.so` library and use the Fibonacci code in your Tcl scripts just as you would any other Tcl extension.

```
load ./libfib.so
for {set i 1} {$i < 5} {incr i} {
    puts "The Fibonacci series at level $i is [fib::fib $i]"
}
```

Unfortunately, most projects are a bit more complex than this.

One problem you run into is that C is a lower level language than Tcl. The C compiler supports data structures that reflect the organization of the data in physical memory, while the Tcl interpreter insulates the programmer from the hardware.

The SWIG solution for this is to generate new Tcl commands for creating and accessing C data structures. The new Tcl commands to create a C data structure will allocate memory for the data structure and return an identifier that Tcl scripts can use to reference the structure. The Tcl script can then pass that identifier to the interface for C functions that need to access the data. As an added benefit, SWIG uses some magic naming conventions to do runtime data checking, so you can't accidentally pass a structure of type `a` to a function expecting a structure of type `b`.

You can create an extension with support for creating and using C arrays by adding a little bit of code to the definition file.

The easiest way to do this is to use SWIG's `%inline` directive. This directive defines functions which should be both included in the final wrapper and exposed to the SWIG parser and code generator.

The SWIG documentation includes this example to show how an array of doubles can be created and accessed:

```
// SWIG helper functions for double arrays
%inline %{
// Create a new array of doubles of a given length
double *new_double(int size) {
    return (double *) malloc(size*sizeof(double));
}
// Delete an array of double
void delete_double(double *a) {
    free (a);
}
```

```
// Retrieve the value of an element of the array
double get_double(double *a, int index) {
    return a[index];
}
// Set the value of an element in the array
void set_double(double *a, int index, double val) {
    a[index] = val;
}
%}
```

The new commands can be used like this.

```
# Create an array of doubles
set squares [new_double 5]
# Fill the array with the square of the index
for {set i 0} {$i < 5} {incr i} {
    set_double $squares $i [expr $i * $i]
}
# Invoke a library procedure that requires a
# pointer to an array of doubles as an argument
foo $squares
```

Structures can be a bit more tricky to use but are very simple to describe in the definition file. All you need to do is include the C struct in the body or in an `%inline` section of the definition file and SWIG will generate a set of interface functions and include them in the Tcl extension.

For example, a structure like this:

```
struct arp {
    unsigned char mac_address[6];
    unsigned char ip_address[4];
}
```

can be accessed with a Tcl script by making a definition file that looks like this:

```
%inline %{
// Define an ARP structure for Machine and IP address
struct arp {
    unsigned char mac_address[6];
    unsigned char ip_address[4];
}
// A helper utility to set values in an array of
// unsigned chars -
// copied from the array example
int unsigned_char_set (unsigned char *ar, \
    int index, unsigned char val) {
    ar[index] = val;
}
// A test function to display the contents of an
// arp structure
int showArp (struct arp *p) {
    int i;
    for (i=0; i<6; i++) {
```

```

        printf("0x%x ", p->mac_address[i]);
    }
    printf("\n");
    for (i=0; i<4; i++) {
        printf("0x%x ", p->ip_address[i]);
    }
    printf("\n");
}
%}

```

When SWIG processes this code, it creates these new Tcl commands:

<code>::fib::new_arp</code>	Allocates memory for a new arp structure and returns the name to the Tcl script that invokes it.
<code>::fib::arp</code>	
<code>::fib::delete_arp</code>	Frees the memory associated with an arp structure
<code>::fib::arp_ip_address_get</code>	Returns a handle to access the ip_address C array element of the arp structure.
<code>::fib::arp_mac_address_get</code>	Returns a handle to access the mac_address C array element of the arp structure.
<code>::fib::showArp</code>	An interface into the showArp C function.
<code>::fib::unsigned_char_set</code>	An interface into the unsigned_char_set C function to assign values to elements in a C array.

The Tcl code to test this resembles the following:

```

set arp [arp::new_arp]
set mac [arp::arp_mac_address_get $arp]
for {set i 0} {$i < 6} {incr i} {
    arp::unsigned_char_set $mac $i $i
}
set ip [arp::arp_ip_address_get $arp]
for {set i 0} {$i < 4} {incr i} {
    arp::unsigned_char_set $ip $i $i
}

arp::showArp $arp

```

The body of a definition file can usually be extracted from an include file. If you are lucky, you can just use the package's primary include file as a definition file.

The `dnet.h` file has too much information that's not relevant to creating a wrapper (and is confusing to the SWIG parser), so the simple solution of using `dnet.h` as a definition file didn't work.

However, all the critical pieces of information (the functions, declarations, and structures used as arguments) are described in the man page, so a set of cut-and-paste operations will create a minimal definition file.

To ensure portability across different word-size machines, the `libdnet` package uses several data types that aren't part of the basic C language. The SWIG parser doesn't recognize these new datatypes. The SWIG solution for unrecognized data types is to consider them to be pointers.

However, the SWIG parser will recognize a `#define` or `typedef` directive to define these datatypes. Adding these lines to the definition file satisfies the SWIG parser:

```

typedef unsigned short uint16_t;
typedef unsigned char  uint8_t;
typedef unsigned int   uint32_t;
typedef unsigned int   ip_addr_t;
typedef unsigned int   size_t;

```

To finish the `dnet.i` definition file, I added versions of the C array access code described above to handle arrays of `uint32_t`, `uint16_t`, and `uint8_t` data.

Once the definition file is complete, SWIG can create a Tcl extension in seconds. The next step is to test the new extension and see if it works.

One of the features of the `libdnet` library is the ability to send raw packets over the Ethernet. This is as low-level as you can get, and will let me generate whatever type of malformed IP packet I need.

The two critical commands are `eth_open`, to open a connection to an Ethernet device, and `eth_send`, to transmit a buffer of binary data (an Ethernet frame).

Syntax: `eth_t *eth_open(const char *device);`

Open a connection to an Ethernet device and return a handle for future use.

`char *device` The name of the Ethernet device to be connected to, such as `eth0`, `pn0`, etc.

Syntax: `ssize_t eth_send(eth_t *e, const void *buf, size_t len);`

Transmit a buffer of data over the Ethernet. The buffer should be a valid Ethernet frame. Returns the number of bytes sent. The checksum will be appended automatically.

`eth_t *e` The handle returned by `eth_open`

`void *buf` The data to send over the link

`size_t len` The number of 8-bit characters to transmit

One problem is that `eth_send` requires that the `buf` buffer be a pointer to an area of memory. A Tcl string won't be accepted by the SWIG wrapper. Fortunately, the SWIG wrapper's data validity checking will accept any pointer as a void pointer, so we can use the `uint_8_t` array commands to create and fill an array of unsigned chars.

Simple code like this will generate garbage packets on the local Ethernet. The data is illegal Ethernet frames, which aren't accepted by other nodes on the network, but running the script will cause the activity lights on an interface card to blink, demonstrating that frames are being sent.

```
# Load the new extension
load ./libdnet.so
# Open a connection to the Ethernet device
set e [dnet::eth_open eth1]
# Create a buffer
set buf [dnet::new_uint_8Array 60]
# Stuff the buffer with incrementing values
for {set i 0} {$i < 60} {incr i} {
    dnet::set_uint_8Array $buf $i $i
}
# And shove it onto the wire 10 times
for {set i 0} {$i < 10} {incr i} {
    dnet::eth_send $e $buf 60

    # Pause for 100 milliseconds
    after 100
}
}
```

The next step is to send a legal packet and see if it's recognized.

An Ethernet frame consists of five fields of data:

Field size (bytes)	Description
6	The destination MAC address.
6	The source MAC address.
2	A type definition. This is 0x0800 for IP datagrams.
46–1500	The datagram.
4	A Cyclic Redundancy Checksum.

The `arp -a` command gave me a list of IP addresses and corresponding MAC addresses to fill in the source MAC address and destination MAC address fields; the type field for an IP datagram is 0x0800, and the CRC will be appended by the transmission code.

To generate a valid IP datagram, I used `tcpdump` with the `-x` option to get a hex dump of an IP packet. I decided to ping the

target node from the node running the Tcl script and grab one of those packets. Using an Echo Request packet provides two sets of validation. Using `tcpdump`, I can watch the packet arrive on the target node, and I can also see if the target machine responds to the fabricated ping request.

Tcl has full support for operating with lists of data. It makes sense to treat a packet as a Tcl list of hex values until it needs to be converted to an array of unsigned chars for the `eth_send` command.

The code below creates an Ethernet frame from the various pieces of data. It uses the `split` command, to convert a colon-delimited MAC address into a list of hex bytes, and the `eval` command, to combine two lists into a single list.

The Tcl `split` command will split string data into a list.

Syntax: `split string ?splitChars?`

`split` Splits a string into a list. Elements are delimited by a marker character.

`string` The string to split.

`?splitChars?` A string of characters to mark elements. By default the markers are whitespace characters (tab, newline, space, carriage return). In this example, the character to split on is the colon separating the bytes in a MAC address.

The `eval` command concatenates the arguments into a string before starting the evaluation. This causes a set of data to lose one level of data grouping. Without `eval`, a command like `lappend list $list2` would be evaluated as `lappend list {a b c}`, which will append the list element `{a b c}` to a list. The command `eval lappend list $list2` would be evaluated as `lappend list a b c`, which will append three list elements, `a`, `b`, and `c` to a list.

This script will generate an Ethernet frame and transmit it to the local network:

```
# The MAC address, obtained with arp -s
set destEther 00:E0:4C:00:14:4D
set srcEther 00:A0:CC:D1:B6:00
# A valid echo request packet,
# obtained with tcpdump
set echo_Request [list 45 00 00 54 00 00 40 00 40 01 \
    05 16 c0 a8 5a 40 c0 a8 5a 02 08 00 98 d9 \
    df 22 00 00 63 cf 4d 3d d5 f3 0e 00 08 09 \
    0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17 \
    18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 \
    26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 \
    34 35 36 37]
# Fill a list with hex values
set packet [split $destEther :]
eval lappend packet [split $srcEther :]
```

```

lappend packet 08 00
eval lappend packet $echo_Request
# How many bytes are we using?
set len [llength $packet]
# Create a C array and fill it.
set buf [dnet::new_uint_8Array $len]
for {set i 0} {$i < $len} {incr i} {
    dnet::set_uint_8Array $buf $i 0x[lindex $packet $i]
}
# And shove it onto the wire 10 times
for {set i 0} {$i < 10} {incr i} {
    dnet::eth_send $e $buf $len

    # Wait 100 milliseconds between frames
    after 100
}

```

This extension provides a platform for generating IP packets. The next article will start describing techniques for validating the packet generator before using the generator to validate another system.

USENIX and SAGE Need You

People often ask how they can contribute to our organizations. Here is a list of tasks for which we hope to find volunteers (some contributions not only reap the rewards of fame and the good feeling of having helped the community, but authors also receive a small honorarium). Each issue we hope to have a list of openings and opportunities.

The SAGEwire and SAGEweb staff are seeking:

- Interview candidates
- Short article contributors (see <http://sagewire.sage.org>)
- White paper contributors for topics like these:

Back-ups	Emerging technology	Privacy
Career development	User education/training	Product round-ups
Certification	Ethics	SAGEwire
Consulting	Great new products	Scaling
Culture	Group tools	Scripting
Databases	Networking	Security implementation
Displays	New challenges	Standards
E-mail	Performance analysis	Storage
Education	Politics and the sysadim	Tools, system
- Local user groups: If you have a local user group affiliated with USENIX or SAGE, please mail the particulars to kolstad@sage.org so they can be posted on the Web site.

;login: is seeking attendees of non-USENIX conferences who can write lucid conference summaries. Contact Tina Darmohray, tmd@usenix.org, for eligibility and remuneration info. Conferences of interest include (but are not limited to): Interop, SOSP, O'Reilly Open Source Conference, Blackhat (multiple venues), SANS, and IEEE networking conferences. Contact login@usenix.org.

;login: always needs conference summarizers for USENIX conferences too! Contact Alain Hénon, ah@usenix.org, if you'd like to help.

musings,

Or What I Did on My Summer Vacation

I once imagined that I would like to spend my life attending conferences. Instead, I am feeling glad to be home, although I am also glad I did get to hang out in a couple of security conferences. And, rather than making you have to drive, ride, or fly, I will share with you parts of my experiences, and something that I think you may find very frightening.

I loathe Las Vegas. Gambling does not appeal to me, so having to walk through three casinos to reach the registration desk at Caesars Palace had me seething inside. I remembered (just in time!) that I am enlightened and cheered up enough to survive the 20-minute check-in line, then another 20-minute wait for the elevator (you'd think this was Eastern Europe, not an expensive hotel), all to attend Black Hat 2002.

The Black Hat conference is designed for security consultants, although I did see DoD types and even some faces from USENIX conferences there. The format consists of three tracks, with intermediate to moderately advanced talks about the security of software and hardware. At the low end, some guys from iDefense gave a lecture about cookies (I liked Kevin Fu's invited talk at last year's Security Symposium better). I enjoyed the explanation of Hogwash and how it had been integrated into Snort codebase as of version 9.2. And how the HoneyNet Project plans to proceed with their version 2 honeynets.

Daemon9, now better known as Mark Schiffman of @stake, described his new library, libradiate, which adds to libnet (low-level networking functions for crafting/reading packet headers) with the headers necessary for 802.11B (WiFi). Schiffman demonstrated Omerta, a program that sniffs a wireless channel and disassociates any network card currently associated with an access point. He did not share the source code, a disappointment to many. He did provide other C code examples, but a show of hands revealed that there were only three C coders in the audience. Rather disappointing for a technical con.

While Schiffman rushed through his code examples, FX and Kimo, of Phenoelit (<http://www.phenoelit.de>), were explaining how to turn HP printers into port scanners, using Java code and a class loader included in networked HP printers. The audience found this very amusing (printers scanning a network!), but someone later pointed out to me that HP network printers already will scan networks looking for print servers. What I had missed was their discussion of heap buffer overflows of low-end Cisco routers. Their exploit invalidates the stored configuration and forces a reboot, at which point the router, realizing its configuration is hosed, begins broadcasting a request for a new configuration from anyone. IOS 12 and Cisco 1000, 1600, and 2600 routers are vulnerable to remote attacks, and the 2500 series to local attacks only.

Remember that I mentioned rumors about exploits to IOS in an earlier column. FX made certain that I (and Cisco) understood that they had not done any reverse engineering, just opened the router, recognized that it used a Motorola 68K processor, and used debug messages and information on the Cisco Web site to create their exploit. FX told me that he did not want to be this year's Sklyarov (the Russian arrested at DefCon 9 for explaining how to defeat Adobe's pitiful encryption in eBooks). I really wish that Cisco had succeeded in rewriting IOS as a modern embedded OS instead of abandoning the effort (as far as I have been able to find out).

by Rik Farrow

Rik Farrow provides UNIX and Internet security consulting and training. He is the author of *UNIX System Security and System Administrator's Guide to System V*.



rik@spirit.com

Hacking has nothing to do with breaking into other people's computers, and everything to do with understanding how things work.

I ran into FX at the next conference I visited, DefCon. DefCon was a bit more subdued this year when compared to previous years, I've been told. I had just enough time to hang out the night before and the first half-day, long enough to get a feel for things and not really missing out on what I hadn't liked about the first DefCon – drunken, chain-smoking teenagers. Keep in mind that DefCon is a serious security conference, with a very low entry fee (\$75). Many speakers from Black Hat also speak at DefCon but give a more technical version of their talks.

I listened to Ofir Arkin present the revised version of Xprobe, which does away with the tree structure for probes and instead focuses on a list of probe modules. The modular structure makes Xprobe easier to extend, but it also loses one of the benefits of the original tool, which was accurate TCP/IP fingerprinting with two or three packets. Being an “older guy” who has lost enough hearing (probably from loud concerts in the sixties) was a real disadvantage at DefCon, as two of the conference rooms were outdoor tents, and the roaring of the AC units attempting to keep the temperature at reasonable levels drowned out some of what the speakers said, as well as most questions.

I am sorry I missed Jennifer Granick's talk on the implications of the PATRIOT Act (and yes, it is an acronym) for security practitioners, Simple Nomad's (<http://www.nmrc.org>) talk about the Hacker Nation and how the “War on Terrorism” affects hacking in general, the two lock-picking sessions, and Richard Thieme's (<http://www.thiemeworks.com>) closing session, reminding the audience that hacking is a form of truth seeking.

Lest this last statement confuse you, remember that hacking has nothing to do with breaking into other people's computers and everything to do with understanding how things work – even if it means taking them apart first. My next trip (and why I left DefCon early) took me to San Francisco for the USENIX Security Symposium. The December issue of *login*: will include the summaries from this conference, as well as other articles dedicated to security, and will be a great edition. I know, as I am the editor and already have some of the articles in hand.

But I don't want to make you wait quite that long. This year's symposium was great, lots of good papers and ITs, and the hall talk was great as well. Professor Felton spoke about the “Freedom to Tinker,” another way of saying that reverse engineering of code is akin to US First Amendment rights. Tinkering with things is not only common (would you buy a car where the hood was sealed?) but is good for the community and economically beneficial. Tinkerers have discovered security mistakes in code, and their activities often result in better, competing products (see <http://www.freedom-to-tinker.com/>).

One hall debate led right into a special evening talk about Palladium and TCPA (Trusted Computing Platform Alliance). Tom Perrine, of San Diego Supercomputing Center, asked, rhetorically, why software was so insecure. His argument: that better and more formal design processes would make a huge difference, even when using the insecure programming languages common today (C and Perl as examples). I piped up with my common assertion that you cannot compel people to use formal design processes, but you might instead provide them with safer tools to use – that is, instead of C, using programming languages that enforce good practices, and make it close to impossible to create buffer and heap overflows or to write code that does not check user input, etc. And that this must be implemented on top of a secure operating system that can run untrusted applications in their own compartments. No one agreed

with me, although Perrine did muse about the virtues of ADA, a programming language developed by the DoD for portability and security, and KSOS, an operating system with a trusted kernel.

The EFF's Lucky Green moderated a panel discussion with Peter Biddle of Microsoft and Seth Schoen of the EFF about Palladium. Palladium is Microsoft's project for developing software and hardware for a trusted kernel (see <http://vitanuova.loyalty.org/2002-07-05.html>, under the Microsoft heading, for details). You might think that I would be happy that *someone* is thinking about hardware support in PCs for running a trusted kernel, but the Microsoft focus is of course not the same as an open source focus for security. You should read Ross Anderson's TCPA FAQ for a very detailed critique (<http://www.cl.cam.ac.uk/%7Erja14/tcpa-faq.html>). But I will give you the nutshell here.

Biddle explained how Microsoft's Palladium will work to provide a trusted operating system. Briefly, after booting the trusted kernel, special hardware calculates a hash of this trusted kernel. Then the regular operating system continues with the boot process. The trusted kernel – officially the Trusted Operating Root (TOR), unofficially the “nub” – provides a limited set of services to the operating system and other applications, particularly the ability to seal and unseal “blobs” – any set of data, be it a program, a text file, or a DVD image. The TOR relies on a bit of hardware, a secure cryptographic coprocessor (SCC or SCP) that can perform asymmetric encryption (used in digital signatures) and symmetric encryption (AES in CBC mode), and support a secure store for keys. The SCP also controls memory management, protecting certain regions of memory so that even a root user or the operating system itself cannot access protected memory. At this point, it sounds like there exists the basis for the trusted kernel and compartments that I have long advocated.

But the plan for these wonderful security features is quite different. Instead of protecting the security of your system from attacks, Palladium protects rights of copyright owners. To quote Anderson:

TCPA and Palladium do not so much provide security for the user as for the PC vendor, the software supplier, and the content industry. They do not add value for the user, but destroy it. They constrain what you can do with your PC in order to enable application and service vendors to extract more money from you. This is the classic definition of an exploitative cartel – an industry agreement that changes the terms of trade so as to diminish consumer surplus.

To provide a few examples of how Palladium and TCPA work to enforce Digital Rights Management (DRM), imagine a system where you cannot migrate files from, say, Microsoft Office 2003 to any other software package. The TOR will not allow you to decrypt the file for the purpose of exporting it to, say, StarOffice. Organizations can configure applications so that data can never be shared, or so that files automatically and irrevocably delete themselves after some time period. No whistle-blowers “leaking” information, no email records detailing dirty deeds, and no more unlicensed copies of Microsoft software, as you *must* have a valid license to run – one that is keyed to your hardware platform using the SCP and the TOR. For the people controlling digital rights, this will be a windfall, as they control how many times you can play a DVD, prevent you from copying it (even by screen scraping), or can even charge you each time you open an application.

Instead of protecting the security of your system from attacks, Palladium protects rights of copyright owners.

We must choose between freedom (with its responsibilities) or passing over control of our computers and aspects of our lives to large corporations.

Microsoft and Intel claim that these new initiatives, once completed, will make your PC more secure, even prevent spam. But, instead of making your own computer more trustworthy for your own use, it will make it trustworthy for the use of content and application providers. Viruses will not be able to affect the TOR or Trusted Agents protected by the TOR, but they will still be able to write to files, delete files, send email – in fact, do almost everything they do today. The only exception will be those files and devices protected by the TOR (which in Palladium includes the keyboard, so no more keystroke sniffers).

Too bad these are DRM initiatives, not real security initiatives. Lucky Green pointed out to me, as does Dar Williams in his July 5 journal entry, that Intel and Microsoft felt they had no choice but to create an unbreakable system for DRM. If they failed to do so, the home entertainment system of the future might not use Intel hardware and Microsoft software. But the very success of these schemes gives each company tremendous leverage, far beyond the virtual monopolies each enjoys today.

You will still be able to run your favorite operating system on Palladium and TCPA-enabled systems. In fact, there were people at the Symposium with IBM T30 laptops that incorporate a TCPA chip. You just won't be able to use any of the features that require the chip, as these make use of a TOR and trusted hardware: for example, an encrypted link to your monitor, DVD-ROM, keyboard, etc.

The RIAA and MPA argue that flagrant copyright violations are destroying their businesses and hurting artists. Perhaps the former might one day be true, but the latter rarely is. Record companies lend money to bands, and it is unusual for the artists that provide the content for RIAA members to make a living as musicians. But providing free digital downloads can help promote artists (read about Janis Ian's experiences as a recording artist and musician, and how free downloads have helped her and Mercedes Lackey, http://www.janisian.com/article-internet_debacle.html).

The TCPA chip has been coined the "Fritz" chip, after Senator Fritz Hollings, who has sponsored a law that would make the selling of any computer or storage device that does not support TCPA illegal in the US (http://www.salon.com/tech/feature/2002/03/29/hollings_bill/).

The sky is not falling. We find ourselves at a crossroads where we must choose between freedom (with its responsibilities) or passing over control of our computers and aspects of our lives to large corporations. I believe the decision is clear, but I know my mother, as well as many of my friends, doesn't understand the issues (yet). Make yourself heard, ask for real security, and don't give up your freedom.

ISPadmin

Stopping Spam, Part 2

Introduction

This installment of ISPadmin will examine techniques for stopping outbound spam (“unsolicited commercial email,” or UCE, originating on your network, destined for a machine or network which you do not control). In the last edition, how to stop spam from the inbound side (from someone else’s network to your mailbox) was covered in detail.

Background

Methods for stopping outbound mail are very different from those used to stop inbound spam. Most of the ways outbound spam is stopped can be classified as follows:

- Controlling access to a mail relay machine (e.g., smtp.isp.net)
- Limiting SMTP access to known blocks of open mail relays (e.g., Korea)
- Limiting the number of outbound SMTP connections a client can make over a period of time
- Capping the amount of k/sec an outbound SMTP connection can make

The methods covered in this article will fall into one of the categories listed above, although the coverage will be grouped differently to enable clearer coverage of the topics.

Generic Methods

First, let’s discuss generic methods that are not tied directly to a specific open source solution or network hardware (e.g., routers). These methods can be applied to any mail infrastructure, though Sendmail-specific information is listed within this section.

RESTRICTING IP

Controlling what IP addresses are allowed to send mail through a mail server is an important step everyone who runs a mail system on the Internet should take. This is a very common method to control access to a mail relay. In the provider’s mail relay machines, a list of IP addresses or blocks is kept that are allowed to relay mail through the relay(s). For Sendmail, the “IP allowed to relay” list is kept in an access database entry similar to the following:

```
209.206.10 RELAY
```

(Sendmail access databases were covered in last issue’s ISPadmin column.) Even if you are not a provider, if you are running Sendmail you should be restricting access to your mail relays in this manner. If you don’t, you run the very high risk of becoming a spam pariah!

POP BEFORE SMTP

The POP before SMTP method requires the end subscriber to simply check their mail before sending it. This method can be used for “roaming” subscribers, who won’t be coming from one of the provider’s own IP address ranges. Once the POP box is accessed successfully, the subscriber’s IP address goes into the IP address “allowed” list on the mail relay(s) for a certain period of time, most commonly 30 minutes. In the case of a Sendmail-based mail relay, the method to control mail relay access can be

by Robert Haskins

Robert Haskins is currently employed by WorldNET Internet Services, an ISP based in Norwood, MA. After many years of saying he wouldn’t work for a telephone company, he is now affiliated with one.



rhaskins@usenix.org

Kai's SpamShield is probably one of the oldest packages out there specifically designed to counter outbound spam.

performed via the access database entry, identical to the approach outlined in the "Restricting IP" section.

MAIL MESSAGE METERING

(Disclaimer: This author developed the Mail Message Metering anti-spam method, which has a patent pending. Describing the method here does not imply the ability to use the system described here.) The Mail Message Metering method is simple in concept and relatively simple to implement. The method is useful to wholesale Internet access providers, although any enterprise that generates lots of outbound mail could use it.

As each subscriber generates an outbound mail message, the network component (switch, RAS gear, DSL aggregating equipment, etc.) redirects the connection to a specially configured mail relay. This specialized mail relay queries a database which contains a current listing of all originating IP addresses that have relayed mail, and associated counts of the number of messages for several time periods (e.g., past minute, past 30 minutes, past hour). If the message would exceed predetermined thresholds, then the message would be re-queued. If the message didn't exceed the limits, then the message would be allowed through and the counts updated appropriately.

Other people and organizations hold anti-spam patents. Of these, Brightmail is probably the best known. However, this author (who is not an attorney) can find no patent (granted or pending) specific to outbound spam.

The benefits of this approach are many:

- Blocks high percentage of outbound spam
- No subscriber and little customer impact
- Configurable and scalable
- Limited impact on authentication (RADIUS) servers

The shortcomings are:

- Requires "white hat" list of legitimate bulk mailers
- Requires use of SMTP redirection (may require additional hardware)

The December 2000 issue of *login*: contained an in-depth article on the Mail Message Metering solution.

Open Source Packages

One open source package is specifically designed to counter outbound spam (Kai's SpamShield). The others described below can be used to control both inbound and outbound spam.

KAI'S SPAMSHIELD 1.0

Kai's SpamShield is probably one of the oldest packages out there specifically designed to counter outbound spam. It is a Perl script run out of cron which works by analyzing the most recent sections of the Sendmail log file (usually maillog). The program counts the IP addresses from which messages are originating. If these counts exceed previously entered thresholds, the sender's access to the mail relay is blocked. While dated (it doesn't appear to have been updated since 1997), it is very effective against outbound spam.

Kai's SpamShield version 2.0 was just announced as of this writing in July 2002. No details on the functionality included in the new version exist on the Web site, however.

BLACKMAIL

Blackmail performs various checks against the headers of incoming and outgoing mail messages. These checks include:

- Known sources of spam
- Specific words and/or phrases
- Resolvable names in headers
- Black hole lists
- To: and From: headers
- Correct header formation

While more recent than Kai's SpamShield, it appears that most of these checks are performed by SpamAssassin as well. One difference would be the fact that Blackmail is written in C, while SpamAssassin is written in Perl.

PROCMail

System-wide procmail filters can be built to assist in the fight against spam. Two such packages are The SpamBouncer and Email Sanitizer. These work by encapsulating the various anti-spam rule sets (e.g., black hole lookups, resolvable to/from domains, etc.) as procmail recipes. While this author has no direct experience with them, there are enough procmail-based tools out there to indicate this is a valid approach.

SMTP PROXY

SMTP proxies (such as Obtuse Systems Corporation's Juniper firewall toolkit or Trusted Information Systems' fwtk) contain basic SMTP filtering that can be used to control outbound spam. In fact, the Mail Message Metering implementation utilized the Juniper firewall toolkit's smtpd as the basis for the message processing. The proxy approach is a minimalistic one, as SpamAssassin contains much more anti-spam functionality built into it. However, they are implemented in C/C++, which may make the proxies more reliable than code written in Perl.

Stopping Spam at the Network

There are ways spam can be controlled by the provider at the network level:

- Blocking access to known open relays via access control lists (ACLs) on routers
- Caller-ID blocking

The downside to these methods is they do take resources on the network components (such as routers), which can cause additional cash outlays by the provider to implement these methods.

BLOCKING ACCESS TO KNOWN OPEN RELAYS

One very effective (but drastic) way to reduce unwanted outbound spam is to simply disallow access to all SMTP servers except for the provider's own mail relays. This could be accomplished by the following ACL on a Cisco router:

```
access-list 101 permit tcp host a.b.c.d any eq smtp
access-list 101 permit tcp host e.f.g.h any eq smtp
```

System-wide procmail filters can be built to assist in the fight against spam.

Perhaps the most important document a service provider has is its Acceptable Use Policy, or AUP.

```
.  
. .  
. .  
access-list 101 deny tcp i.j.k.l.0 0.0.0.255 any eq smtp  
ip any any  
access-list 101 permit
```

The first two access-list statements allow access to legitimate mail relays, and more permit hosts/networks could be added. The third access-list statement denies all other access to port 25 (SMTP) outside of what is specified in the permit list. The final statement allows all other traffic to be routed normally.

A variation on this idea is to block outbound SMTP access to known networks that house open relays, such as Korean networks. A dialup customer should be using the mail relays provided, rather than misconfigured ones located halfway around the world!

OTHER RAS/NETWORK TECHNIQUES

Many spammers will block caller ID to make it harder to track them down. One technique that is used to block spammers from wholesale dialup networks is to disallow outbound SMTP access to anyone who calls in without providing caller ID. This will stop a lot of spam. Also, RAS filters can be loaded dynamically onto the modem ports via RADIUS, allowing SMTP access to a certain set of IP addresses and excluding the rest. In fact, UUNET requires its customers to pass a RADIUS attribute (Ascend-Data-Filter), allowing outbound SMTP access to its wholesale customers' mail relay, and nothing else.

Other tactics that can be attempted include:

- limiting outbound SMTP connection rate
- limiting SMTP bandwidth

This author is not aware where this has been tried "in the wild" on a production network.

Miscellaneous Topics

This section contains odds and ends regarding both inbound and outbound spam.

ACCEPTABLE USE POLICY

Perhaps the most important document a service provider has is its Acceptable Use Policy, or AUP. Without a properly written AUP, it is impossible to legally shut off customers who abuse a provider's network. All organizations, be they providers, small companies, large companies, nonprofits or others should have an AUP. While it takes time and effort to write a good one, the headache it cures in the long run makes it well worth it.

LEGAL ASPECTS

A book could be written on the legal aspects of UCE. In the US the only laws currently governing spam at the federal level surround fax broadcasting (governed by the Federal Communications Commission) and the legality of claims made by spammers (governed by the Federal Trade Commission). Case law is being built every day. In July 2002, Earthlink was awarded US\$25 million in a lawsuit against spammers. The FTC has been active in pursuing spammers who make illegal claims.

In the US, the only codified anti-spam law is at the state level. David E. Sorkin has a great site that summarizes the current status of anti-spam law, both inside and outside the US.

STAFF

At most ISPs, customer support and/or the network operations center personnel handle spam complaints. At Ziplink, the company dedicated approximately two staff positions to handle the influx of spam complaints, with a 70,000 port dial-in network. Many complaints are duplicates, or are sent in error, which causes additional overhead.

Automated systems such as Spamcop work well. However, they are not infallible and do make mistakes. One benefit of such systems is the elimination of duplication of effort that automated systems can provide. Spamcop will stop sending spam reports to the provider, once the provider tells Spamcop the spammer has been deactivated. However, Spamcop continues to send duplicate spam reports, with the same “foot-print” (i.e., source IP address, subject line, etc.) until the provider takes action.

COSTS

The additional strain spam puts on staff, machines, and networks is hard to quantify. If we use an assumption that 33% of all email is spam, that loosely translates into 33% higher costs for the provider. Those two additional staff positions mentioned above could be eliminated if spam were not a problem. A server or two could probably be reallocated at a small- to mid-sized ISP, while a larger provider could probably eliminate more. The upstream network connections, if the provider buys transit, would be less without spam.

USENET NEWS SPAM

Most news servers these days are able to control news spammers without much difficulty. InterNetNews (INN) v2.3.2 has an “exponential backoff” feature. The associated control parameters are:

- backoffauth
- backoffdb
- backoffk
- backoffpostfast
- backoffpostslow
- backofftrigger

Check the man pages for `inn.conf` and search for “backoff” for more information. If the Highwinds Software series (Typhoon/Cyclone/Twister) of news servers is used, a Perl program is available to rate limit article posting. This rate limiting works very well.

PLACES TO SEND YOUR SPAM

Ever wonder where you can send spam you receive (besides to the provider that originates it)? A list of email addresses appears below; if anyone knows of additional email addresses to send junk mail to, please send them and they will be published in a future column. Some of these addresses are just statistics trackers, others are for actual complaints, and some are commercial services which block spam using the email submitted to generate rules for protecting their customers.

<i>spamrecycle@chooseyourmail.com</i>	The spam recycling center (statistics)
<i>uce@ftc.gov</i>	FTC's junk mail address
<i>fraud@uspis.gov</i>	For complaints involving US Postal Service mail
<i>enforcement@sec.gov</i>	For securities-related complaints involving US publicly listed companies
<i>cyberfraud@nasaa.org</i>	For securities-related complaints involving North American publicly listed companies
<i>otcfraud@cder.fda.gov</i>	For food/drug-related complaints
<i>junk@brightmail.com</i>	Honeypot address for Brightmail spam filtering service

Conclusion

There are available tools for ISPs (and others) to control outbound spam. Mail transfer agents (MTAs) such as Sendmail can be configured to allow certain IP address ranges to relay mail, which all organizations running a mail server on the Internet today should employ. Outside of MTAs, Kai's SpamShield can be utilized to control outbound spam, and other mail proxy agents can be useful as well. These open source methods work, but are not perfect and take effort to implement. Steps can be taken at the router/network-device level as well, but these are not adaptive and must be regularly updated. Some proprietary methods (such as Mail Message Metering) do exist but are applicable to certain classes of spam sources (such as large ISPs) and are covered by intellectual property law.

References

- Blackmail: <http://www.jsm-net.demon.co.uk/blackmail/blackmail.html>
- Brightmail: <http://www.brightmail.com/>
- Brightmail anti-spam patents: <http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&u=%2Fmetahtml%2Fsearch-adv.htm&r=0&p=1&f=S&l=50&Query=in%2F%22paul%3B+sunil%22%0D%0A&d=ft00>
- David E. Sorkin's spam-law site: <http://www.spamlaws.com/>
- Email Sanitizer: <http://www.impsec.org/email-tools/procmail-security.html>
- Highwinds Software (Typhoon/Cyclone/Twister): <http://www.highwinds-software.com/discussion/index.html>
- INN: <http://www.isc.org/products/INN/inn-current.html>
- ISP-Planet article on Earthlink spam lawsuit: <http://www.internetnews.com/isp-news/article.php/1430591>
- Kai's SpamShield: <http://spamshield.conti.nu/>
- Mail Message Metering: <http://www.ziplink.net/ziplink/solutions/mmm/>
- Obtuse Systems Juniper firewall toolkit smtpd: <http://www.obtuse.com/smtpd.html>
- POP-before-SMTP: <http://popbsmtp.sourceforge.net/>
- Relay control in Sendmail for roaming users: <http://www.Sendmail.org/~ca/email/roaming.html>
- SpamCon Foundation: <http://www.spamcon.org/>
- SpamCon Foundation's list of places to send junk email: <http://www.spamcon.org/recipients/spam-response/help-statistics.shtml>
- Spamcop: <http://spamcop.net/>
- The SpamBouncer: <http://www.spambouncer.org/>
- Trusted Information Systems fwtk: <http://www.fwtk.org/fwtk/>

remote monitoring with SNMP

Introduction

It is possible to monitor and administer a small number of computers individually, for example, by running an interactive session on each one. We would be most likely interested in observing the CPU, memory, and disk space utilizations or verifying that a particular process is running. We could run commands like `df` or `ps` at regular intervals in order to assure ourselves that everything is fine. We could also create various smart scripts to perform automatic monitoring functions locally and notify the administrator about the exceptions by, say, email. However, this approach has obvious limitations as the number of servers increases, for it demands a manual modification of the local scripts should the threshold values or the email address change. In general, we would want to have available one centralized point (management station) where we could set up the thresholds and process the notifications about the exceptions. We would prefer to have a unified, simple installation and configuration of the part of the monitoring software which runs on the managed nodes since that would allow for more robust and automated installation procedures for a large number of the systems. Also, we would prefer such a monitoring method to be able to work for the different platforms found in data centers nowadays.

SNMP (Simple Network Management Protocol) fits the task well despite some limitations. It takes care of network communication between the management station and managed nodes. It organizes the management information on the client side so that it can be retrieved and modified by the management station via a small number of SNMP operations. It does not process, filter, or correlate the management information retrieved from the clients. It passes the information to other application programs, or, put differently, the management applications take advantage of SNMP in order to communicate with their clients. One of the compelling reasons for using SNMP is the fact that the SNMP daemon or service is a part of the standard installation of all major modern operating systems. Enterprise-grade database systems, firewalls, and other applications often contain an SNMP module which can communicate the application-specific management information. The components of the network infrastructure such as routers or switches support SNMP; many other devices such as uninterruptible power supplies allow for the installation of a card with an embedded SNMP daemon. Therefore, with relatively little extra burden caused by the planning and configuration, we can have a multi-platform, network-monitoring capability based on open standards.

SNMP standards are defined by the RFC documents created by the Internet Engineering Task Force (IETF). The evolution of SNMP has not been straightforward, varying as different ideas were getting more or less attention. There are three versions of the protocol; the newest (SNMP v3) was standardized on March 27, 2002. From the point of view of the IETF, SNMP is a part of the Internet Standard Management Framework and all three versions share the same basic structure and components:

1. Managed nodes, each with an SNMP entity that provides remote access to management information. These are usually called “agents.”
2. One or more SNMP entities with management applications. These are traditionally called “managers.”

by Jozef Skvarcek

Jozef Skvarcek is currently working as a system administrator. He holds a PhD in Physics. Computer technology and science are his long-time hobbies.



jozef@photonfield.net

3. Management protocol to communicate management information between the SNMP entities.
4. Management information.

The architecture of SNMP is modular and it consists of:

1. Data definition language, called Structure of Management Information (SMI). It defines the fundamental data types, object model, and rules for writing the MIB modules.
2. Management Information Base (MIB) modules. They define the objects and event notifications (“traps”).
3. Protocol operations. The operations are performed by Protocol Data Units (PDU).
4. Security and administration.

The SMI can be viewed as a collection of management objects residing in a virtual information store, which is the MIB. Collections of related objects are defined in MIB modules written in the SNMP MIB module language. An object in this scheme is called an “Object Identifier” (OID), an ordered sequence of non-negative integers. OID can also be represented by strings; the whole structure resembles DNS. Only a small number of MIBs (the core set) is required to be supported by each conforming agent.

NAME	DEFINITION
MIB-II	RFC 1213
Interfaces MIB	RFC 1573
SNMPv2 Core	RFC 1907

These provide access to a small common set of management information such as the system name, location, contact information and statistics about the network interfaces. There are a large number of non-mandatory MIB modules from the IETF. Another set of MIB modules are written by hardware and software vendors to support the management of their products. These enterprise-specific MIBs specify objects that lie under the “1.3.6.1.4.1” (“iso.org.dod.internet.private.enterprise”) branch, sometimes simply called the “enterprise” branch. The administration of the enterprise branch is delegated to the vendors who normally provide their MIB modules as text files on some media, say, CD-ROMs or floppy disks that ship with their products. The administrator can study the files in order to determine which OIDs and traps are available. For example, the administrator would likely be searching for the OIDs holding the current values of input/output voltages or output load if he were reading the MIB which came with a power supply unit.

There are a small number of PDUs; the most frequently used are GET, GET-NEXT, SET, and TRAP. The PDUs are encapsulated into the SNMP messages and then are transmitted over the IP network using the UDP protocol. The SNMP daemon usually listens on port 161 for all PDUs except TRAP, which is sent to port 162. If you have installed a network management application with the GUI such as IBM Tivoli Netview, the PDUs are generated by choosing the appropriate items from the menus. NET-SNMP, the excellent open source SNMP agent, implements the PDUs as command-line executables.

SNMP and Security

The IETF explicitly specifies the following security threats:

1. “Modification of information” – This is the danger of the modification of SNMP information during network transmission by an unauthorized person.
2. “Masquerade” – This is the danger that an unauthorized user may assume the identity of another user with more administrative rights.
3. “Disclosure” – This is the threat that an unauthorized person may observe the SNMP communication between the managed agents and the management station.
4. “Message stream modification” – This is the danger that the SNMP messages can be re-ordered, delayed, or replayed in order to force some unauthorized management operations.

Versions 1 and 2 of the SNMP protocol provided hooks for multiple authentication schemes; however, they did not explicitly specify any but a trivial authentication scheme based on “community” strings. The “community” is included within an SNMP message, the message is transmitted in the clear-text format, and then the community is verified by the receiving entity. Obviously, this security scheme does not really solve the problems posed by the threats and is vulnerable to various sorts of attacks. It is the goal of version 3 to address the threats. It employs the traditional concept of a user who is identified by a username and secret key (passphrase). The security information is associated with the user. The cryptographic algorithms MD5 and SHA are used for the authentication and DES for the encryption. Other protocols are permitted by the standard. The standard also mandates time synchronization between the SNMP entities in order to tackle the message stream modification threat. Note, this time synchronization is performed by the SNMP agent software and is independent of other methods of system-time adjustment such as the NTP. These security methods make SNMPv3 operations safe on the public IP networks. On the other hand, user-based authentication adds another user database not correlated with the system accounts in general. That fact increases the complexity of the administration and implies derived security threats. This problem can be handled by installing an SNMP agent software which supports some kind of centralized and secure user authentication. For example, the latest version of NET-SNMP (version 5) can take advantage of the Kerberos protocol for that purpose.

Which SNMP version to use? SNMPv3 seems to be the trivial answer after what has been said in the previous paragraph. Unfortunately, the standard is too new to be fully supported by all software agents. Your choice will likely be limited by the capabilities of your management station, too. For example, popular enterprise-grade network management applications such as Tivoli Netview or HP OpenView support only SNMPv1 and v2. There are several ways to reduce the risks that stem from running SNMPv1 or v2:

1. Set the community strings. Traditionally, “public” is used as the default for the read-only community and “private” is used for the read-write community. Many agents install with these communities pre-defined. They should be changed before the agent is started for the first time.
2. Disable the write access. In most cases the read-only access is sufficient to satisfy management objectives.
3. Allow PDUs from your management station only. The configuration file of an SNMP agent normally allows you to limit the access only from pre-defined IP

addresses or domain names. If you decide to use domain names, then your DNS server should be safe from unauthorized manipulation of the DNS data.

4. Use a private, closed network for SNMP traffic.

As with other software, security vulnerabilities may be discovered in the code occasionally. Therefore, it is important to keep an eye on the security alerts issued for SNMP software you use and apply the security patches whenever necessary.

We shall take a look at the configuration and utilization of the SNMPv3 protocol in an illustrative case in the following section.

A Practical Example

We used a machine called “Jupiter” as the manager station and a machine called “Europa” as the agent. During preparation of this article, Jupiter was a PC running RedHat Linux 7.2 and Europa was a SPARC box running Solaris 8. NET-SNMP agent software version 4.2.3 was used on both systems. The NET-SNMP software on Jupiter came with the distribution, while the one on Europa was compiled from sources. There were several factors that motivated our use of this particular software. NET-SNMP supports SNMPv3; it is open source, which reduces the cost of the management infrastructure; it is under active development; and it is part of the standard installations of many GNU/Linux distributions. Last but not least, despite the complexity of the SNMP implementation, NET-SNMP has been relatively easy to configure and has been working reliably. It will be assumed in this section that the agent has been properly installed on both systems. (We shall provide more information regarding the installation in the next section.) It is assumed that OpenSSL, required for the encryption, has been installed, too.

Our first objective was to retrieve the system uptime from Europa on Jupiter. How did we know to choose this particular information? Every SNMP agent has to support the core set of MIBs, one of which is MIB-II, defined by RFC 1213. We read the document and found the object `sysUpTime` with OID 1.3.6.1.2.1.1.3 (`iso.org.dod.internet.mgmt.mib-2.system.sysUpTime`), described as the time (in hundredths of a second) since the network management portion of the system was last re-initialized. The `sysUpTime` object is a member of “System Group,” which contains other useful objects such as `sysName`, `sysContact`, and `sysLocation`. A MIB may be difficult to read, since the language is meant to be processed by a machine rather than read by a human. (Please consult the books mentioned in the Bibliography section for further information.)

We started by configuring the agent on Europa. We needed to create a user (“john”) with a passphrase “secret123” (the passphrase must have at least 8 characters). We put into file `/var/ucd-snmp/snmpd.conf` the line

```
createUser john MD5 secret123 DES
```

This version of NET-SNMP does not support SHA protocol yet, and therefore we don’t really have a choice but to use MD5 for the authentication. It is possible to use a different password for the DES; you would need to put it behind DES on the line. In the next step it is necessary to create the configuration file `/usr/local/etc/snmp/snmpd.conf`. NET-SNMP uses the View-Based Access Control Model (VACM), defined by RFC 2575. VACM provides the administrator with the capability of allowing a particular user access to only a specified subset of the OID tree at the agent. In our case, we want to give “john” complete read-write access to the management information. Also, we want to enforce SNMPv3 and the highest security level, which means that

“john” will have to use the strong authentication and encryption. Below is an example of the content of a simplified file which satisfies these objectives.

```
# /usr/local/etc/snmp/snmpd.conf
#
# Map the security name into a group name:
#
#   groupName  securityModel  securityName
group johngrp   usm             john

#
# Create a view for us to let the group have rights to:
#
#   name  incl/excl  subtree  mask(optional)
view all   included   .1

####
# Finally, grant access to the view.
#
#   group  context  sec.model  sec.level  prefix  read  write  notif
access johngrp  ""        usm        priv      exact  all   all   none

#
# Set value for `system.sysLocation' object
#
syslocation Datacenter

#
# Set value for `system.sysContact' object
#
syscontact Networking

#
END
#
```

For the sake of brevity, we can't discuss the file in detail; please consult the documentation which comes with the NET-SNMP software. Finally, we are ready to launch the SNMP daemon on Europa with the command:

```
# /usr/local/sbin/snmpd
```

At this point the daemon should be listed among the running processes on Europa. An important transformation to the file `/var/ucd-snmp/snmpd.conf` occurs during the start-up of the daemon. The `createUser` line is replaced by john's security key. For the calculation of the key, the daemon used the password and the IP address of Europa. This fact is worth noting, since the security key would need to be re-created should the IP change.

The `snmpd` is running on Europa, but how can we access the information from Jupiter? We can do it with the GET-NEXT PDU, which is implemented by the command `snmpwalk`. We can run on Jupiter, for example, the following command:

```
$ snmpwalk -v 3 -u john -l authPriv -a MD5 -A secret123 -x DES \
-X secret123 europa .iso.org.dod.internet.mgmt.mib-2.system
```

which retrieves the values of the OIDs in the “system” branch. We get several lines as the output and recognize the OIDs defined by MIB-II. Among them are the following:

```
system.sysDescr.0 = SunOS europa 5.8 Generic_108528-12 sun4u
system.sysObjectID.0 = OID: enterprises.ucdavis.ucdSnmpAgent.solaris
system.sysUpTime.0 = Timeticks: (383093) 1:03:50.93
system.sysContact.0 = Networking
system.sysName.0 = europa
system.sysLocation.0 = Datacenter
```

The zero trailing the OID names is the “instance.” Some OIDs may have multiple instances; for example, the OID describing the mounting point of a disk partition has as many instances as there are partitions.

One may object that the `snmpwalk` command is too long and poses a security risk. A local user may learn the passwords by running the `ps` command while we are executing `snmpwalk`. To deal with this, we can create the file `.snmp/snmp.conf` in our home directory with the following content:

```
defVersion 3
defSecurityName john
defAuthType MD5
defAuthPassword secret123
defPrivType DES
defPrivPassword secret123
defSecurityLevel authPriv
```

This file should be readable only for the user since it includes the passwords. Having this file in place, we are able to simplify the `snmpwalk` command to

```
$ snmpwalk europa .iso.org.dod.internet.mgmt.mib-2.system
```

If we recall our original objective, our task was to retrieve the value of the `sysUpTime` object. We already have the result; the appropriate line is in the output from `snmpwalk`. We can get the value of that single object by using the GET PDU, which is implemented by the `snmpget` command. We can run on Jupiter the following:

```
$ snmpget europa .iso.org.dod.internet.mgmt.mib-2.system.sysUpTime.0
```

which returns as output the line

```
system.sysUpTime.0 = Timeticks: (1015919) 2:49:19.19.
```

Our second objective, which will be less trivial, is to monitor the disk space in the root partition on Europa. The necessary information is provided by Host MIB, defined by RFC 1514. Although this MIB is not mandatory, it is supported by many agents, NET-SNMP among them. The MIB specifies the OIDs under the `iso.org.dod.internet.mgmt.mib-2.host` branch. The command:

```
$ snmpwalk europa .iso.org.dod.internet.mgmt.mib-2.host
```

produces long output containing much interesting information, such as process names currently running on the system, their arguments, partitions, and mounting points, and so on. We are interested in the latest. Among the output were the following lines:

```
host.hrStorage.hrStorageTable.hrStorageEntry.hrStorageIndex.1 = 1
host.hrStorage.hrStorageTable.hrStorageEntry.hrStorageType.1 = OID:
    host.hrStorage.hrStorageTypes.hrStorageFixedDisk
host.hrStorage.hrStorageTable.hrStorageEntry.hrStorageDescr.1 = /
host.hrStorage.hrStorageTable.hrStorageEntry.hrStorageAllocationUnits.1 =
    4096 Bytes
```



```
host.hrStorage.hrStorageTable.hrStorageEntry.hrStorageSize.1 = 756012
host.hrStorage.hrStorageTable.hrStorageEntry.hrStorageUsed.1 = 120936
```

It makes sense, doesn't it? We should verify that the OIDs really are what they seem by reading their descriptions in the MIB file. There were many instances of these OIDs, but we picked up instance 1 since the OID `hrStorageDescr.1` returned "/" and that was what we had been looking for. Then we found the rest of the OIDs with that instance number. The size of the file system could now be calculated by multiplying `hrStorageSize` by `hrStorageAllocationUnits`, and the result would be in bytes. Space utilization as a percentage can be calculated using the formula

```
( hrStorageUsed / hrStorageSize ) * 100.
```

The monitoring can be automated by creating, say, a Perl script which could regularly poll the agent. Such a script could parse a configuration file where we could define the threshold. The script would then fire up an alarm if space utilization greater than the threshold were detected. The procedure could be further scaled up to monitor a large number of systems.

NET-SNMP Compilation Notes

The NET-SNMPv4.2.3 sources were downloaded from the project's home site. The agent compiled without any problem on RedHat Linux 7.2 with GCCv2.96 and on Solaris 8 with Sun Work Shop 6.1. After the sources are unpacked we need to run the "configure" script. The script takes several arguments; you can use:

```
# ./configure --help
```

in order to get the list. SNMPv3 is supported as the default, but the Host MIB is not. It is necessary to include the support by the command-line argument:

```
# ./configure --with-mib-modules="host"
```

The agent can support several more MIBs and other functionality; for example, it is possible to include the support for the `tcp_wrappers`. The configuration uses `/usr/local` as the default installation prefix, which is why the `snmpd.conf` file is located in the `/usr/local/etc/snmp` directory. It is possible to change the location of this and other NET-SNMP files by choosing the appropriate arguments for the configure script. Please read the included documentation for more information. Then we can compile the binaries:

```
# make
```

and install them with

```
# make install
```

The distribution does not provide provisions for those who wish to create the software packages. There are many Makefile files inside the source tree. One has to manually edit the install targets so that the software will be installed in a directory suitable for the build of the package.

Notes on SNMP Agents on Some Operating Systems

REDHAT LINUX 7.2

Typically, two packages are installed on a production box from RedHat 7.2 CD media:

- ucd-snmp
- ucd-snmp-utils

A third package `ucd-snmp-devel` which provides the API, can be installed on a development system. Don't be confused by the names; the packages are the NET-SNMP distribution version 4.2.3. (NET-SNMP had been previously known as UCD-SNMP.) The full path to the configuration file is `/etc/snmp/snmpd.conf`. The packages from the original CDs contained a security vulnerability and the `snmpd` daemon did not start under certain circumstances. We replaced them with the latest versions of the RPM packages from the home site of the NET-SNMP project and that fixed both problems.

SOLARIS

The agent is part of the product called "Solstice Enterprise Agents," which is installed automatically during a typical installation. The product consists of five packages:

- SUNWsacom
- SUNWsasnm
- SUNWsadmi
- SUNWmibii
- SUNWsasdk

where the last one provides the development platform and is not installed if only a runtime environment is required. The configuration files are located in the `/etc/snmp/conf` directory. However, in our opinion this agent is difficult to configure and the documentation is not sufficient. Also, the Host MIB does not seem to be supported and neither does SNMPv3. Therefore we suggest using NET-SNMP instead.

WINDOWS 2000

The agent is installed automatically during a typical installation. Make sure that the "SNMP Agent Service" is enabled for automatic startup. The agent is configured through the SNMP Agent Service Properties window popup in a manner similar to the other system components. The relevant registry keys are located in `HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\SNMP`.

The documentation can be found in the Windows 2000 Server Resource Kit, TCP/IP Core Networking Guide. The agent works well on small systems, but it seems to have certain problems serving the Host MIB information on larger multi-processor boxes. It does not support SNMPv3.

Conclusions

The scope of this article is limited to describing the configuration of the SNMP agent and demonstrating use of management information for the monitoring of the system parameters. The management information served by the Core and the Host MIBs we discussed is sufficient for monitoring:

1. Disk space utilization
2. CPU load
3. Swap space utilization
4. Running processes
5. TCP/IP statistics

and other parameters. The aim is to give the interested reader enough knowledge for starting practical, secure remote monitoring. The other parts of the SNMP are left to the reader for further research.

Although much can be achieved by scripting SNMP polls, as hinted in the paragraph about the monitoring of disk space usage, in many cases it would be desirable to have a GUI-based network management application with a rich set of features. A few commercial products such as IBM Tivoli Netview or HP OpenView, as well as free ones such as GxSNMP, are available, although they do not support SNMPv3 yet.

We did not write much about the TRAP PDU. When a problem comes up, traps are emitted by certain hardware and software components such as network routers, UPSes, and databases that support the feature. The traps are usually processed by a network management application, which can sort them out, perform correlations, and issue notifications. In many cases the traps are more efficient than a periodical polling, since they do not cyclically consume the network bandwidth and host resources.

Acknowledgments

We would like to thank the members of the UniGroup in New York City for valuable suggestions.

Bibliography and Web Sites

BOOKS

Essential SNMP, by D.R. Mauro and K.J. Schmidt, O'Reilly, 2001

Practical Guide to SNMP v3 and Network Management, by D. Zeltserman, Prentice Hall, 1999

SNMP Network Management, by P. Simoneau, McGraw-Hill, 1999

Understanding SNMP MIBs, by D. Perkins and E. McGinnis, Prentice Hall, 1997

WEB SITES

GxSNMP: <http://www.gxsnmp.org/>

HP OpenView: <http://www.openview.hp.com/>

IBM Tivoli Netview: <http://www.tivoli.com/products/index/netview/>

NET-SNMP (UCD-SNMP): <http://net-snmp.sourceforge.net/>

RFC: <http://www.ietf.org/rfc.html>

SNMPv3: <http://www.ibr.cs.tu-bs.de/projects/snmpv3/>

The details of the standardization of SNMPv3: <http://www.ietf.org/IESG/actions.html>

cabling

Just the Tip of the Infrastructure

by James Carlini

James Carlini is president of Carlini & Associates, Inc., a management consulting firm focused on developing marketing strategies and applications of strategic integrated information technology. He offers seminars.



carlini@northwestern.edu
773-370-1888

Cabling problems are common in all buildings and data centers. In performing cabling assessments for more than 15 years for many property management companies and major organizations (as well as acting as an expert witness on several multimillion-dollar lawsuits related to cabling infrastructure problems), I can safely say that your building has cabling problems.

From Wacker Drive to Kansas City to Beverly Hills, I have never walked through a building without finding major problems resulting from a lack of management of the inside wiring and cabling facilities. These problems are going to cost you and your organization a lot of money – maybe even the sustainability of your business.

Huge Insurance Risk Second-Round Funding

Some years ago, most property management firms and large organizations didn't care about cabling problems. They were too busy with more important things, like what type of espresso machine they should be installing in the lunchroom. They would let maintenance go until something drastic happened: overhead cabling troughs would get maxed out; tenants (in properties) would move, leaving abandoned cable under floor cable troughs so that raceways got maxed out; a new application would have to be installed, and dead (abandoned) cable would need to be removed to make room for the new network media.

Today, there are some new rules pertaining to maintaining inside wiring within an organization. Managers in charge have to understand that there is huge liability now if cabling is messed up.

The insurance companies have finally perked up and are pushing the issue of cabling infrastructure. National electrical codes are specifying that abandoned cable must be eliminated and that cable ducts must be free of dead cable.

In liability cases, if a fire spreads due to cabling conduits not being properly fire stopped (covered with a flame-retardant material so the pathway, or riser system, cannot act as a “chimney” for smoke and other toxic fumes to travel through the building), the insurance company can claim that your organization's building was not up to code. No payment. Claim dismissed.

Let's say that the fire caused millions of dollars of damage. Your organization needs the money to restore its capabilities – quickly. The insurance company can wiggle out of paying the claim because they can show that the cabling was not up to the national electrical code as well as the municipality's building code.

Is Your Facility Management Awake?

A fast way to find out if the people responsible for cabling within your facility are “on top of the job” is to ask them if they know about the TWINS in the building.

What are the TWINS? It's a fast view of the health of your cabling, which is the lifeline of your business. TWINS stands for Total pair, Working pair, In-service pair, Non-working pair, and Spare pair.

An example of using TWINS would be to evaluate the amount of spare capacity you have coming to the building. You might think you have a lot of spare capacity because you know you have a 5,000-pair cable coming into the building and you are only using 2,000 of that 5,000.

Here is a quick example of how that assumption can be grossly misleading:

Total Pair: 5,000 – Working Pair: 2,200

In-Service Pair: 2,000 – Non-Working Pair: 3,000 – Spare Pair: 200

(Determine non-working pairs through testing.)

All of a sudden, you realize that you don't have the extra capacity you thought you had. You thought you had more than enough for that new telemarketing department you were going to move into the building that needed 500 pair for their incoming lines. Now it's a crisis. Don't think that the phone company is going to come rescue you.

I have actually seen a building that could not be leased up because they ran out of cable to the building. The phone company diverted "spare cable" to another building being built, and the existing building could not lease up the last 20 percent of the office space. For those who are curious about how long it took for them to run more cable to that building, it was a priority order and it took 18 months. If you or your telecom manager really know the TWINS formula for cabling, you may save yourself a crisis or two. Try asking about this TWINS check with your facilities manager.

Data Centers Are Centers for Potential Disaster

So many organizations have built data centers and call centers on the cheap. The newer centers are sometimes more susceptible to cabling problems because the people in charge didn't spend enough money to do a proper job of insulation, fire stopping, compartmentalizing, and creating a redundant approach.

There are many areas for improvement that are often overlooked because the people in charge may not know what a good data center should be. A great example of the right job is the Chicago E-911 center. Another is AT&T's 10 S. Canal facility, where they have four jet aircraft engines as power back-ups if ComEd fails.

I used to take my students through both because I wanted to show them what "the right way" looked like. These were great field trips for those professionals who were jaded and thought they had seen it all. They were impressed and shocked at the level of reliability and redundancy, but jet aircraft engines can chew up a lot of fuel and are not cheap. Neither is filling up their 330,000-gallon tank (a big price difference compared to a PC surge protector).

Fiber-optic cable and copper cables running through cable trays with the proper loading and strapping gave a real-life example, rather than just talking about a quality standard. No overfilled trays, no spaghetti cable messes or other problems that I saw in so many other buildings – just a perfect design.

Pay Now or Pay Dearly Later

So many organizations have gotten by with substandard design and maintenance on cabling and data centers that they might read this with a "so what" attitude.

Those are the companies that will eventually go out of business or be acquired by their competitors when they hit their first major outage and can't get their data center working, or when they have a fire, a bad design, or disaster and find out they can't collect the insurance because the insurance company sends out some person to review the problem before writing a check for \$10 million.

Some of you know-it-alls say that will never happen and insurance companies will always pay out. Sorry to burst your bubble. I know this happens because I was that person, and the check wasn't written.

I have actually seen a building that could not be leased up because they ran out of cable to the building.

the bookworm

BOOKS REVIEWED IN THIS COLUMN

HACKER'S DELIGHT

HENRY S. WARREN, JR.

Boston, MA: Addison-Wesley, 2002. Pp. 306.
ISBN 0-201-91465-4.

UNIVERSAL COMMAND GUIDE

GUY LOTGERING & THE UCG TRAINING TEAM

New York: Hungry Minds, 2002. Pp. 1591 +
CD-ROM. ISBN 0-7645-4833-6.

ABSOLUTE BSD

MICHAEL LUCAS

San Francisco: No Starch Press, 2002. Pp. 616.
ISBN 1-886411-74-3.

SENDMAIL PERFORMANCE TUNING

NICK CHRISTENSON

Boston, MA: Addison-Wesley, 2002. Pp. 228.
ISBN 0-321-11570-8.

ECONOMETRICS, VOL. 3.

DALE W. JORGENSON

Cambridge, MA: MIT Press, 2002. Pp. 461.
ISBN 0-262-10094-0.

THE COMMUNICATIONS TOOLKIT

P.H. LONGSTAFF

Cambridge, MA: MIT Press, 2002. Pp. 271.
ISBN 0-262-12246-4.

THE ESSENTIAL GUIDE TO NETWORKING

JIM KEOGH

Upper Saddle River, NJ: Prentice Hall, 2001.
Pp. 407. ISBN 0-13-030548-0.

THE PRACTICAL SQL HANDBOOK, 3RD ED.

J.S. BOWMAN ET AL.

Reading, MA: Addison-Wesley, 1996. Pp. 454.
ISBN 0-201-44787-8.

ORACLE WEB APPLICATIONS

ANDREW ODEWAHN

Sebastopol, CA: O'Reilly, 1999. Pp. 256. ISBN
1-565-92687-0.

XML AND SQL SERVER 2000

JOHN GRIFFIN

Indianapolis, IN: New Riders, 2001. Pp. 384.
ISBN 0-735-71112-7.

WEB DATABASE APPLICATIONS

HUGH E. WILLIAMS & DAVID LANE

Sebastopol, CA: O'Reilly, 2002. Pp. 582.
ISBN 0-596-00041-3.

by Peter H. Salus

Peter H. Salus is a member of the ACM, the Early English Text Society, and the Trollope Society, and is a life member of the American Oriental Society. He is Chief Knowledge Officer at Matrix NetSystems. He owns neither a dog nor a cat.

peter@matrix.net



This is going to be a strange, disjunctive column.

I was going to rest a bit and write about the SQL books I've been looking at since June. But a number of things have hit my mailbox, and the SQL stuff will just take some space at the end of this piece.

There are five books I really enjoyed this past few weeks. One of them is a little out of line, but I think worthwhile. The others are obvious choices.

Hacking

Hank Warren has been hacking for about 40 years. In that time he has come up with a number of neat ways to do math and some algorithms that are quite useful.

Hacker's Delight is, to me, the computer's response to W.W. Sawyer's *Mathematician's Delight*, which I discovered when I was in high school. Warren has made the world of elegant and efficient hacking come alive, and has managed to be useful at the same time.

Commands

The Universal Command Guide weighs in at about 4 kilos. It may be the most unreadable book I've got; but it's also the most useful one I've seen.

Every command for Windows 95, 98, Me, NT 4.0, 2000, XP; Solaris 7.8; AIX 4.3.3; OpenBSD 2.7; RedHat Linux 7; NetWare 3.12, 4.11, 5.1.6; Mac OS 9.1; and DOS 6.22 is listed. Over 8000 of

them. And they're indexed and cross referenced.

The CD-ROM is very well done, and the indexing is superb. You may not need this tome, but you do need the CD.

FreeBSD

I read most of *Absolute BSD* on my flight home from the security symposium. A very fine piece of work, it isn't about how to implement BSD solutions, but it is about managing systems in situ. Its big lack is that there's no bibliography.

Sendmail

Christenson has not written an ordinary book. This is a book for the sysadmin who spends a lot of time working on her mail servers. UNIX mail servers. All over the world, more mail servers run Sendmail than anything else. About 15 years ago, I looked at the `sendmail.cf` file running on our Sun 3/150. I hope to never have to look at one again.

Featuring Sendmail 8.12, this book may possibly enable me to never look at one again. It is not a replacement for Costales and Allman (1997), but if you meddle with mail servers, you need this volume.

Hard-Core Economics

Dale Jorgenson is a professor of economics at Harvard. But don't let that or the fact that his subject is supposed to be tedious put you off.

Econometrics 3 is the third volume of Jorgenson's papers. It is subtitled "Economic Growth in the Information Age." I was really taken by a number of the articles in this hefty book: "Information Technology and the US Economy" and "Computers and Growth" serve as a great introduction to Jorgenson's interests. "What Ever Happened to Productivity Growth?" and "Did We Lose the War on Poverty?" were really enlightening.

book reviews

Two Other Good Books

While I'm on a roll, I want to tip my hat to P.H. Longstaff, whose interests are in communications business and policy. As we recognize that computers (and the Internet) are a part of that communications business, it becomes ever more necessary to try to understand it. *The Communications Toolkit* does a really good job of providing the flexible strategies we'll need in the future.

Keogh's book is the best introduction to networking for your grandmother I've come across. It is so simple and straightforward that even politicians should be able to comprehend it. At times Keogh dumbs things down a great deal, but this may be necessary.

SQL

After I read the Sleepycat BDB book, I realized that I needed to know more about SQL and its uses. I looked at and read too many items. But here are the four that I found most useful.

Bowman and her colleagues produced their volume over five years ago. Though there are some parts that are dated, I think it may be the very best book I've seen on SQL.

Odehahn's book is also a bit dated, but it is both clear and well written. The early chapters are a bit too introductory, but the final four were very, very good. On occasion, I wanted a bit more detail, though.

Griffin's book is far more up-to-date. It's good, with many useful examples.

Williams and Lane is also good, especially the PHP and MySQL sections. I can recommend all four.

INCIDENT RESPONSE:

A STRATEGIC GUIDE TO HANDLING SYSTEM AND NETWORK SECURITY BREACHES

EUGENE SCHULTZ AND RUSSELL SHUMWAY
New Riders Publishing, ISBN 1-578-70256-9

Reviewed by Anton Chuvakin
Anton Chuvakin, Ph.D., GCIA, is a senior security analyst with a major security company.

anton@netForensics.com

Incident Response by Eugene Schultz and Russell Shumway – the third book with this title that I have reviewed – had to overcome a certain expectation barrier, even though the authors are recognized experts in the security field. It passed the barrier with flying colors, being different but still covering many facets of the intricate incident response (IR) process, with sections on technology, procedures, and, especially, people.

The book starts with security basics. A risk assessment overview with loss estimates and a summary of digital risks (such as privilege escalation, break-in, denial-of-service, etc.) is provided. This material appears to be useful mostly for newcomers to the security field. Formal six-stage incident-response methodology is then presented by the authors: the preparation, detection, containment, eradication, recovery, and follow-up (PDCERF) process helps create a solid skeleton to support the fluid form of the IR process.

Admittedly, the book is less hands-on oriented than some other IR manuals; the reader will not find things like computer forensics-tool command-line options and ext2fs file system internals here. However, the book shines in its coverage of the human aspect of incident response. Written by an ex-CIA Ph.D. psychologist, the amazing chapter on social sciences and incident response covers a diverse range of topics: cyber-crime profiling techniques, such as victim counseling and victimology; “modus operandi” identification; attack pattern

recognition; establishment of threat level and communication with attackers. The chapter provides an exciting journey into the mind of a computer criminal, a cyber-sleuth, and a cybercrime victim. Also covered are insider attacks, often considered to be the doom of information security. The question “Why do insiders attack?” is thoroughly analyzed. The author overlays social methods on standard incident-response procedure (detection/containment/eradication/recovery) to help understand the crucial role the human element plays in any security incident.

Two chapters are devoted to high-level computer forensics overview. Hard disk basics are explained – FAT, cluster, secure deletion are all given appropriate space. The book goes on to talk about the “guiding principles” of the investigation. A brief overview of forensic software and hardware is also provided but only serves to familiarize the reader with the names of common packages and utilities. For example, TCT coroner kit is only given about 15 lines of text.

Honeypots also take an honorable place in the book. Their role in IR is studied in detail and is deemed important. Honeypots are also tied to PDCERF, and their value in studying attackers, shielding IT resources, and even gathering evidence for court prosecution is recognized. Some common ways of implementing honeypots (such as via virtual environment) are discussed. The authors even digress to touch upon the ethical implication of honeypots.

Another gem is a stimulating chapter on future directions in IR. The ambitious prediction of intelligent automated incident response and attacker tracking tools is made by the authors. While it is known that automated response to security incidents must be viewed with caution, the potential seems to exist for future automated IR “helpers.”

book reviews

An overview of legal issues is a must for any IR book. A brief and to-the-point section on US laws and international cybercrime treaties is available.

Last but not least, a short response and reporting checklist is compiled by the authors. It is based on the six-step IR process and will help investigators to structure their efforts and assist with data collection. Also included is a copy of a “Site Security Handbook” (RFC2196), with an extensive list of references.

Overall, the book is an extremely useful guide for security managers and those tasked with organizing/maintaining incident response teams. Skilled computer crime investigators will not learn anything new from this book, but they will likely enjoy the book nevertheless.

HONEYPOTS: TRACKING HACKERS

LANCE SPITZNER
Addison Wesley Professional,
ISBN: 0-321-10895-7, 480 pp.

Reviewed by Anton Chuvakin

If you liked *Know Your Enemy* by the HoneyNet Project, you will undoubtedly like HoneyNet Project founder Lance Spitzner's *Tracking Hackers* much more. In fact, even if you did not like *Know Your Enemy*, you will probably be impressed with the new book on honeypots and their use for tracking hackers.

The structure of the book is different from *Know Your Enemy*. Spitzner begins with his first honeypot penetration experience and goes on to talk about all aspects of honeypots. In-depth and structured background on honeypot technology is provided. Honeypots are sorted by the level of interaction with the attacker they are able to provide.

In addition, the book covers the business benefits of using honeypots. By classifying honeypots by their value in the areas of prevention, detection, and response

(exactly as done in HoneyNet Project white papers), Spitzner analyzes honeypot technology's contribution to an overall security posture. He also describes the differences between research and production honeypots and demonstrates the benefits of both for various deployment scenarios.

A large part of the book is devoted to particular honeypot solutions – “honeyd” by Niels Provos, plus several commercial honeypots – with detailed explanation of how they work. For example, there is a clear description of ARP spoofing and how it is used by the “honeyd” honeypot daemon. An interesting chapter on “homegrown” honeypot solutions (such as the ones used to capture popular worms of 2001) sheds some light on the simplest honeypots that can be built for specific purposes, such as one to capture a popular attack by means of a simple port listener. Use of a UNIX `chroot()` jail environment for honeypots is also analyzed.

Of course, a special chapter is devoted to honeynets, HoneyNet Project's primary weapon in the war against malicious hackers. Generation II honeynet technology is introduced in the book. The chapter not only lists honeynet deployment and maintenance suggestions but also talks about the risks of honeynets.

Another great feature of the book is a chapter on honeypot implementation strategies and methods, such as using NAT to forward traffic to a honeypot and DMZ honeypot installation. The information is then further demonstrated using two full honeypot case studies, from planning to operation.

What is even more important, maintaining the honeypot architecture, is covered in a separate chapter. Honeypots are a challenge to run, mainly since no “lock it down and maintain state” is possible. One has to constantly build defenses and

hide and dodge attacks that cannot be defended against.

Tracking Hackers also has a “Legal Issues” chapter, written with a lot of feedback from a Department of Justice official. It dispels some of the misconceptions about honeypots, such as the “entrapment” issue, and summarizes wiretap laws and related data-capture problems.

The book is almost the cutting edge of honeypot research and technology; to truly get the cutting edge and learn about the HoneyNet Project's latest activities in detail, wait for the second edition of *Know Your Enemy* (coming out next year). In the “Future of Honeypots” chapter, Spitzner includes material on honeypot-based early warning system and distributed deployments, analysis of new threats, expanding research applications, and making honeypots easier to deploy and maintain.

To conclude: Marcus Ranum's enthusiastic preface is not an overstatement. It is indeed a great book, both for security professionals and for others interested in this exciting technology. While I was already familiar with most of the information in the book, it was a fascinating read! This book is a real page turner.

SAGE Elections

by Bryan Andregg

Chair, SAGE 2002
Election Committee



andregg@sage.org

“Autumn approaches
in descending darkness
leading to LISA.

The year’s end
means the term’s end
and elections
are upon us.”

As LISA approaches this year, the end of the term for the current SAGE Executive Committee is upon us. As the Chair for the 2002 Election Committee it is my job to set the schedule for the elections, make sure that a mechanism is in place for them, and make sure that the Nominating Committee provides a suitable slate of candidates. That said, I think my real job is to make sure people get involved, promote candidates, and VOTE! From elections past most members seem to see this last step, the voting,

as an artifact of some pre-historic process that they shouldn’t bother themselves with.

The schedule for elections is laid out in the SAGE policy document; it was newly amended this year for more flexibility. The key policies summarized are: nominations are open for two months before elections, a candidates forum is scheduled for LISA, elections take place a couple of weeks after LISA, eligibility to vote is set based on the election date. Specific to this year, the schedule is:

Nominations open:	Sept. 1, 2002
Nominations close:	Nov. 22, 2002
Candidates Forum:	LISA, Nov. 7th, 8:00 p.m.
Eligibility Date:	Nov. 22, 2002
Balloting opens:	Dec. 2, 2002
Balloting closes:	Dec. 15, 2002
Results announced:	As soon as practical following balloting

The mechanism for voting will be on-line voting again this year. For those of you who remember, two years ago was our first attempt at on-line voting and it wasn’t a resounding success. Operating System and browser compatibility requirements mired an alright system. This year we are implementing a system without those limitations. I will be working with the creator of the on-line balloting system to have it donated so

that election costs are kept at a minimum.

The Nominating Committee this year is being led by the estimable Cat Okita. Cat will be assembling a committee in the next couple of weeks (after I write this; it should be together by publication) to solicit and generate nominations. Email with suggested nominations or comments to that committee can be sent to exec-nom-com@sage.org.

The SAGE Exec believes in an open election process. So, in addition to the nominating committee the official SAGE policy document also specifies that any five members of SAGE acting together can nominate a candidate. These nominations should be submitted to the SAGE secretary, Trey Harris, by sending email to trey@sage.org.

Candidates will be able to address the membership through SAGEwire, the interactive portion of SAGEweb. Members will be able to address questions for candidates through SAGEwire forums and see the responses in the same. Follow-up comments or questions are encouraged.

I am the chair of the election committee, so any questions about what’s been in here should be addressed to me at andregg@sage.org. That said, don’t forget to VOTE!

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 the latest booklet when they join SAGE, and they receive a 33% member discount on all SAGE booklets. In addition SAGE members can freely access the full texts of the booklets on the Web.

SAGE sponsors an in-depth annual survey of sysadmin salaries collated with job responsibilities. Results are available to members online.

The SAGE Web site offers a members-only Jobs-Offered and Positions-Sought Job Center. SAGE sponsors members-only mailing lists. The archive of the SAGE members list is available on the Web for members only.

SAGE EXECUTIVE DIRECTOR

Rob Kolstad: kolstad@sage.org

SAGE MEMBERSHIP

office@sage.org

SAGE ONLINE SERVICES

list server: majordomo@sage.org
Web: <http://www.sage.org/>
<http://SAGEwire.sage.org>
<http://SAGEweb.sage.org>
<http://www.sagecert.org>

SAGE STG Executive Committee

PRESIDENT:

David Parter parter@sage.org

VICE-PRESIDENT:

Geoff Halprin geoff@sage.org

SECRETARY:

Trey Harris trey@sage.org

TREASURER

Bryan C. Andregg andregg@sage.org

EXECUTIVES:

Tim Gassaway gassaway@sage.org

Gabriel Krabbe gabe@sage.org

Josh Simon jss@sage.org

USENIX news

USENIX MEMBER BENEFITS

As a member of the USENIX Association, you receive the following benefits:

FREE SUBSCRIPTION TO *;login:*, the Association's magazine, published six times a year, featuring technical articles, system administration articles, tips and techniques, practical columns on security, 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: www.usenix.org/publications/login/

ACCESS TO PAPERS from the USENIX Conferences online starting with 1993: www.usenix.org/publications/library/proceedings/

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.

SPECIAL DISCOUNTS on a variety of products, books, software, and periodicals. See <http://www.usenix.org/membership/specialdisc.html> for details.

FOR MORE INFORMATION REGARDING MEMBERSHIP OR BENEFITS, PLEASE SEE

<http://www.usenix.org/membership/>

OR CONTACT

office@usenix.org
Phone: 510 528 8649

Twenty-five Years Ago in *;login:*

by Peter H. Salus

USENIX Historian

peter@usenix.org

Yes. It was *;login:* 25 years ago.

In October 1977 the big problems were: being late with *;login:*; back orders on the third distribution tape; lack of contributions for a fourth distribution.

The news was: the new Toronto software release; "You are reminded that a new Toronto release must be submitted even if you had a previous release"; a one-day West Coast meeting in January; and "The membership in the Users' Group now exceeds 250."

In November 1977 there were three major announcements. The first was of the death of Joseph F. Ossanna, the author of NROFF/TROFF. (Joe had given a paper at the May 1977 meeting in Champaign-Urbana.) This was a major loss to the field as a whole and to Bell Labs in particular. (Ten years later, in the October 1987 issue of *;login:*, Jaap Akkerhuis reviewed Emerson & Paulsell, *troff Typesetting for UNIX Systems*. I still use groff/troff.)

The second (also relating to Bell Labs) was that *The C Programming Language* by Ritchie and Kernighan was to be published by Prentice-Hall – at \$9.95. Dennis and Brian, if I've not said it often enough, thank you very much.

Third, there was to be a meeting (in May) in New York at Columbia University.

It's worth noting that the May 1978 meeting at Columbia, chaired by Lou Katz, was attended by 350 folks, by far the largest meeting of UNIX users up to then.

Oh, yes. The October 1987 issue of *;login:* also carried an article by Charlie Sauer et al. on "RT PC Distributed Services: File System." It really seems long ago!

Finally, in October 1987, USENIX ran two workshops: the Fourth Computer Graphics Workshop in the Cambridge (MA) Marriott (thanks, Tom Duff and, again, Lou Katz); and the POSIX Portability Workshop in the Berkeley (CA) Marina Marriott (thanks, Jim McGinness and, especially, John Quarterman).

I was at both workshops. They were exciting.

USENIX SUPPORTING MEMBERS

Freshwater Software
Interhack Corporation
Lucent Technologies
Microsoft Research
Motorola Australia Software Centre
New Riders Publishing

OSDN
Sendmail, Inc.
Sun Microsystems, Inc.
Sybase, Inc.
Taos: The Sys Admin Company
UUNET Technologies, Inc.
Zimian

SPONSORED BY **USENIX**, THE ADVANCED COMPUTING SYSTEMS ASSOCIATION,
AND **SAGE**, THE SYSTEM ADMINISTRATORS GUILD

LISA'02

16TH SYSTEMS ADMINISTRATION CONFERENCE

PHILADELPHIA, PENNSYLVANIA NOVEMBER 3-8, 2002

Strengthening the practice of system administration
through technical education, training, and research

- Improve your systems and networks with performance tuning, automation, monitoring, security practices, and intrusion detection.
- Create high-availability servers and disaster recovery plans.
- Make your DNS and LDAP servers more useful, robust, and efficient.
- Tune your storage solutions with AFS, SAMBA, SANs, NAS, and beyond.

Learn from solution-based
technical presentations, including:

- **KEYNOTE, Jim Reese:**
Scaling the Web:
An Overview of Google
- **Larry Wall:** Perl 6
- **Paul Vixie:** Internet Governance,
Peering, and Legislation
- **Curtis Preston:** Disk-to-Disk Backup
- **Len Sassaman:** Internet Privacy
- **Paul Proctor:** Intrusion Detection
- **Daniel V. Klein:** Spam
- **Brad Knowles:** Domain Name Servers
- **Tony Bourke:** Server Load Balancing

Stay informed—LISA offers over
35 tutorials on cutting-edge
technologies and techniques

Register by October 11, 2002, and SAVE! www.usenix.org/lisa02

conference reports

This issue's reports focus on the 2002 USENIX Annual Technical Conference, on the 2002 Linux Kernel Developers Summit (Page 87), and on Java™ VM '02 (Page 91).

OUR THANKS TO THE SUMMARIZERS:

For the USENIX Annual Technical Conference:
Josh Simon, who organized the collecting of the summaries in his usual flawless fashion

Steve Bauer
Florian Buchholz
Matt Butner
Pradipta De
Xiaobo Fan
Hai Huang
Scott Kilroy
Teri Lampoudi
Josh Lothian
Bosko Milekic
Juan Navarro
David E. Ott
Amit Purohit
Brennan Reynolds
Matt Selsky
J.D. Welch
Li Xiao
Praveen Yalagandula
Haijin Yan
Gary Zacheiss

For the Linux Kernel Summit:
David B. Sharp

For Java™ VM:
Jose F. Osorio

2002 USENIX Annual Technical Conference MONTEREY, CALIFORNIA, USA JUNE 10-15, 2002

ANNOUNCEMENTS

Summarized by Josh Simon

The 2002 USENIX Annual Technical Conference was very exciting. The general track had 105 papers submitted (up 28% from 82 in 2001) and accepted 25 (19 from students); the FREENIX track had 53 submitted (up from 52 in 2001) and accepted 26 (7 from students).

The two annual USENIX-given awards were presented by outgoing USENIX Board President Dan Geer. The USENIX Lifetime Achievement Award (also



James Gosling

known as the "Flame" because of the shape of the award) went to James Gosling for his contributions, including the Pascal compiler for Multics, Emacs, an early

SMP UNIX, work on X11 and Sun's windowing system, the first encrypt, and Java. The Software Tools Users Group (STUG) Award was presented to the Apache Foundation and accepted by Rasmus Lerdorf. In addition to the well-known Web server, Apache produces



Rasmus Lerdorf

Jakarta, mod_perl, mod_tcl, and XML parser, with over 80 members in at least 15 countries.

KEYNOTE ADDRESS

THE INTERNET'S COMING SILENT SPRING
Lawrence Lessig, Stanford University
Summarized by David E. Ott

In a talk that received a standing ovation, Lawrence Lessig pointed out the recent legal crisis that is stifling innovation by extending notions of private ownership of technology beyond reasonable limits.

Several lessons from history are instructive: (1) Edwin Armstrong, the creator of FM radio technology, became an enemy to RCA, which launched a legal campaign to suppress the technology; (2) packet switching networks, proposed by Paul Baron, were seen by AT&T as a new, competing technology that had to be suppressed; (3) Disney took Grimm tales, by then in the public domain, and retold them in magically innovative ways. Should building upon the past in this way be considered an offense?

"The truth is, architectures can allow"; that is, freedom for innovation can be built into architectures. Consider the Internet: a simple core allows for an unlimited number of smart end-to-end applications.

Compare an AT&T proprietary core network to the Internet's end-to-end model. The number of innovators for the former is one company, while for the latter it's potentially the number of people connected to it. Innovations for AT&T are designed to benefit the owner, while the Internet is a wide-open board that allows all kinds of benefits to all kinds of groups. The diversity of contributors in the Internet arena is staggering.

The auction of spectrum by the FCC is another case in point. The spectrum is usually seen as a fixed resource characterized by scarcity. The courts have seen spectrum as something to be owned as property. Technologists, however, have shown that capacity can be a function of

architecture. As David Reed argues, capacity can scale with the number of users – assuming an effective technology and an open architecture.

Developments over the last three years are disturbing and can be summarized as two layers of corruption: intellectual-property rights constraining technical innovation, and proprietary hardware platforms constraining software innovation.

Issues have been framed by the courts largely in two mistaken ways: (1) it's their property – a new technology shouldn't interfere with the rights of current technology owners; (2) it's just theft – copyright laws should be upheld by suppressing certain new technologies.

In fact, we must reframe the debate from “it's their property” to a “highways” metaphor that acts as a neutral platform for innovation without discrimination. “Theft” should be reframed as “Walt Disney,” who built upon works from the past in richly creative ways that demonstrate the utility of allowing work to reach the public domain.

In the end, creativity depends upon the balance between property and access, public and private, controlled and common access. Free code builds a free culture, and open architectures are what give rise to the freedom to innovate. All of us need to become involved in reframing the debate on this issue; as Rachel Carson's *Silent Spring* points out, an entire ecology can be undermined by small changes from within.

INVITED TALKS

THE IETF, OR, WHERE DO ALL THOSE RFCs COME FROM, ANYWAY?

Steve Bellovin, AT&T Labs – Research
Summarized by Josh Simon

The Internet Engineering Task Force (IETF) is a standards body, but not a legal entity, consisting of individuals (not organizations) and driven by a consensus-based decision model. Anyone who “shows up” – be it at the thrice-

annual in-person meetings or on the email lists for the various groups – can join and be a member. The IETF is concerned with Internet protocols and open standards, not LAN-specific (such as Appletalk) or layer-1 or -2 (like copper versus fiber).

The organizational structure is loose. There are many working groups, each with a specific focus, within several areas. Each area has an area director, who collectively form the Internet Engineering Steering Group (IESG). The six permanent areas are Internet (with working groups for IPv6, DNS, and ICMP), Transport (TCP, QoS, VoIP, and SCTP), Applications (mail, some Web, LDAP), Routing (OSPF, BGP), Operations and Management (SNMP), and Security (IPSec, TLS, S/MIME). There are also two other areas: SubIP is a temporary area for things underneath the IP protocol stack (such as MPLS, IP over wireless, and traffic engineering), and there's a General area for miscellaneous and process-based working groups.

Internet Requests for Comments (RFCs) fall into three tracks: Standard, Informational, and Experimental. Note that this means that not all RFCs are standards. The RFCs in the Informational track are generally for proprietary protocols or April first jokes; those in the Experimental track are results, ideas, or theories.

The RFCs in the Standard track come from working groups in the various areas through a time-consuming, complex process. Working groups are created with an agenda, a problem statement, an email list, some draft RFCs, and a chair. They typically start out as a BoF session. The working group and the IESG make a charter to define the scope, milestones, and deadlines; the Internet Advisory Board (IAB) ensures that the working group proposals are architecturally sound. Working groups are narrowly focused and are supposed to die off once the problem is solved and all milestones achieved. Working groups meet and work mainly through the email list,

though there are three in-person high-bandwidth meetings per year. However, decisions reached in person must be ratified by the mailing list, since not everybody can get to three meetings per year. They produce RFCs which go through the Standard track; these need to go through the entire working group before being submitted for comment to the entire IETF and then to the IESG. Most RFCs wind up going back to the working group at least once from the area director or IESG level.

The format of an RFC is well-defined and requires it be published in plain 7-bit ASCII. They're freely redistributable, and the IETF reserves the right of change control on all Standard-track RFCs.

The big problems the IETF is currently facing are security, internationalization, and congestion control. Security has to be designed into protocols from the start. Internationalization has shown us that 7-bit-only ASCII is bad and doesn't work, especially for those character sets that require more than 7 bits (like Kanji); UTF-8 is a reasonable compromise. But what about domain names? While not specified as requiring 7-bit ASCII in the specifications, most DNS applications assume a 7-bit character set in the namespace. This is a hard problem. Finally, congestion control is another hard problem, since the Internet is not the same as a really big LAN.

INTRODUCTION TO AIR TRAFFIC MANAGEMENT SYSTEMS

Ron Reisman, NASA Ames Research Center; Rob Savoye, Seneca Software
Summarized by Josh Simon

Air traffic control is organized into four domains: *surface*, which runs out of the airport control tower and controls the aircraft on the ground (e.g., taxi and takeoff); *terminal area*, which covers aircraft at 11,000 feet and below, handled by the Terminal Radar Approach Control (TRACON) facilities; *en route*, which covers between 11,000 and 40,000 feet, including climb, descent, and at-

altitude flight, runs from the 20 Air Route Traffic Control Centers (ARTCC, pronounced “artsy”); and *traffic flow management*, which is the strategic arm. Each area has sectors for low, high, and very-high flight. Each sector has a controller team, including one person on the microphone, and handles between 12 and 16 aircraft at a time. Since the number of sectors and areas is limited and fixed, there’s limited system capacity. The events of September 11, 2001, gave us a respite in terms of system usage, but based on path growth patterns, the air traffic system will be oversubscribed within two to three years. How do we handle this oversubscription?

Air Traffic Management (ATM) Decision Support Tools (DST) use physics, aeronautics, heuristics (expert systems), fuzzy logic, and neural nets to help the (human) aircraft controllers route aircraft around. The rest of the talk focused on capacity issues, but the DST also handle safety and security issues. The software follows open standards (ISO, POSIX, and ANSI). The team at NASA Ames made Center-TRACON Automation System (CTAS), which is software for each of the ARTCCs, portable from Solaris to HP-UX and Linux as well. Unlike just about every other major software project, this one really is standard and portable; co-presenter Rob Savoye has experience in maintaining gcc on multiple platforms and is the project lead on the portability and standards issues for the code. CTAS allows the ARTCCs to upgrade and enhance individual aspects or parts of the system; it isn’t a monolithic all-or-nothing entity like the old ATM systems.

Some future areas of research include a head-mounted augmented reality device for tower operators, to improve their situational awareness by automating human factors; and new digital global positioning system (DGPS) technologies which are accurate within inches instead of feet.

Questions centered around advanced avionics (e.g., getting rid of ground control), cooperation between the US and Europe for software development (we’re working together on software development, but the various European countries’ controllers don’t talk well to each other), and privatization.

ADVENTURES IN DNS

Bill Manning, ISI

Summarized by Brennan Reynolds

Manning began by posing a simple question: is DNS really a critical infrastructure? The answer is not simple. Perhaps seven years ago, when the Internet was just starting to become popular, the answer was a definite no. But today, with IPv6 being implemented and so many transactions being conducted over the Internet, the question does not have a clear-cut answer. The Internet Engineering Task Force (IETF) has several new modifications to the DNS service that may be used to protect and extend its usability.

The first extension Manning discussed was Internationalized Domain Names (IDN). To date, all DNS records are based on the ASCII character set, but many addresses in the Internet’s global network cannot be easily written in ASCII characters. The goal of IDN is to provide encoding for hostnames that is fair, efficient, and allows for a smooth transition from the current scheme. The work has resulted in two encoding schemes: ACE and UTF-8. Each encoding is independent of the other, but they can be used together in various combinations. Manning expressed his opinion that while neither is an ideal solution, ACE appears to be the lesser of two evils. A major hindrance getting IDN rolled out into the Internet’s DNS root structure is the increase in zone file complexity.

Manning’s next topic was the inclusion of IPv6 records. In an IPv4 world, it is possible for administrators to remember the numeric representation of an address. IPv6 makes the addresses too

long and complex to be easily remembered. This means that DNS will play a vital role in getting IPv6 deployed. Several new resource records (RR) have been proposed to handle the translation, including AAAA, A6, DNAME and BIT-STRING. Manning commented on the difference between IPv4 and v6 as a transport protocol and that systems tuned for v4 traffic will suffer a performance hit when using v6. This is largely attributed to the increase in data transmitted per DNS request.

The final extension Manning discussed was DNSSec. He introduced this extension as a mechanism that protects the system from itself. DNSSec protects against data spoofing and provides authentication between servers. It includes a cryptographic signature of the RR set to ensure authenticity and integrity. By signing the entire set, the amount of computation is kept to a minimum. The information itself is stored in a new RR within each zone file on the DNS server.

Manning briefly commented on the use of DNS to provide a PKI infrastructure, stating that that was not the purpose of DNS and therefore it should not be used in that fashion. The signing of the RR sets can be done hierarchically, resulting in the use of a single trusted key at the root of the DNS tree to sign all sets to the leafs. However, the job of key replacement and rollover is extremely difficult for a system that is distributed across the globe.

Manning stated that in an operation test bed, with all of these extensions enabled, the packet size for a single query response grew from 518 bytes to larger than 18,000. This results in a large increase of bandwidth usage for high volume name servers and puts. Therefore, in Manning’s opinion, not all of these features will be deployed in the near future. For those looking for more information, Bill’s Web site can be found at <http://www.isi.edu/otdr>.

THE JOY OF BREAKING THINGS

Pat Parseghian, Transmeta

Summarized by Juan Navarro

Pat Parseghian described her experience in testing the Crusoe microprocessor at the Transmeta Lab for Compatibility (TLC), where the motto is, "You make it . . . we break it!" and the goal is to make engineers miserable.

She first gave an overview of Crusoe, whose most distinctive feature is a code morphing layer that translates x86 instructions into native VLIW. Such peculiarity makes the testing process particularly challenging because it involves testing two processors (the "external" x86 and the "internal" VLIW) and also because of variations on how the code-morphing layer works (it may interpret code the first few times it sees an instruction sequence and translate and save in a translation cache afterwards). There are also reproducibility issues, since the VLIW processor may run at different speeds to save energy.

Pat then described the tests that they subject the processor to (hardware compatibility tests, common applications, operating systems, envelope-pushing games, and hard-to-acquire legacy applications) and the issues that must be considered when defining testing policies. She gave some testing tips, including organizational issues like tracking resources and results.

To illustrate the testing process, Pat gave a list of possible causes of a system crash that must be investigated. If it is the silicon, then it might be because it is damaged or because of a manufacturing problem or a design flaw. If the problem is in the code-morphing layer, is the fault with the interpreter or with the translator? The fault could also be external to the processor: it could be the homemade system BIOS, a faulty motherboard, or an operator error. Or Transmeta might not be at fault at all: the crash might be due to a bug in the OS or the application. The key to pinpointing

the cause of a problem is to isolate it by identifying the conditions that reproduce the problem and repeating those conditions in other test platforms and non-Crusoe systems. Then relevant factors must be identified based on product knowledge, past experience, and common sense.

To conclude, Pat suggested that some of TLC's testing lessons can be applied to products that the audience was involved in, and assured us that breaking things is fun.

TECHNOLOGY, LIBERTY, FREEDOM, AND WASHINGTON

Alan Davidson, Center for Democracy and Technology

Summarized by Steve Bauer

"Experience should teach us to be most on our guard to protect liberty when the Government's purposes are beneficent. Men born to freedom are naturally alert to repel invasion of their liberty by evil-minded rulers. The greatest dangers to liberty lurk in insidious encroachment by men of zeal, well-meaning but without understanding." – Louis Brandeis, *Olmstead v. U.S.*

Alan Davison, an associate director at the CDT (<http://www.cdt.org>) since 1996, concluded his talk with this quote. In many ways it aptly characterizes the importance to the USENIX community of the topics he covered. The major themes of the talk were the impact of law on individual liberty, system architectures, and appropriate responses by the technical community.

The first part of the talk provided the audience with an overview of legislation either already introduced or likely to be introduced in the US Congress. This included various proposals to protect children, such as relegating all sexual content to a .xxx domain or providing kid-safe zones such as .kids.us. Other similar laws discussed were the Children's Online Protection Act and legislation dealing with virtual child pornography.

Other lower-profile pieces covered were laws prohibiting false contact information in emails and domain registration databases. Similar proposals exist that would prohibit misleading subject lines in emails and require messages to clearly identify if they are advertisements. Briefly mentioned were laws impacting online gambling.

Bills and laws affecting the architectural design of systems and networks and various efforts to establish boundaries and borders in cyberspace were then discussed. These included the Consumer Broadband and Television Promotion Act and the French cases against Yahoo and its CEO for allowing the sale of Nazi materials on the Yahoo auction site.

A major part of the talk focused on the technological aspects of the USA-Patriot Act passed in the wake of the September 11 attacks. Topics included "pen registers" for the Internet, expanded government power to conduct secret searches, roving wiretaps, nationwide service of warrants, sharing grand jury and intelligence information, and the establishment of an identification system for visitors to the US.

The technological community needs to be aware and care about the impact of law on individual liberty and the systems the community builds. Specific suggestions included designing privacy and anonymity into architectures and limiting information collection, particularly any personally identifiable data, to only what is essential. Finally, Alan emphasized that many people simply do not understand the implications of law. Thus the technology community has a important role in helping make these implications clear.

TAKING AN OPEN SOURCE PROJECT TO MARKET

Eric Allman, Sendmail Inc.

Summarized by Scott Kilroy

Eric started the talk with the upsides and downsides to open source software. The upsides include rapid feedback, high-

quality work (mostly), lots of hands, and an intensely engineering-driven approach. The downsides include no support structure, limited project marketing expertise, and volunteers having limited time.

In 1998 Sendmail was becoming a success disaster. A success disaster includes two key elements: (1) volunteers start spending all their time supporting current functionality instead of enhancing (this heavy burden can lead to project stagnation and, eventually, death); (2) competition for the better-funded sources can lead to FUD (fear, uncertainty, and doubt) about the project.

Eric then led listeners down the path he took to turn Sendmail into a business. Eric envisioned Sendmail Inc. as a smallish and close-knit company but soon realized that the family atmosphere he desired could not last as the company grew. He observed that maybe only the first 10 people will buy into your vision, after which each individual is primarily interested in something else (e.g., power, wealth, or status). He warns that there will be people working for you that you do not like. Eric particularly did not care for sales people, but he emphasized how important sales, marketing, finance, and legal people are to companies.

The nature of companies in infancy can be summed up as: you need money, and you need investors in order to get money. Investors want a return on investment, so money management becomes critical. Companies therefore must optimize money functions. Eric's experience with investors were lessons all in themselves. More than giving money, good investors can provide connections and insight, so you don't want investors who don't believe in you.

A company must have bug history, change management, and people who understand the product and have a sense of the target market. Eric now sees the importance of marketing target research. A company can never forget

the simple equation that drives business: $\text{profit} = \text{revenue} - \text{expense}$.

Finally, the concept of value should be based on what is valuable to the customer. Eric observed that his customers needed documentation, extra value in the product that they couldn't get elsewhere, and technical support.

Eric learned hard lessons along the way. If you want to do interesting open source, it might be best not to be too successful. You don't want to let the larger dragons (companies) notice you. If you want a commercial user base, you have to manage process, watch corporate culture, provide value to customers, watch bottom lines, and develop a thick skin.

Starting a company is easy but perpetuating it is extremely difficult!

INFORMATION VISUALIZATION FOR SYSTEMS PEOPLE

Tamara Munzner, University of British Columbia

Summarized by J.D. Welch

Munzner presented interesting evidence that information visualization through interactive representations of data can help people to perform tasks more effectively and reduce the load on working memory.

A popular technique in visualizing data is abstracting it through node-link representation – for example, a list of cross-references in a book. Representing the relationships between references in a graphical (node-link) manner “offloads cognition to the human perceptual system.” These graphs can be produced manually, but automated drawing allows for increased complexity and quicker production times.

The way in which people interpret visual data is important in designing effective visualization systems. Attention to pre-attentive visual cues are key to success. Picking out a red dot among a field of identically shaped blue dots is significantly faster than the same data repre-

sented in a mixture of hue and shape variations. There are many pre-attentive visual cues, including size, orientation, intersection, and intensity. The accuracy of these cues can be ranked, with position triggering high accuracy and color or density on the low end.

Visual cues combined with different data types – quantitative (e.g., 10 inches), ordered (small, medium, large) and categorical (apples, oranges) – can also be ranked, with spatial position being best for all types. Beyond this commonality, however, accuracy varied widely, with length ranked second for quantitative data, eighth for ordinal, and ninth for categorical data types.

These guidelines about visual perception can be applied to information visualization, which, unlike scientific visualization, focuses on abstract data and choice of specialization. Several techniques are used to clarify abstract data, including multi-part glyphs, where changes in individual parts are incorporated into an easier to understand gestalt, interactivity, motion, and animation. In large data sets, techniques like “focus + context,” where a zoomed portion of the graph is shown along with a thumbnail view of the entire graph, are used to minimize user disorientation.

Future problems include dealing with huge databases, such as the Human Genome, reckoning dynamic data, like the changing structure of the Web or real-time network monitoring, and transforming “pixel bound” displays into large “digital wallpaper”-type systems.

FIXING NETWORK SECURITY BY HACKING THE BUSINESS CLIMATE

Bruce Schneier, Counterpane Internet Security

Summarized by Florian Buchholz

Bruce Schneier identified security as one of the fundamental building blocks of the Internet. A certain degree of security is needed for all things on the Net, and the limits of security will become limits of the Net itself. However, companies are

hesitant to employ security measures. Science is doing well, but one cannot see the benefits in the real world. An increasing number of users means more problems affecting more people. Old problems such as buffer overflows haven't gone away, and new problems show up. Furthermore, the amount of expertise needed to launch attacks is decreasing.

Schneier argues that complexity is the enemy of security and that while security is getting better, the growth in complexity is outpacing it. As security is fundamentally a people problem, one shouldn't focus on technologies but, rather, should look at businesses, business motivations, and business costs. Traditionally, one can distinguish between two security models: threat avoidance, where security is absolute, and risk management, where security is relative and one has to mitigate the risk with technologies and procedures. In the latter model, one has to find a point with reasonable security at an acceptable cost.

After a brief discussion on how security is handled in businesses today, in which he concluded that businesses "talk big about it, but do as little as possible," Schneier identified four necessary steps to fix the problems.

First, enforce liabilities. Today, no real consequences have to be feared from security incidents. Holding people accountable will increase the transparency of security and give an incentive to make processes public. As possible options to achieve this, he listed industry-defined standards, federal regulation, and lawsuits. The problems with enforcement, however, lie in difficulties associated with the international nature of the problem and the fact that complexity makes assigning liabilities difficult. Furthermore, fear of liability could have a chilling effect on free software development and could stifle new companies.

As a second step, Schneier identified the need to allow partners to transfer liability. Insurance would spread liability risk among a group and would be a CEO's primary risk analysis tool. There is a need for standardized risk models and protection profiles and for more securities as opposed to empty press releases. Schneier predicts that insurance will create a marketplace for security where customers and vendors will have the ability to accept liabilities from each other. Computer-security insurance should soon be as common as household or fire insurance, and from that development, insurance will become the driving factor of the security business.

The next step is to provide mechanisms to reduce risks, both before and after software is released; techniques and processes to improve software quality, as well as an evolution of security management, are therefore needed. Currently, security products try to rebuild the walls – such as physical badges and entry doorways – that were lost when getting connected. Schneier believes this "fortress metaphor" is bad; one should think of the problem more in terms of a city. Since most businesses cannot afford proper security, outsourcing is the only way to make security scalable. With outsourcing, the concept of best practices becomes important and insurance companies can be tied to them; outsourcing will level the playing field.

As a final step, Schneier predicts that rational prosecution and education will lead to deterrence. He claims that people feel safe because they live in a lawful society, whereas the Internet is classified as lawless, very much like the American West in the 1800s or a society ruled by warlords. This is because it is difficult to prove who an attacker is; prosecution is hampered by complicated evidence gathering and irrational prosecution. Schneier believes, however, that education will play a major role in turning the Internet into a lawful society. Specifically, he pointed out that we need laws that can be explained.

Risks won't go away; the best we can do is manage them. A company able to manage risk better will be more profitable, and we need to give CEOs the necessary risk-management tools.

There were numerous questions. Listed below are the more complicated ones in a paraphrased Q&A format:

Q: Will liability be effective; will insurance companies be willing to accept the risks? A: The government might have to step in; it needs to be seen how it plays out.

Q: Is there an analogy to the real world in the fact that in a lawless society only the rich can afford security? Security solutions differentiate between classes. A: Schneier disagreed with the statement, giving an analogy to front-door locks, but conceded that there might be special cases.

Q: Doesn't homogeneity hurt security? A: Homogeneity is oversold. Diverse types can be more survivable, but given the limited number of options, the difference will be negligible.

Q: Regulations in airbag protection have led to deaths in some cases. How can we keep the pendulum from swinging the other way? A: Lobbying will not be prevented. An imperfect solution is probable; there might be reversals of requirements such as the airbag one.

Q: What about personal liability? A: This will be analogous to auto insurance: liability comes with computer/Net access.

Q: If the rule of law is to become reality, there must be a law enforcement function that applies to a physical space. You cannot do that with any existing government agency for the whole world. Should an organization like the UN assume that role? A: Schneier was not convinced global enforcement is possible.

Q: What advice should we take away from this talk? A: Liability is coming. Since the network is important to our infrastructure, eventually the problems

will be solved in a legal environment. You need to start thinking about how to solve the problems and how the solutions will affect us.

The entire speech (including slides) is available at <http://www.counterpane.com/presentation4.pdf>.

LIFE IN AN OPEN SOURCE STARTUP

Daryll Strauss, Consultant

Summarized by Teri Lampoudi

This talk was packed with morsels of insight and tidbits of information about what life in an open source startup is like (though “was like” might be more appropriate); what the issues are in starting, maintaining, and commercializing an open source project; and the way hardware vendors treat open source developers. Strauss began by tracing the timeline of his involvement with developing 3-D graphics support for Linux: from the 3dfx voodoo1 driver for Linux in 1997, to the establishment of Precision Insight in 1998, to its buyout by VA Linux, to the dismantling of his group by VA, to what the future may hold.

Obviously, there are benefits from doing open source development, both for the developer and for the end user. An important point was that open source develops entire technologies, not just products. A misapprehension is the inherent difficulty of managing a project that accepts code from a large number of developers, some volunteer and some paid. The notion that code can just be thrown over the wall into the world is a Big Lie.

How does one start and keep afloat an open source development company? In the case of Precision Insight, the subject matter – developing graphics and OpenGL for Linux – required a considerable amount of expertise: intimate knowledge of the hardware, the libraries, X and the Linux kernel, as well as the end applications. Expertise is marketable. What helps even further is having a *visible* virtual team of experts, people who have an established track record of contributions to open source

in the particular area of expertise. Support from the hardware and software vendors came next. While everyone was willing to contract PI to write the portions of code that were specific to their hardware, no one wanted to pay the bill for developing the underlying foundation that was necessary for the drivers to be useful. In the PI model, the infrastructure cost was shared by several customers at once, and the technology kept moving. Once a driver is written for a vendor, the vendor is perfectly capable of figuring out how to write the next driver necessary, thereby obviating the need to contract the job. This and questions of protecting intellectual property – hardware design in this case – are deeply problematic with the open source mode of development. This is not to say that there is no way to get over them, but that they are likely to arise more often than not.

Management issues seem equally important. In the PI setting, contracts were flexible – both a good and a bad thing – and developers overworked. Further, development “in a fishbowl,” that is, under public scrutiny, is not easy. Strauss stressed the value of good communication and the use of various out-of-sync communication methods, like IRC and mailing lists.

Finally, Strauss discussed what portions of software can be free and what can be proprietary. His suggestion was that horizontal markets want to be free, whereas vertically, one can develop proprietary solutions. The talk closed with a small stroll through the problems that project politics brings up, from things like merging branches to mailing list and source tree readability.

GENERAL TRACK SESSIONS

FILE SYSTEMS

Summarized by Haijin Yan

STRUCTURE AND PERFORMANCE OF THE DIRECT ACCESS FILE SYSTEM

Kostas Magoutis, Salimah Addetia, Alexandra Fedorova, and Margo I. Seltzer, Harvard University; Jeffrey Chase, Andrew Gallatin, Richard Kisley, and Rajiv Wickremesinghe, Duke University; and Eran Gabber, Lucent Technologies

This paper won the Best Paper award.

The Direct Access File System (DAFS) is a new standard for network-attached storage over direct-access transport networks. DAFS takes advantage of user-level network interface standards that enable client-side functionality in which remote data access resides in a library rather than in the kernel. This reduces the overhead of memory copy for data movement and protocol overhead. Remote Direct Memory Access (RDMA) is a direct access transport network that allows the network adapter to reduce copy overhead by accessing application buffers directly.

This paper explores the fundamental structural and performance characteristics of network file access using a user-level file system structure on a direct-access transport network with RDMA. It describes DAFS-based client and server reference implementations for FreeBSD and reports experimental results, comparing DAFS to a zero-copy NFS implementation. It illustrates the benefits and trade-offs of these techniques to provide a basis for informed choices about deployment of DAFS-based systems and similar extensions to other network file protocols such as NFS. Experiments show that DAFS gives applications direct control over an I/O system and increases the client CPU's usage while the client is doing I/O. Future work includes how to address longstanding problems related to the integration of the application and file system for high-performance applications.

CONQUEST: BETTER PERFORMANCE THROUGH A DISK/PERSISTENT-RAM HYBRID FILE SYSTEM

An-I A. Wang, Peter Reiher, and Gerald J. Popek, UCLA; Geoffrey M. Kuening, Harvey Mudd College

Motivated by the declining cost of persistent RAM, the authors propose the Conquest file system, which stores all small files, metadata, executables, and shared libraries in persistent RAM; disks hold only the data content of remaining large files. Compared to alternatives such as caching and RAM file systems, Conquest has the advantages of efficiency, consistency, and reliability at a reduced cost. Using popular benchmarks, experiments show that Conquest incurs little overhead while achieving faster performance. Future work includes designing mechanisms for adjusting file-size threshold dynamically and finding a better disk layout for large data blocks.

EXPLOITING GRAY-BOX KNOWLEDGE OF BUFFER-CACHE MANAGEMENT

Nathan C. Burnett, John Bent, Andrea C. Arpaci-Dusseau, and Remzi H. Arpaci-Dusseau, University of Wisconsin, Madison

Knowing what algorithm is used to manage the buffer cache is very important for improving application performance. However, there is currently no interface for finding this algorithm. This paper introduces Dust, a simple fingerprinting tool that is able to identify the buffer-cache replacement policy. Dust automatically identifies the cache size and replacement policy based on the configuring attributes of access orders, recency, frequency, and long-term history. Through simulation, Dust was able to distinguish between a variety of replacement algorithm policies found in the literature: FIFO, LRU, LFU, Clock, Segmented FIFO, 2Q, and LRU-K. Further experiments of fingerprinting real operating system such as NetBSD, Linux, and Solaris show that Dust is able to identify the hidden cache replacement algorithm.

By knowing the underlying cache replacement policy, a cache-aware Web server can reschedule the requests on a cached request-first policy to obtain performance improvement. Experiments show that cache-aware scheduling improves average response time and system throughput.

OPERATING SYSTEMS (AND DANCING BEARS)

Summarized by Matt Butner

THE JX OPERATING SYSTEM

Michael Golm, Meik Felser, Christian Wawersich, and Jürgen Kleinöder, University of Erlangen-Nürnberg

The talk opened by emphasizing the need for, and the practicality of, a functional Java OS. Such implementation in the JX OS attempts to mimic the recent trend toward using highly abstracted languages in application development in order to create an OS as functional and powerful as one written in a lower-level language.

The JX is a micro-kernel solution that uses separate JVMs for each entity of the kernel, and, in some cases, for each application. The separated domains do not share objects and have no thread migration, and each domain implements its own code. The interesting part of the presentation was the discussion of system-level programming with Java; key areas discussed were memory management and interrupt handlers. The authors concluded by noting that their type-safe and modular system resulted in a robust system with great configuration flexibility and acceptable performance.

DESIGN EVOLUTION OF THE EROS SINGLE-LEVEL STORE

Jonathan S. Shapiro, Johns Hopkins University; Jonathan Adams, University of Pennsylvania

This presentation outlined current characteristics of file systems, and some of their least desirable characteristics. It was based on the revival of the EROS

single-level-store approach for current attempts to capitalize on system consistency and efficiency. Their solution did not relate directly to EROS but, rather, to the design and use of the constructs and notions on which EROS is based. By extending the mapping of the memory system to include the disk, systems are further able to ensure global persistence without regard for a disk structure. In explaining the need for absolute system consistency and an environment which by definition is not allowed to crash, the goal of having such an exhaustive design becomes clear. The cool part of this work is the availability of its design and architecture to the public.

THINK: A SOFTWARE FRAMEWORK FOR COMPONENT-BASED OPERATING SYSTEM KERNELS

Jean-Philippe Fassino, France Télécom R&D; Jean-Bernard Stefani, INRIA; Julia Lawall, DIKU; Gilles Muller, INRIA

This presentation discussed the need for component-based operating systems and respective structures to ensure flexibility for arbitrary-sized systems. Think provides a binding model that maps uniformed components for OS developers and architects to follow, ensuring consistent implementations for an arbitrary system size. However, the goal is not to force developers into a predefined kernel but to promote the use of certain components in varied ways.

Think's concentration is primarily on embedded systems, where short development time is necessary but is constrained by rigorous needs and limited resources. The need to build flexible systems in such an environment can be costly and implementation-specific, but Think creates an environment supported by the ability to dynamically load type-safe components. This allows for more flexible systems that retain functionality because of Think's ability to bind fine-grained components. Benchmarks revealed that dedicated micro-kernels can show performance improvements in comparison to mono-

lithic kernels. Notable future work includes building components for low-end appliances and the development of a real-time OS component library.

BUILDING SERVICES

Summarized by Pradipta De

NINJA: A FRAMEWORK FOR NETWORK SERVICES

J. Robert von Behren, Eric A. Brewer, Nikita Borisov, Michael Chen, Matt Welsh, Josh MacDonald, Jeremy Lau, and David Culler, University of California, Berkeley

Robert von Behren presented Ninja, an ongoing project that aims to provide a framework for building robust and scalable Internet-based network services, like Web-hosting, instant-messaging, email, and file-sharing applications. Robert drew attention to the difficulties of writing cluster applications. One has to take care of data consistency and issues of concurrency, and for robust applications there are problems related to fault tolerance. Ninja works as a wrapper to relieve the user of these problems. The goal of the project is building network services that are scalable and highly available; maintaining persistent data; providing graceful degradation; and supporting online evolution.

The use of clusters distinguishes this setup from generic distributed systems in terms of reliability and security, as well as providing a partition-free network. The next important feature of this project is the new programming model, which is more restrictive than a general programming model but still expressive enough to write most of the applications. This model, described as “single program multiple connection,” uses intertask parallelism instead of multi-threaded concurrency and is characterized by all nodes running the same program, with connections delegated to the nodes by a centralized connection manager (CM). A CM takes care of hiding the details of mapping an external connection to an internal node. Ninja

also uses a design called “staged event-driven architecture,” where each service is broken down into a set of stages connected by event queues. This architecture is suitable for a modular design and helps in graceful degradation by adaptive load shedding from the event queues.

The presentation concluded with examples of implementation and evaluation of a Web server and an email system called NinjaMail, to show the ease of authoring and the efficacy of using the Ninja framework for service development.

USING COHORT-SCHEDULING TO ENHANCE SERVER PERFORMANCE

James R. Larus and Michael Parkes, Microsoft Research

James Larus presented a new scheduling policy for increasing the throughput of server applications. Cohort scheduling batches execution of similar operations arising in different server requests. The usual programming paradigm in handling server requests is to spawn multiple concurrent threads and switch from one thread to another whenever a thread gets blocked for I/O. Since the threads in a server mostly execute unrelated pieces of code, the locality of reference is reduced; hence the effectiveness of different caching mechanisms. One way to solve this problem is to throw in more hardware. But Larus presented a complementary view where the program behavior is investigated and used to improve the performance.

The problem in this scheme is to identify pieces of code that can be batched together for processing. One simple way is to look at the next program counter values and use them to club threads together. However, this talk presented a new programming abstraction, “staged computation,” which replaces the thread model with “stages.” A stage is an abstraction for operations with similar behavior and locality. The StagedServer library can be used for programming in this model. It is a collection of C++

classes that implement staged computation and cohort scheduling on either a uniprocessor or multiprocessor. It can be used to define stages.

The presentation showed the experimental evaluation of the cohort scheduling over the thread-based model by implementing two servers: a Web server, which is mainly I/O bound, and a publish-subscribe server, which is mainly compute bound. The SURGE benchmark was used for the first experiment and the Fabret workload for the second. The results showed that cohort-scheduling-based implementation gave a better throughput than the thread-based implementation at high loads.

NETWORK PERFORMANCE

Summarized by Xiaobo Fan

ETE: PASSIVE END-TO-END INTERNET SERVICE PERFORMANCE MONITORING

Yun Fu, Amin Vahdat, Duke University; Ludmila Cherkasova, Wenting Tang, HP Labs

This paper won the Best Student Paper award.

Ludmila Cherkasova began by listing several questions most Web service providers want answered in order to improve service quality. She reviewed the difficulties of making accurate and efficient end-to-end Web service measurement and the shortcomings of currently available approaches. They propose a passive trace-based architecture, called EtE, to monitor Web server performance on behalf of end users.

The first step is to collect network packets passively. The second module reconstructs all TCP connections and extracts HTTP transactions. To reconstruct Web page accesses, they first build a knowledge base indexed by client IP and URL and then group objects to the related Web pages they are embedded in. Statistical analysis is used to handle non-matched objects. EtE Monitor can generate three groups of metrics to measure Web service performance: response time, Web caching, and page abortion.

To demonstrate the benefits of EtE monitor, Cherkasova talked about two case studies and, based on the calculated metrics, gave some insightful explanations about what's happening behind the variations of Web performance. Through validation they claim their approach provides a very close approximation to the real scenario.

THE PERFORMANCE OF REMOTE DISPLAY MECHANISMS FOR THIN-CLIENT COMPUTING

S. Jae Yang, Jason Nieh, Matt Selsky, Nikhil Tiwari, Columbia University
Noting the trend toward thin-client computing, the authors compared different techniques and design choices in measuring the performance of six popular thin-client platforms – Citrix MetaFrame, Microsoft Terminal Services, Sun Ray, Tarantella, VNC, and X. After pointing out several challenges in benchmarking thin clients, Yang proposed slow-motion benchmarking to achieve non-invasive packet monitoring and consistent visual quality. Basically, they insert delays between separate visual events in the benchmark applications of the server side so that the client's display update can catch up with the server's processing speed. The experiments are conducted on an emulated network over a range of network bandwidths.

Their results show that thin clients can provide good performance for Web applications in LAN environments, but only some platforms performed well for video benchmark. Pixel-based encoding may achieve better performance and bandwidth efficiency than high-level graphics. Display caching and compression should be used with care.

A MECHANISM FOR TCP-FRIENDLY TRANSPORT-LEVEL PROTOCOL COORDINATION

David E. Ott and Ketan Mayer-Patel, University of North Carolina, Chapel Hill

A revised transport-level protocol optimized for cluster-to-cluster (C-to-C)

applications is what David Ott tries to explain in his talk. A C-to-C application class is identified as one set of processes communicating to another set of processes across a common Internet path. The fundamental problem with C-to-C applications is how to coordinate all C-to-C communication flows so that they share a consistent view of the common C-to-C network, adapt to changing network conditions, and cooperate to meet specific requirements. Aggregation points (AP) are placed at the first and last hop of the common data path to probe network conditions (latency, bandwidth, loss rate, etc.). To carry and transfer this information, a new protocol – Coordination Protocol (CP) – is inserted between the network layer (IP) and the transport layer (TCP, UCP, etc.). Ott illustrated how the CP header is updated and used when packets originate from source and traverse through local and remote AP to arrive at their destination, and how AP maintains a per-cluster state table and detects network conditions. Through simulation results, this coordination mechanism appears effective in sharing common network resources among C-to-C communication flows.

STORAGE SYSTEMS

Summarized by Praveen Yalagandula

MY CACHE OR YOURS? MAKING STORAGE MORE EXCLUSIVE

Theodore M. Wong, Carnegie Mellon University; John Wilkes, HP Labs

Theodore Wong explained the inefficiency of current “inclusive” caching schemes in storage area networks – when a client accesses a block, the block is read from the disk and is cached at both the disk array cache and at the client's cache. Then he presented the concept of “exclusive” caching, where a block accessed by a client is only cached in that client's cache. On eviction from the client's cache, they have come up with a DEMOTE operation to move the data block to the tail of the array cache.

The new exclusion caching schemes are evaluated on both single-client and multiple-client systems. The single-client results are presented for two different types of synthetic workloads: Random (transaction-processing type workloads) and Zipf (typical Web workloads). The exclusive policy was quite effective in achieving higher hit rates and lower read latencies. They also showed a 2.2 times hit-rate improvement over inclusive techniques in the case of a real-life workload, `httpd`, with a single-client setting.

The DEMOTE scheme also performed well in the case of multiple-client systems when the data accessed by clients is disjointed. In the case of shared data workloads, this scheme performed worse than the inclusive schemes. Theodore then presented an adaptive exclusive caching scheme – a block accessed by a client is placed at the tail in the client's cache and also in the array cache at an appropriate place determined by the popularity of the block. The popularity of the block is measured by maintaining a ghost cache to accumulate the number of times each block is accessed. This new adaptive scheme has achieved a maximum of 52% speedup in the mean latency in the experiments with real-life workloads.

For more information, visit <http://www.cs.cmu.edu/~tmwong/research> and <http://www.hpl.hp.com/research/itc/csl/ssp>.

BRIDGING THE INFORMATION GAP IN STORAGE PROTOCOL STACKS

Timothy E. Denehy, Andrea C. Arpaci-Dusseau, Remzi H. Arpaci-Dusseau, University of Wisconsin, Madison

Currently there is a huge information gap between storage systems and file systems. The interface exposed by storage systems to file systems, based on blocks and providing only read/write interfaces, is very narrow. This leads to poor performance as a whole because of duplicated functionality in both systems and

reduced functionality resulting from storage systems' lack of file information.

The speaker presented two enhancements: ExRaid, an exposed RAID, and I.LFS, Informed Log-Structured File System. ExRAID is an enhancement to the block-based RAID storage system that exposes the following three types of information to the file system: (1) regions – contiguous portions of the address space comprising one or multiple disks; (2) performance information about queue lengths and throughput of the regions revealing disk heterogeneity to the file systems; and (3) failure information – dynamically updated information conveying the number of tolerable failures in each region.

I.LFS allows online incremental expansion of the storage space, performs dynamic parallelism using ExRAID's performance information to perform well on heterogeneous storage systems, and provides a range of different mechanisms with different granularities for redundancy of files using ExRAID's region failure characteristics. These new features are added to LFS with only a 19% increase in the code size.

More information:

<http://www.cs.wisc.edu/wind>.

MAXIMIZING THROUGHPUT IN REPLICATED DISK STRIPING OF VARIABLE BIT-RATE STREAMS

Stergios V. Anastasiadis, Duke University; Kenneth C. Sevcik and Michael Stumm, University of Toronto

There is an increasing demand for the continuous real-time streaming of media files. Disk striping is a common technique used for supporting many connections on the media servers. But even with 1.2 million hours of mean time between failures, there will be more than one disk failure per week with 7000 disks. Fault tolerance can be achieved by using data replication and reserving extra bandwidth during normal operation. This work focused on supporting the most common variable bit-rate stream formats (e.g., MPEG).

The authors used prototype media server EXEDRA to implement and evaluate different replication and bandwidth reservation schemes. This system supports a variable bit-rate scheme, does stride-based disk allocation, and is capable of supporting variable-grain disk striping. Two replication policies are presented: deterministic – data blocks are replicated in a round-robin fashion on the secondary replicas; random – data blocks are replicated on the randomly chosen secondary replicas. Two bandwidth reservation techniques are also presented: mirroring reservation – disk bandwidth is reserved for both primary and replicas of the media file during the playback; and minimum reservation – a more efficient scheme in which bandwidth is reserved only for the sum of primary data access time and the maximum of the backup data access times required in each round.

Experimental results showed that deterministic replica placement is better than random placement for small disks, minimum disk bandwidth reservation is twice as good as mirroring in the throughput achieved, and fault tolerance can be achieved with a minimal impact on the throughput.

TOOLS

Summarized by Li Xiao

SIMPLE AND GENERAL STATISTICAL PROFILING WITH PCT

Charles Blake and Steven Bauer, MIT
This talk introduced a Profile Collection Toolkit (PCT) – a sampling-based CPU profiling facility. A novel aspect of PCT is that it allows sampling of semantically rich data such as function call stacks, function parameters, local or global variables, CPU registers, or other execution contexts. The design objectives of PCT were driven by user needs and the inadequacies or inaccessibility of prior systems.

The rich data collection capability is achieved via a debugger-controller program, dbct1. The talk then introduced the profiling collector and data collec-

tion activation. PCT file format options provide flexibility and various ways to store exact states. PCT also has analysis options; two examples were given – “Simple” and “Mixed User and Linux Code.”

The talk then went into how general sampling helps in the following areas: a debugger-controller state machine; example script fragments; a simple example program; and general value reports in terms of call-path histograms, numeric value histograms, and data generality. Related work, such as gprof and Expect, was then summarized. The contribution of this work is to provide a portable general value sampling tool. The limitations are that PCT does not support much of strip, and count inaccuracies happen because of its statistical nature. Based on this limitation, -g is preferred over strip.

Possible future directions included supporting more debuggers, such as dbx and xdb; script generators for other kinds of traces; more canned reports for general values; and a libgdb-based library sampler. PCT is available at <http://pdos.lcs.mit.edu/pct>. PCT runs on almost any UNIX-like system.

ENGINEERING A DIFFERENCING AND COMPRESSION DATA FORMAT

David G. Korn and Kiem Phong Vo, AT&T Laboratories – Research

This talk began with an equation: “Differencing + Compression = Delta Compression.” Compression removes information redundancy in a data set. Examples are gzip, bzip, compress, and pack. Differencing encodes differences between two data sets. Examples are diff -e, fdelta, bdiff, and xdelta. Delta compression compresses a data set given another, combining differencing and compression, and reduces to pure compression when there's no commonality.

After presenting an overall scheme for a delta compressor and showing delta compression performance, this talk discussed the encoding format of a newly designed Vcdiff for delta compression. A

Vcdiff instruction code table consists of 256 entries of each coding up to a pair of instructions, and recodes I-byte indices and any additional data.

The talk then showed Vcdiff's performance with Web data collected from CNN, compared with W3C standard Gdiff, Gdiff+gzip, and diff+gzip, where Gdiff was computed using Vcdiff delta instructions. The results of two experiments, "First" and "Successive," were presented. In "First," each file is compressed against the first file collected; in "Successive," each file is compressed against the one in the previous hour. The diff+gzip did not work well because diff was line-oriented. Vcdiff performed favorably compared with other formats.

Vcdiff is part of the Vcodex package; Vcodex is a platform for all common data transformations, delta compression, plain compression, encryption, transcoding (e.g., uuencode, base64). It is structured in three layers for maximum usability. Base library uses Disciplines and Methods interfaces.

The code for Vcdiff can be found at <http://www.research.att.com/sw/tools>. They are moving Vcdiff to the IETF standard as a comprehensive platform for transforming data. Please refer to <http://www.ietf.org/internet-draft/draft-korn-vcdiff.06.txt>.

WHERE IN THE NET . . .

Summarized by Amit Purohit

A PRECISE AND EFFICIENT EVALUATION OF THE PROXIMITY BETWEEN WEB CLIENTS AND THEIR LOCAL DNS SERVERS

Zhuoqing Morley Mao, University of California, Berkeley; Charles D. Cranor, Fred Dougliis, Michael Rabinovich, Oliver Spatscheck, and Jia Wang, AT&T Labs – Research

Content Distribution Networks (CDNs) attempt to improve Web performance by delivering Web content to end users from servers located at the edge of the network. When a Web client requests content, the CDN dynamically chooses a server to route the request to, usually the

one that is nearest the client. As CDNs have access only to the IP address of the local DNS server (LDNS) of the client, the CDN's authoritative DNS server maps the client's LDNS to a geographic region within a particular network and combines that with network and server load information to perform CDN server selection.

This method has two limitations. First, it is based on the implicit assumption that clients are close to their LDNS. This may not always be valid. Second, a single request from an LDNS can represent a differing number of Web clients – called the hidden load factor. Knowledge of the hidden load factor can be used to achieve better load distribution.

The extent of the first limitation and its impact on the CDN server selection is dealt with. To determine the associations between clients and their LDNS, a simple, non-intrusive, and efficient mapping technique was developed. The data collected was used to study the impact of proximity on DNS-based server selection using four different proximity metrics: (1) autonomous system (AS) clustering – observing whether a client is in the same AS as its LDNS – concluded that LDNS is very good for coarse-grained server selection, as 64% of the associations belong to the same AS; (2) network clustering – observing whether a client is in the same network-aware cluster (NAC) – implied that DNS is less useful for finer-grained server selection, since only 16% of the client and LDNS are in the same NAC; (3) traceroute divergence – the length of the divergent paths to the client and its LDNS from a probe point using traceroute – implies that most clients are topologically close to their LDNS as viewed from a randomly chosen probe site; (4) round-trip time (RTT) correlation (some CDNs select servers based on RTT between the CDN server and the client's LDNS) examines the correlation between the message RTTs from a probe point to the client and its local DNS;

results imply this is a reasonable metric to use to avoid really distant servers.

A study of the impact that client-LDNS associations have on DNS-based server selection concludes that knowing the client's IP address would allow more accurate server selection in a large number of cases. The optimality of the server selection also depends on the server density, placement, and selection algorithms.

Further information can be found at <http://www.eecs.berkeley.edu/~zmao/myresearch.html> or by contacting zmao@eecs.berkeley.edu.

GEOGRAPHIC PROPERTIES OF INTERNET ROUTING

Lakshminarayanan Subramanian, Venkata N. Padmanabhan, Microsoft Research; Randy H. Katz, University of California at Berkeley

Geographic information can provide insights into the structure and functioning of the Internet, including interactions between different autonomous systems, by analyzing certain properties of Internet routing. It can be used to measure and quantify certain routing properties such as circuitous routing, hot-potato routing, and geographic fault tolerance.

Traceroute has been used to gather the required data, and Geotrack tool has been used to determine the location of the nodes along each network path. This enables the computation of "linearized distances," which is the sum of the geographic distances between successive pairs of routers along the path.

In order to measure the circuitousness of a path, a metric "distance ratio" has been defined as the ratio of the linearized distance of a path to the geographic distance between the source and destination of the path. From the data, it has been observed that the circuitousness of a route depends on both geographic and network location of the end hosts. A large value of the distance ratio enables us to flag paths that are highly

circuitous, possibly (though not necessarily) because of routing anomalies. It is also shown that the minimum delay between end hosts is strongly correlated with the linearized distance of the path.

Geographic information can be used to study various aspects of wide-area Internet paths that traverse multiple ISPs. It was found that end-to-end Internet paths tend to be more circuitous than intra-ISP paths, the cause for this being the peering relationships between ISPs. Also, paths traversing substantial distances within two or more ISPs tend to be more circuitous than paths largely traversing only a single ISP. Another finding is that ISPs generally employ hot-potato routing.

Geographic information can also be used to capture the fact that two seemingly unrelated routers can be susceptible to correlated failures. By using the geographic information of routers we can construct a geographic topology of an ISP. Using this we can find the tolerance of an ISP's network to the total network failure in a geographic region.

For further information, contact lakme@cs.berkeley.edu.

PROVIDING PROCESS ORIGIN INFORMATION TO AID IN NETWORK TRACEBACK

Florian P. Buchholz, Purdue University; Clay Shields, Georgetown University
Network traceback is currently limited because host audit systems do not maintain enough information to match incoming network traffic to outgoing network traffic. The talk presented an alternative, assigning origin information to every process and logging it during interactive login creation.

The current implementation concentrates mainly on interactive sessions in which an event is logged when a new connection is established using SSH or Telnet. The method is effective and could successfully determine stepping stones and reliably detect the source of a DDoS attack. The speaker then talked about the related work done in this area,

which led to discussion of the latest development in the area of "host causality."

The speaker ended with the limitations and future directions of the framework. This framework could be extended and could find many applications in the area of security. Current implementation doesn't take care of startup scripts and cron jobs, but incorporating the origin information in FS could solve this problem. In the current implementation, logging is just implemented as a proof of concept. It could be made safe in many ways, and this could be another important aspect of future work.

PROGRAMMING

Summarized by Amit Purohit

CYCLONE : A SAFE DIALECT OF C

Trevor Jim, AT&T Labs – Research; Greg Morrisett, Dan Grossman, Michael Hicks, James Cheney, Yanling Wang, Cornell University

Cyclone is designed to provide protection against attacks such as buffer overflows, format string attacks, and memory management errors. The current structure of C allows programmers to write vulnerable programs. Cyclone extends C so that it has the safety guarantee of Java while keeping C syntax, types, and semantics untouched.

The Cyclone compiler performs static analysis of the source code and inserts runtime checks into the compiled output at places where the analysis cannot determine that the code execution will not violate safety constraints. Cyclone imposes many restrictions to preserve safety, such as NULL checks. These checks do not exist in normal C.

The speaker then talked about some sample code written in Cyclone and how it tackles safety problems. Porting an existing C application to Cyclone is pretty easy, with fewer than 10% change required. The current implementation concentrates more on safety than on performance – hence, the performance

penalty is substantial. Cyclone was able to find many lingering bugs. The final step of the project will be to write a compiler to convert a normal C program to an equivalent Cyclone program.

COOPERATIVE TASK MANAGEMENT WITHOUT MANUAL STACK MANAGEMENT

Atul Adya, Jon Howell, Marvin Theimer, William J. Bolosky, John R. Douceur, Microsoft Research

The speaker described the definitions and motivations behind the distinct concepts of task management, stack management, I/O management, conflict management, and data partitioning. Conventional concurrent programming uses preemptive task management and exploits the automatic stack management of a standard language. In the second approach, cooperative tasks are organized as event handlers that yield control by returning control to the event scheduler, manually unrolling their stacks. In this project they have adopted a hybrid approach that makes it possible for both stack management styles to coexist. Thus, programmers can code assuming one or the other of these stack management styles will operate, depending upon the application. The speaker also gave a detailed example of how to use their mechanism to switch between the two styles.

The talk then continued with the implementation. They were able to preempt many subtle concurrency problems by using cooperative task management. Paying a cost up-front to reduce a subtle race condition proved to be a good investment.

Though the choice of task management is fundamental, the choice of stack management can be left to individual taste. This project enables use of any type of stack management in conjunction with cooperative task management.

IMPROVING WAIT-FREE ALGORITHMS FOR INTERPROCESS COMMUNICATION IN EMBEDDED REAL-TIME SYSTEMS

Hai Huang, Padmanabhan Pillai, Kang G. Shin, University of Michigan

The main characteristic of the real-time/time-sensitive system is its predictable response time. But concurrency management creates hurdles to achieving this goal because of the use of locks to maintain consistency. To solve this problem, many wait-free algorithms have been developed, but these are typically high-cost solutions. By taking advantage of the temporal characteristics of the system, however, the time and space overhead can be reduced.

The speaker presented an algorithm for temporal concurrency control and applied this technique to improve three wait-free algorithms. A single-writer, multiple-reader, wait-free algorithm and a double-buffer algorithm were proposed.

Using their transformation mechanism, they achieved an improvement of 17–66% in ACET and a 14–70% reduction in memory requirements for IPC algorithms. This mechanism is extensible and can be applied to other non-blocking IPC algorithms as well. Future work involves reducing the synchronizing overheads in more general IPC algorithms with multiple-writer semantics.

MOBILITY

Summarized by Praveen Yalagandula

ROBUST POSITIONING ALGORITHMS FOR DISTRIBUTED AD-HOC WIRELESS SENSOR NETWORKS

Chris Savarese, Jan Rabaey, University of California, Berkeley ; Koen Langendoen, Delft University of Technology

The Pico Radio Network comprises more than a hundred sensors, monitors, and actuators equipped with wireless connectivity with the following properties: (1) no infrastructure, (2) computation in a distributed fashion instead of centralized computation, (3) dynamic topology, (4) limited radio range,

(5) nodes that can act as repeaters, (6) devices of low cost and with low power, and (7) sparse anchor nodes – nodes with GPS capability. A positioning algorithm determines the geographical position of each node in the network.

In two-dimensional space, each node needs three reference positions to estimate its geographical position. There are two problems that make positioning difficult in the Pico Radio Network type setting: (1) a sparse anchor node problem, and (2) a range error problem.

The two-phase approach that the authors have taken in solving the problem consists of: (1) a hop-terrain algorithm – in this first phase, each node roughly guesses its location by the distance calculated using multi-hops to the anchor points; and (2) a refinement algorithm – in this second phase, each node uses its neighbors' positions to refine its own position estimate. To guarantee the convergence, this approach uses confidence metrics: assigning a value of 1.0 for anchor nodes and 0.1 to start for other nodes and increasing these with each iteration.

A simulation tool, OMNet++, was used for both phases and in various scenarios. The results show that they achieved position errors of less than 33% in a scenario with 5% anchor nodes, an average connectivity of 7, and 5% range measurement error.

Guidelines for anchor node deployment are: high connectivity (>10), a reasonable fraction of anchor nodes (> 5%), and, for anchor placement, covered edges.

APPLICATION-SPECIFIC NETWORK MANAGEMENT FOR ENERGY-AWARE STREAMING OF POPULAR MULTIMEDIA FORMATS

Surendar Chandra, University of Georgia; and Amin Vahdat, Duke University
The main hindrance in supporting the increasing demand for mobile multimedia on PDA is the battery capacity of the

small devices. The idle-time power consumption is almost the same as the receive power consumption on typical wireless network cards (1319mW vs. 1425mW), while the sleep state consumes far less power (177mW). To let the system consume energy proportional to the stream quality, the network card should transition to sleep state aggressively between each packet.

The speaker presented previous work, where different multimedia streams were studied for PDAs: MS Media, Real Media, and Quicktime. It was found that if the inter-packet gap is predictable, then huge savings are possible: for example, about 80% savings in MS media streams with few packet losses. The limitation of the IEEE 802.11b power-save mode comes into play when there are two or more nodes, producing a delay between beacon time and when the node receives/sends packets. This badly affects the streams, since higher energy is consumed while waiting.

The authors propose traffic shaping for energy conservation where this is done by proxy such that packets arrive at regular intervals. This is achieved using a local proxy in the access point and a client-side proxy in the mobile host. Simulations show that traffic shaping reduces energy consumption and also reveals that the higher bandwidth streams have a lower energy metric (mJ/kB).

More information is at <http://greenhouse.cs.uga.edu>.

CHARACTERIZING ALERT AND BROWSE SERVICES OF MOBILE CLIENTS

Atul Adya, Paramvir Bahl, and Lili Qiu, Microsoft Research

Even though there is a dramatic increase in Internet access from wireless devices, there are not many studies done on characterizing this traffic. In this paper, the authors characterize the traffic observed on the MSN Web site with both notification and browse traces.

Around 33 million browsing accesses and about 3.25 million notification entries are present in the traces. Three types of analyses are done for each one of these two services: content analysis, concerning the most popular content categories and their distribution; popularity analysis, or the popularity distribution of documents; and user behavior analysis.

The analysis of the notification logs shows that document access rates follow a Zipf-like distribution, with most of the accesses concentrated on a small number of messages; and the accesses exhibit geographical locality – users from same locality tend to receive similar notification content. The browser log analysis shows that a smaller set of URLs are accessed most times, though the access pattern does not fit any Zipf curve; and the highly accessed URLs remain stable. A correlation study between the notification and browsing services shows that wireless users have a moderate correlation of 0.12.

FREENIX TRACK SESSIONS

BUILDING APPLICATIONS

Summarized by Matt Butner

INTERACTIVE 3-D GRAPHICS APPLICATIONS FOR TCL

Oliver Kersting, Jürgen Döllner, Hasso Plattner Institute for Software Systems Engineering, University of Potsdam
The integration of 3-D image rendering functionality into a scripting language permits interactive and animated 3-D development and application without the formalities and precision demanded by low-level C/C++ graphics and visualization libraries. The large and complex C++ API of the Virtual Rendering System (VRS) can be combined with the conveniences of the Tcl scripting language. The mapping of class interfaces is done via an automated process and generates respective wrapper classes, all of which ensures complete API accessibility and functionality without any signifi-

cant performance retribution. The general C++-to-Tcl mapper SWIG grants the necessary object and hierarchal abilities without the use of object-oriented Tcl extensions. The final API mapping techniques address C++ features such as classes, overloaded methods, enumerations, and inheritance relations. All are implemented in a proof-of-concept that maps the complete C++ API of the VRS to Tcl and are showcased in a complete interactive 3-D-map system.

THE AGFL GRAMMAR WORK LAB

Cornelis H.A. Coster, Erik Verbruggen, University of Nijmegen (KUN)

The growth and implementation of Natural Language Processing (NLP) is the cornerstone of the continued evolution and implementation of truly intelligent search machines and services. In part due to the growing collections of computer-stored human-readable documents in the public domain, the implementation of linguistic analysis will become necessary for desirable precision and resolution. Subsequently, such tools and linguistic resources must be released into the public domain, and they have done so with the release of the AGFL Grammar Work Lab under the GNU Public License, as a tool for linguistic research and the development of NLP-based applications.

The AGFL (Affix Grammars over a Finite Lattice) Grammar Work Lab meshes context-free grammars with finite set-valued features that are acceptable to a range of languages. In computer science terms, “Syntax rules are procedures with parameters and a non-deterministic execution.” The English Phrases for Information Retrieval (EP4IR), released with the AGFL-GWL as a robust grammar of English, is an AGFL-GWL generated English parser that outputs “Head/Modified” frames. The sentences “CompanyX sponsored this conference” and “This conference was sponsored by CompanyX” both generate [CompanyX,[sponsored, confer-

ence]]. The hope is that public availability of such tools will encourage further development of grammar and lexical software systems.

SWILL: A SIMPLE EMBEDDED WEB SERVER LIBRARY

Sotiria Lampoudi, David M. Beazley, University of Chicago

This paper won the FREENIX Best Student Paper award.

SWILL (Simple Web Interface Link Library) is a simple Web server in the form of a C library whose development was motivated by a wish to give cool applications an interface to the Web. The SWILL library provides a simple interface that can be efficiently implemented for tasks that vary from flexible Web-based monitoring to software debugging and diagnostics. Though originally designed to be integrated with high-performance scientific simulation software, the interface is generic enough to allow for unbounded uses. SWILL is a single-threaded server, relying upon non-blocking I/O through the creation of a temporary server which services I/O requests. SWILL does not provide SSL or cryptographic authentication but does have HTTP authentication abilities.

A fantastic feature of SWILL is its support for SPMD-style parallel applications which utilize MPI, proving valuable for Beowulf clusters and large parallel machines. Another practical application was the implementation of SWILL in a modified Yalnix emulator by University of Chicago Operating Systems courses, which utilized the added Yalnix functionality for OS development and debugging. SWILL requires minimal memory overhead and relies upon the HTTP/1.0 protocol.

NETWORK PERFORMANCE

Summarized by Florian Buchholz

LINUX NFS CLIENT WRITE PERFORMANCE

Chuck Lever, Network Appliance; Peter Honeyman, CITI, University of Michigan

Lever introduced a benchmark to measure an NFS client write performance. Client performance is difficult to measure due to hindrances such as poor hardware or bandwidth limitations. Furthermore, measuring application performance does not identify weaknesses specifically at the client side. Thus a benchmark was developed trying to exercise only data transfers in one direction between server and application. For this purpose, the benchmark was based on the block sequential write portion of the Bonnie file system benchmark. Once a benchmark for NFS clients is established, it can be used to improve client performance.

The performance measurements were performed with an SMP Linux client and both a Linux NFS server and a Network Appliance F85 filer. During testing, a periodic jump in write latency time was discovered. This was due to a rather large number of pending write operations that were scheduled to be written after certain threshold values were exceeded. By introducing a separate daemon that flushes the cached write request, the spikes could be eliminated, but as a result the average latency grows over time. The problem could be traced to a function that scans a linked list of write requests. After having added a hashtable to improve lookup performance, the latency improved considerably.

The improved client was then used to measure throughput against the two servers. A discrepancy between the Linux server and the filer test was noticed and the reason for that traced back to a global kernel lock that was unnecessarily held when accessing the network stack. After correcting this, performance improved further. However,

the measurements showed that a client may run slower when paired with fast servers on fast networks. This is due to heavy client interrupt loads, more network processing on the client side, and more global kernel lock contention.

The source code of the project is available at <http://www.citi.umich.edu/projects/nfs-perf/patches/>

A STUDY OF THE RELATIVE COSTS OF NETWORK SECURITY PROTOCOLS

Stefan Miltchev and Sotiris Ioannidis, University of Pennsylvania; Angelos Keromytis, Columbia University

With the increasing need for security and integrity of remote network services, it becomes important to quantify the communication overhead of IPSec and compare it to alternatives such as SSH, SCP, and HTTPS.

For this purpose, the authors set up three testing networks: direct link, two hosts separated by two gateways, and three hosts connecting through one gateway. Protocols were compared in each setup, and manual keying was used to eliminate connection setup costs. For IPSec the different encryption algorithms AES, DES, 3DES, hardware DES, and hardware 3DES were used. In detail, FTP was compared to SFTP, SCP and FTP over IPSec, HTTP to HTTPS and HTTP over IPSec, and NFS to NFS over IPSec and local disk performance.

The result of the measurements were that IPSec outperforms other popular encryption schemes. Overall, unencrypted communication was fastest, but in some cases, like FTP, the overhead can be small. The use of crypto hardware can significantly improve performance. For future work, the inclusion of setup costs, hardware-accelerated SSL, SFTP, and SSH were mentioned.

CONGESTION CONTROL IN LINUX TCP

Pasi Sarolahti, University of Helsinki; Alexey Kuznetsov, Institute for Nuclear Research at Moscow

Having attended the talk and read the paper, I am still unclear about whether the authors are merely describing the design decisions of TCP congestion control or whether they are actually the creators of that part of the Linux code. My guess leans toward the former.

In the talk, the speaker compared the TCP protocol congestion control measures according to IETF and RFC specifications with the actual Linux implementation, which does conform to the basic principles but nevertheless has differences. A specific emphasis was placed on retransmission mechanisms and the congestion window. Also, several TCP enhancements – the NewReno algorithm, Selective ACKs (SACK), Forward ACKs (FACK) – were discussed and compared.

In some instances, Linux does not conform to the IETF specifications. The fast recovery does not fully follow RFC 2582 since the threshold for triggering retransmit is adjusted dynamically and the congestion window's size is not changed. Also, the roundtrip-time estimator and the RTO calculation differ from RFC 2988 since it uses more conservative RTT estimates and a minimum RTO of 200ms. The performance measures showed that with the additional Linux-specific features enabled, slightly higher throughput, more steady data flow, and fewer unnecessary retransmissions can be achieved.

XTREME XCITEMENT

Summarized by Steve Bauer

THE FUTURE IS COMING: WHERE THE X WINDOW SHOULD GO

Jim Gettys, Compaq Computer Corp.

Jim Gettys, one of the principal authors of the X Window System, outlined the near-term objectives for the system, primarily focusing on the changes and infrastructure required to enable replication and migration of X applications. Providing better support for this functionality would enable users to retrieve or duplicate X applications between their servers at home and work.

One interesting example of an application that currently is capable of migration and replication is Emacs. To create a new frame on DISPLAY try: “M-x make-frame-on-display <Return> DISPLAY <Return>”.

However, technical challenges make replication and migration difficult in general. These include the “major headaches” of server-side fonts, the nonuniformity of X servers and screen sizes, and the need to appropriately retrofit toolkits. Authentication and authorization issues are obviously also important. The rest of the talk delved into some of the details of these interesting technical challenges.

HACKING IN THE KERNEL

Summarized by Hai Huang

AN IMPLEMENTATION OF SCHEDULER ACTIVATIONS ON THE NETBSD OPERATING SYSTEM

Nathan J. Williams, Wasabi Systems
Scheduler activation is an old idea. Basically, there are benefits and drawbacks to using solely kernel-level or user-level threading. Scheduler activation is able to combine the two layers of control to provide more concurrency in the system.

In his talk, Nathan gave a fairly detailed description of the implementation of scheduler activation in the NetBSD kernel. One important change in the implementation is to differentiate the thread context from the process context. This is done by defining a separate data structure for these threads and relocating some of the information that was embedded in the process context to these thread contexts. Stack was especially a concern due to the upcall. Special handling must be done to make sure that the upcall doesn't mess up the stack so that the preempted user-level thread can continue afterwards. Lastly, Nathan explained that signals were handled by upcalls.

AUTHORIZATION AND CHARGING IN PUBLIC WLANS USING FREEBSD AND 802.1X

Pekka Nikander, Ericsson Research
NomadicLab

802.1x standards are well known in the wireless community as link-layer authentication protocols. In this talk, Pekka explained some novel ways of using the 802.1x protocols that might be of interest to people on the move. It is possible to set up a public WLAN that would support various charging schemes via virtual tokens which people can purchase or earn and later use.

This is implemented on FreeBSD using the netgraph utility. It is basically a filter in the link layer that would differentiate traffic based on the MAC address of the client node, which is either authenticated, denied, or let through. The overhead for this service is fairly minimal.

ACPI IMPLEMENTATION ON FREEBSD

Takanori Watanabe, Kobe University
ACPI (Advanced Configuration and Power Management Interface) was proposed as a joint effort by Intel, Toshiba, and Microsoft to provide a standard and finer-control method of managing power states of individual devices within a system. Such low-level power management is especially important for those mobile and embedded systems that are powered by fixed-capacity energy batteries.

Takanori explained that the ACPI specification is composed of three parts: tables, BIOS, and registers. He was able to implement some functionalities of ACPI in a FreeBSD kernel. ACPI Component Architecture was implemented by Intel, and it provides a high-level ACPI API to the operating system. Takanori's ACPI implementation is built upon this underlying layer of APIs.

ACCESS CONTROL

Summarized by Florian Buchholz

DESIGN AND PERFORMANCE OF THE OPENBSD STATEFUL PACKET FILTER (PF)

Daniel Hartmeier, Systor AG

Daniel Hartmeier described the new stateful packet filter (pf) that replaced IPFilter in the OpenBSD 3.0 release. IPFilter could no longer be included due to licensing issues and thus there was a need to write a new filter, making use of optimized data structures.

The filter rules are implemented as a linked list which is traversed from start to end. Two actions may be taken according to the rules: “pass” or “block.” A “pass” action forwards the packet and a “block” action will drop it. Where more than one rule matches a packet, the last rule wins. Rules that are marked as “final” will immediately terminate any further rule evaluation. An optimization called “skip-steps” was also implemented, where blocks of similar rules are skipped if they cannot match the current packet. These skip-steps are calculated when the rule set is loaded. Furthermore, a state table keeps track of TCP connections. Only packets that match the sequence numbers are allowed. UDP and ICMP queries and replies are considered in the state table, where initial packets will create an entry for a pseudo-connection with a low initial timeout value. The state table is implemented using a balanced binary search tree. NAT mappings are also stored in the state table, whereas application proxies reside in user space. The packet filter also is able to perform fragment reassembly and to modulate TCP sequence numbers to protect hosts behind the firewall.

Pf was compared against IPFilter as well as Linux's Iptables by measuring throughput and latency with increasing traffic rates and different packet sizes. In a test with a fixed rule set size of 100, Iptables outperformed the other two filters, whose results were close to each

other. In a second test, where the rule set size was continually increased, Iptables consistently had about twice the throughput of the other two (which evaluate the rule set on both the incoming and outgoing interfaces). A third test compared only pf and IPFilter, using a single rule that created state in the state table with a fixed-state entry size. Pf reached an overloaded state much later than IPFilter. The experiment was repeated with a variable-state entry size and pf performed much better than IPFilter for a small number of states.

In general, rule set evaluation is expensive and benchmarks only reflect extreme cases, whereas in real life, other behavior should be observed. Furthermore, the benchmarks show that stateful filtering can actually improve performance due to cheap state-table lookups as compared to rule evaluation.

ENHANCING NFS CROSS-ADMINISTRATIVE DOMAIN ACCESS

Joseph Spadavecchia and Erez Zadok, Stony Brook University

The speaker presented modification to an NFS server that allows an improved NFS access between administrative domains. A problem lies in the fact that NFS assumes a shared UID/GID space, which makes it unsafe to export files outside the administrative domain of the server. Design goals were to leave protocol and existing clients unchanged, a minimum amount of server changes, flexibility, and increased performance.

To solve the problem, two techniques are utilized: "range-mapping" and "file-cloaking." Range-mapping maps IDs between client and server. The mapping is performed on a per-export basis and has to be manually set up in an export file. The mappings can be 1-1, N-N, or N-1. In file-cloaking, the server restricts file access based on UID/GID and special cloaking-mask bits. Here users can only access their own file permissions. The policy on whether or not the file is visible to others is set by the cloaking mask, which is logically ANDed with the

file's protection bits. File-cloaking only works, however, if the client doesn't hold cached copies of directory contents and file-attributes. Because of this the clients are forced to re-read directories by incrementing the mtime value of the directory each time it is listed.

To test the performance of the modified server, five different NFS configurations were evaluated. An unmodified NFS server was compared against one server with the modified code included but not used, one with only range-mapping enabled, one with only file-cloaking enabled, and one version with all modifications enabled. For each setup, different file system benchmarks were run. The results show only a small overhead when the modifications are used, generally an increase of below 5%. Another experiment tested the performance of the system with an increasing number of mapped or cloaked entries on a system. The results show that an increase from 10 to 1000 entries resulted in a maximum of about 14% cost in performance.

One member of the audience pointed out that if clients choose to ignore the changed mtimes from the server and thus still hold caches of the directory entries, the file-cloaking mechanism could be defeated. After a rather lengthy debate, the speaker had to concede that the model doesn't add any extra security. Another question was asked about scalability of the setup of range mapping. The speaker referred to application-level tools that could be developed for that purpose.

The software is available at <ftp://ftp.fsl.cs.sunysb.edu/pub/enf>.

ENGINEERING OPEN SOURCE SOFTWARE

Summarized by Teri Lampoudi

NINGAUI: A LINUX CLUSTER FOR BUSINESS

Andrew Hume, AT&T Labs – Research; Scott Daniels, Electronic Data Systems Corp.

Ningai is a general purpose, highly available, resilient architecture built

from commodity software and hardware. Emphatically, however, Ningai is not a Beowulf. Hume calls the cluster design the "Swiss canton model," in which there are a number of loosely affiliated independent nodes, with data replicated among them. Jobs are assigned by bidding and leases, and cluster services done as session-based servers are sited via generic job assignment. The emphasis is on keeping the architecture end-to-end, checking all work via checksums, and logging everything. The resilience and high availability required by their goal of 8-5 maintenance – vs. the typical 24-7 model where people get paged whenever the slightest thing goes wrong, regardless of the time of day – is achieved by job restartability. Finally, all computation is performed on local data, without the use of NFS or network attached storage.

Hume's message is a hopeful one: despite the many problems encountered – things like kernel and service limits, auto-installation problems, TCP storms, and the like – the existence of source code and the paranoid practice of logging and checksumming everything has helped. The final product performs reasonably well, and it appears that the resilient techniques employed do make a difference. One drawback, however, is that the software mentioned in the paper is not yet available for download.

CPCMS: A CONFIGURATION MANAGEMENT SYSTEM BASED ON CRYPTOGRAPHIC NAMES

Jonathan S. Shapiro and John Vanderburgh, Johns Hopkins University

This paper won the FREENIX Best Paper award.

The basic notion behind the project is the fact that everyone has a pet complaint about CVS, and yet it is currently the configuration manager in most widespread use. Shapiro has unleashed an alternative. Interestingly, he did not begin but ended with the usual host of reasons why CVS is bad. The talk instead began abruptly by characterizing the job

of a software configuration manager, continued by stating the namespaces which it must handle, and wrapped up with the challenges faced.

X MEETS Z: VERIFYING CORRECTNESS IN THE PRESENCE OF POSIX THREADS

Bart Massey, Portland State University; Robert T. Bauer, Rational Software Corp.

Massey delivered a humorous talk on the insight gained from applying Z formal specification notation to system software design rather than the more informal analysis and design process normally used.

The story is told with respect to writing XCB, which replaces the Xlib protocol layer and is supposed to be thread friendly. But where threads are concerned, deadlock avoidance becomes a hard problem that cannot be solved in an ad-hoc manner. But full model checking is also too hard. In this case Massey resorted to Z specification to model the XCB lower layer, abstract away locking and data transfers, and locate fundamental issues. Essentially, the difficulties of searching the literature and locating information relevant to the problem at hand were overcome. As Massey put it, “the formal method saved the day.”

FILE SYSTEMS

Summarized by Bosko Milekic

PLANNED EXTENSIONS TO THE LINUX EXT2/EXT3 FILESYSTEM

Theodore Y. Ts'o, IBM; Stephen Tweedie, Red Hat

The speaker presented improvements to the Linux Ext2 file system with the goal of allowing for various expansions while striving to maintain compatibility with older code. Improvements have been facilitated by a few extra superblock fields that were added to Ext2 not long before the Linux 2.0 kernel was released. The fields allow for file system features to be added without compromising the existing setup; this is done by providing

bits indicating the impact of the added features with respect to compatibility. Namely, a file system with the “incompat” bit marked is not allowed to be mounted. Similarly, a “read-only” marking would only allow the file system to be mounted read-only.

Directory indexing changes linear directory searches with a faster search using a fixed-depth tree and hashed keys. File system size can be dynamically increased, and the expanded inode, doubled from 128 to 256 bytes, allows for more extensions.

Other potential improvements were discussed as well, in particular, pre-allocation for contiguous files which allows for better performance in certain setups by pre-allocating contiguous blocks. Security-related modifications, extended attributes and ACLs, were mentioned. An implementation of these features already exists but has not yet been merged into the mainline Ext2/3 code.

RECENT FILESYSTEM OPTIMISATIONS ON FREEBSD

Ian Dowse, Corvil Networks; David Malone, CNRI, Dublin Institute of Technology

David Malone presented four important file system optimizations for FreeBSD OS: soft updates, dirpref, vmiodir, and dirhash. It turns out that certain combinations of the optimizations (beautifully illustrated in the paper) may yield performance improvements of anywhere between 2 and 10 orders of magnitude for real-world applications.

All four of the optimizations deal with file system metadata. Soft updates allow for asynchronous metadata updates. Dirpref changes the way directories are organized, attempting to place child directories closer to their parents, thereby increasing locality of reference and reducing disk-seek times. Vmiodir trades some extra memory in order to achieve better directory caching. Finally, dirhash, which was implemented by Ian Dowse, changes the way in which entries

are searched in a directory. Specifically, FFS uses a linear search to find an entry by name; dirhash builds a hashtable of directory entries on the fly. For a directory of size n , with a working set of m files, a search that in certain cases could have been $O(n*m)$ has been reduced, due to dirhash, to effectively $O(n + m)$.

FILESYSTEM PERFORMANCE AND SCALABILITY IN LINUX 2.4.17

Ray Bryant, SGI; Ruth Forester, IBM LTC; John Hawkes, SGI

This talk focused on performance evaluation of a number of file systems available and commonly deployed on Linux machines. Comparisons, under various configurations, of Ext2, Ext3, ReiserFS, XFS, and JFS were presented.

The benchmarks chosen for the data gathering were pgmeter, filemark, and AIM Benchmark Suite VII. Pgmeter measures the rate of data transfer of reads/writes of a file under a synthetic workload. Filemark is similar to postmark in that it is an operation-intensive benchmark, although filemark is threaded and offers various other features that postmark lacks. AIM VII measures performance for various file-system-related functionalities; it offers various metrics under an imposed workload, thus stressing the performance of the file system not only under I/O load, but also under significant CPU load.

Tests were run on three different setups: a small, a medium, and a large configuration. ReiserFS and Ext2 appear at the top of the pile for smaller and medium setups. Notably, XFS and JFS perform worse for smaller system configurations than the others, although XFS clearly appears to generate better numbers than JFS. It should be noted that XFS seems to scale well under a higher load. This was most evident in the large-system results, where XFS appears to offer the best overall results.

THINGS TO THINK ABOUT*Summarized by Bosko Milekic***SPEEDING UP KERNEL SCHEDULER BY REDUCING CACHE MISSES**

Shuji Yamamura, Akira Hirai, Mitsuru Sato, Masao Yamamoto, Akira Naruse, Kouichi Kumon, Fujitsu Labs

This was an interesting talk pertaining to the effects of cache coloring for task structures in the Linux kernel scheduler (Linux kernel 2.4.x). The speaker first presented some interesting benchmark numbers for the Linux scheduler, showing that as the number of processes on the task queue was increased, the performance decreased. The authors used some really nifty hardware to measure the number of bus transactions throughout their tests and were thus able to reasonably quantify the impact that cache misses had in the Linux scheduler.

Their experiments led them to implement a cache coloring scheme for task structures, which were previously aligned on 8KB boundaries and, therefore, were being eventually mapped to the same cache lines. This unfortunate placement of task structures in memory induced a significant number of cache misses as the number of tasks grew in the scheduler.

The implemented solution consisted of aligning task structures to more evenly distribute cached entries across the L2 cache. The result was, inevitably, fewer cache misses in the scheduler. Some negative effects were observed in certain situations. These were primarily due to more cache slots being used by task structure data in the scheduler, thus forcing data previously cached there to be pushed out.

WORK-IN-PROGRESS REPORTS*Summarized by Brennan Reynolds***RESOURCE VIRTUALIZATION TECHNIQUES FOR WIDE-AREA OVERLAY NETWORKS**

Kartik Gopalan, University Stony Brook

This work addressed the issue of provisioning a maximum number of virtual overlay networks (VON) with diverse quality of service (QoS) requirements on a single physical network. Each of the VONs is logically isolated from others to ensure the QoS requirements. Gopalan mentioned several critical research issues with this problem that are currently being investigated. Dealing with how to provision the network at various levels (link, route, or path) and then enforce the provisioning at run-time is one of the toughest challenges. Currently, Gopalan has developed several algorithms to handle admission control, end-to-end QoS, route selection, scheduling, and fault tolerance in the network.

For more information, visit <http://www.ecsl.cs.sunysb.edu/>.

VISUALIZING SOFTWARE INSTABILITY

Jennifer Bevan, University of California, Santa Cruz

Detection of instability in software has typically been an afterthought. The point in the development cycle when the software is reviewed for instability is usually after it is difficult and costly to go back and perform major modifications to the code base. Bevan has developed a technique to allow the visualization of unstable regions of code that can be used much earlier in the development cycle. Her technique creates a time series of dependent graphs that include clusters and lines called fault lines. From the graphs a developer is able to easily determine where the unstable sections of code are and proactively restructure them. She is currently working on a prototype implementation.

For more information, visit http://www.cse.ucsc.edu/~jbevan/evo_viz/.

RELIABLE AND SCALABLE PEER-TO-PEER WEB DOCUMENT SHARING

Li Xiao, William and Mary College

The idea presented by Xiao would allow end users to share the content of the Web browser caches with neighboring Internet users. The rationale for this is that today's end users are increasingly connected to the Internet over high-speed links, and the browser caches are becoming large storage systems. Therefore if an individual accesses a page which does not exist in their local cache, Xiao is suggesting that they first query other end users for the content before trying to access the machine hosting the original. This strategy does have some serious problems associated with it that still need to be addressed, including ensuring the integrity of the content and protecting the identity and privacy of the end users.

SEREL: FAST BOOTING FOR UNIX

Leni Mayo, Fast Boot Software

Serel is a tool that generates a visual representation of a UNIX machine's boot-up sequence. It can be used to identify the critical path and can show if a particular service or process blocks for an extended period of time. This information could be used to determine where the largest performance gains could be realized by tuning the order of execution at boot-up. Serel creates a dependency graph expressed in XML during boot-up. This graph is then used to create the visual representation. Currently the tool only works on POSIX-compliant systems, but Mayo stated that he would be porting it to other platforms. Other extensions that were mentioned included having the metadata include the use of shared libraries and monitoring the suspend/resume sequence of portable machines.

For more information, visit <http://www.fastboot.org/>.

BERKELEY DB XML

John Merrells, Sleepycat Software
Merrells gave a quick introduction and overview of the new XML library for Berkeley DB. The library specializes in storage and retrieval of XML content through a tool called XPath. The library allows for multiple containers per document and stores everything natively as XML. The user is also given a wide range of elements to create indices with, including edges, elements, text strings, or presence. The XPath tool consists of a query parser, generator, optimizer, and execution engine. To conclude his presentation, Merrell gave a live demonstration of the software.

For more information, visit
<http://www.sleepycat.com/xml/>.

CLUSTER-ON-DEMAND (COD)

Justin Moore, Duke University
Modern clusters are growing at a rapid rate. Many have pushed beyond the 5000-machine mark, and deploying them results in large expenses as well as management and provisioning issues. Furthermore, if the cluster is “rented” out to various users it is very time-consuming to configure it to a user’s specs regardless of how long they need to use it. The COD work presented creates dynamic virtual clusters within a given physical cluster. The goal was to have a provisioning tool that would automatically select a chunk of available nodes and install the operating system and middleware specified by the customer in a short period of time. This would allow a greater use of resources, since multiple virtual clusters can exist at once. By using a virtual cluster, the size can be changed dynamically. Moore stated that they have created a working prototype and are currently testing and benchmarking its performance.

For more information, visit
<http://www.cs.duke.edu/~justin/cod/>.

CATACOMB

Elias Sinderson, University of California, Santa Cruz

Catacomb is a project to develop a database-backed DASL module for the Apache Web server. It was designed as a replacement for the WebDAV. The initial release of the module only contains support for a MySQL database but could be extended to others. Sinderson briefly touched on the performance of her module. It was comparable to the `mod_dav` Apache module for all query types but search. The presentation was concluded with remarks about adding support for the lock method and including ACL specifications in the future.

For more information, visit
<http://ocean.cse.ucsc.edu/catacomb/>.

SELF-ORGANIZING STORAGE

Dan Ellard, Harvard University
Ellard’s presentation introduced a storage system that tuned itself, based on the workload of the system, without requiring the intervention of the user. The intelligence was implemented as a virtual self-organizing disk that resides below any file system. The virtual disk observes the access patterns exhibited by the system and then attempts to predict what information will be accessed next. An experiment was done using an NFS trace at a large ISP on one of their email servers. Ellard’s self-organizing storage system worked well, which he attributes to the fact that most of the files being requested were large email boxes. Areas of future work include exploration of the length and detail of the data collection stage, as well as the CPU impact of running the virtual disk layer.

VISUAL DEBUGGING

John Costigan, Virginia Tech
Costigan feels that the state of current debugging facilities in the UNIX world is not as good as it should be. He proposes the addition of several elements to programs like `ddd` and `gdb`. The first addition is including separate heap and

stack components. Another is to display only certain fields of a data structure and have the ability to zoom in and out if needed. Finally, the debugger should provide the ability to visually present complex data structures, including linked lists, trees, etc. He said that a beta version of a debugger with these abilities is currently available.

For more information, visit <http://infovis.cs.vt.edu/datastruct/>.

IMPROVING APPLICATION PERFORMANCE THROUGH SYSTEM-CALL COMPOSITION

Amit Purohit, University of Stony Brook
Web servers perform a huge number of context switches and internal data copying during normal operation. These two elements can drastically limit the performance of an application regardless of the hardware platform it is run on. The Compound System Call (CoSy) framework is an attempt to reduce the performance penalty for context switches and data copies. It includes a user-level library and several kernel facilities that can be used via system calls. The library provides the programmer with a complete set of memory-management functions. Performance tests using the CoSy framework showed a large saving for context switches and data copies. The only area where the savings between conventional libraries and CoSy were negligible was for very large file copies.

For more information, visit
<http://www.cs.sunysb.edu/~purohit/>.

ELASTIC QUOTAS

John Oscar, Columbia University
Oscar began by stating that most resources are flexible but that, to date, disks have not been. While approximately 80 percent of files are short-lived, disk quotas are hard limits imposed by administrators. The idea behind elastic quotas is to have a non-permanent storage area that each user can temporarily use. Oscar suggested the creation of a `/ehome` directory structure to be used in deploying elastic quotas. Currently, he

has implemented elastic quotas as a stackable file system. There were also several trends apparent in the data. Many users would ssh to their remote host (good) but then launch an email reader on the local machine which would connect via POP to the same host and send the password in clear text (bad). This situation can easily be remedied by tunneling protocols like POP through SSH, but it appeared that many people were not aware this could be done. While most of his comments on the use of the wireless network were negative, the list of passwords he had collected showed that people were indeed using strong passwords. His recommendations were to educate and encourage people to use protocols like IPSec, SSH, and SSL when conducting work over a wireless network, because you never know who else is listening.

SYSTRACE

Niels Provos, University of Michigan
How can one be sure the applications one is using actually do exactly what their developers said they do? Short answer: you can't. People today are using a large number of complex applications, which means it is impossible to check each application thoroughly for security vulnerabilities. There are tools out there that can help, though. Provos has developed a tool called systrace that allows a user to generate a policy of acceptable system calls a particular application can make. If the application attempts to make a call that is not defined in the policy, the user is notified and allowed to choose an action. Systrace includes an auto-policy generation mechanism which uses a training phase to record the actions of all programs the user executes. If the user chooses not to be bothered by applications breaking policy, systrace allows default enforcement actions to be set. Currently, systrace is implemented for FreeBSD and NetBSD, with a Linux version coming out shortly.

For more information, visit <http://www.citi.umich.edu/u/provos/systrace/>.

VERIFIABLE SECRET REDISTRIBUTION FOR SURVIVABLE STORAGE SYSTEMS

Ted Wong, Carnegie Mellon University
Wong presented a protocol that can be used to re-create a file distributed over a set of servers, even if one of the servers is damaged. In his scheme the user must choose how many shares the file is split into and the number of servers it will be stored across. The goal of this work is to provide persistent, secure storage of information, even if it comes under attack. Wong stated that his design of the protocol was complete and he is currently building a prototype implementation of it.

For more information, visit <http://www.cs.cmu.edu/~tmwong/research/>.

THE GURU IS IN SESSIONS

LINUX ON LAPTOP/PDA

Bdale Garbee, HP Linux Systems Operation

Summarized by Brennan Reynolds

This was a roundtable-like discussion with Garbee acting as moderator. He is currently in charge of the Debian distribution of Linux and is involved in porting Linux to platforms other than i386. He has successfully help port the kernel to the Alpha, Sparc, and ARM and is currently working on a version to run on VAX machines.

A discussion was held on which file system is best for battery life. The Riser and Ext3 file systems were discussed; people commented that when using Ext3 on their laptops the disc would never spin down and thus was always consuming a large amount of power. One suggestion was to use a RAM-based file system for any volume that requires a large number of writes and to only have the file system written to disk at infrequent intervals or when the machine is shut down or suspended.

A question was directed to Garbee about his knowledge of how the HP-Compaq merger would affect Linux. Garbee thought that the merger was good for Linux, both within the company and for the open source community as a whole. He stated that the new company would be the number one shipper of systems with Linux as the base operating system. He also fielded a question about the support of older HP servers and their ability to run Linux, saying that indeed Linux has been ported to them and he personally had several machines in his basement running it.

A question which sparked a large number of responses concerned problems people had with running Linux on mobile platforms. Most people actually did not have many problems at all. There were a few cases of a particular machine model not working, but there were only two widespread issues: docking stations and the new ACPI power management scheme. Docking stations still appear to be somewhat of a headache, but most people had developed ad-hoc solutions for getting them to work. The ACPI power management scheme developed by Intel does not appear to have a quick solution. Ted T'so, one of the head kernel developers, stated that there are fundamental architectural problems with the 2.4 series kernel that do not easily allow ACPI to be added. However, he also stated that ACPI is supported in the newest development kernel, 2.5, and will be included in 2.6.

The final topic was the possibility of purchasing a laptop without paying any charge/tax for Microsoft Windows. The consensus was that currently this is not possible. Even if the machine comes with Linux, or without any operating system, the vendors have contracts in place which require them to pay Microsoft for each machine they ship if that machine is capable of running a Microsoft operating system. The only way this will change is if the demand for

other operating systems increases to the point that vendors renegotiate their contracts with Microsoft, and no one saw this happening in the near future.

LARGE CLUSTERS

Andrew Hume, AT&T Labs – Research
Summarized by Teri Lampoudi

This was a session on clusters, big data, and resilient computing. Given that the audience was primarily interested in clusters and not necessarily big data or beating nodes to a pulp, the conversation revolved mainly around what it takes to make a heterogeneous cluster resilient. Hume also presented a FREENIX paper on his current cluster project, which explained in more detail much of what was abstractly claimed in the guru session.

Hume's use of the word "cluster" referred not to what I had assumed would be a Beowulf-type system but to a loosely coupled farm of machines in which concurrency was much less of an issue than it would be in a Beowulf. In fact, the architecture Hume described was designed to deal with large amounts of independent transaction processing, essentially the process of billing calls, which requires no interprocess message passing or anything of the sort a parallel scientific application might.

Where does the large data come in? Since the transactions in question consist of large numbers of flat files, a mechanism for getting the files onto nodes and off-loading the results is necessary. In this particular cluster, named Ningau, this task is handled by the "replication manager," which generated a large amount of interest in the audience. All management functions in the cluster, as well as job allocation, constitute services that are to be bid for and leased out to nodes. Furthermore, failures are handled by simply repeating the failed task until it is completed properly if that is possible, and if it is not, then looking through detailed logs to discover

what went wrong. Logging and testing for the successful completion of each task are what make this model resilient. Every management operation is logged at some appropriate level of detail, generating 120MB/day, and every single file transfer is md5 checksummed. This was characterized by Hume as a "patient" style of management, where no one controls the cluster, scheduling behavior is emergent rather than stringently planned, and nodes are free to drop in and out of the cluster; a downside to this model is the increase in latency, offset by the fact that the cluster continues to function unattended outside of 8-to-5 support hours.

In response to questions about the projected scalability of such a scheme, Hume said the system would presumably scale from the present 60 nodes to about 100 without modifications, but that scaling higher would be a matter for further design. The guru session ended on a somewhat unrelated note regarding the problems of getting data off mainframes and onto Linux machines – issues of fixed- vs. variable-length encoding of data blocks resulting in useful information being stripped by FTP, and problems in converting COBOL copybooks to something useful on a C-based architecture. To reiterate Hume's claim, there are homemade solutions for some of these problems, and the reader should feel free to contact Mr. Hume for them.

WRITING PORTABLE APPLICATIONS

Nick Stoughton, MSB Consultants
Summarized by Josh Lothian

Nick Stoughton addressed a concern that developers are facing more and more: writing portable applications. He proposed that there is no such thing as a "portable application," only those that have been ported. Developers are still in search of the Holy Grail of applications programming: a write-once, compile-anywhere program. Languages such as Java are leading up to this, but virtual

machines still have to be ported, paths will still be different, etc. Web-based applications are another area that could be promising for portable applications.

Nick went on to talk about the future of portability. The recently released POSIX 2001 standards are expected to help the situation. The 2001 revision expands the original POSIX sepc into seven volumes. Even though POSIX 2001 is more specific than the previous release, Nick pointed out that even this standard allows for small differences where vendors may define their own behaviors. This can only hurt developers in the long run. Along these lines, it was pointed out that there may be room for automated tools that have the capability to check application code and determine the level of portability and standards compliance of the source.

NETWORK MANAGEMENT, SYSTEM PERFORMANCE TUNING

Jeff Allen, Tellme Networks Inc.
Summarized by Matt Selsky

Jeff Allen, author of Cricket, discussed network tuning. The basic idea of tuning is measure, twiddle, and measure again. Cricket can be used for large installations, but it's overkill for smaller installations, for which Mrtg is better suited. You don't need Cricket to do measurement. Doing something like a shell script that dumps data to Excel is good, but you need to get some sort of measurement. Cricket was not designed for billing; it was meant to help answer questions.

Useful tools and techniques covered included looking for the difference between machines in order to identify the causes for the differences; measuring but also thinking about what you're measuring and why; remembering to use strace and tcpdump to identify problems. Problems repeat themselves; performance tuning requires an intricate understanding of all the layers involved to solve the problem and simplify the automation. (Having a computer science

background helps but is not essential.) If you can reproduce the problem, you then should investigate each piece to find the bottleneck. If you can't reproduce the problem, then measure. You need to understand all the system interactions. Measurement can help determine whether things are actually slow or if the user is imagining it.

Troubleshooting begins with a hunch, but scientific processes are essential. You should be able to determine the event stream, or when each event occurs; having observation logs can help. Some hints provided include checking cron for periodic anomalies, slowing down `/bin/rm` to avoid I/O overload, and looking for unusual kernel CPU usage.

Also, try to use lightweight monitoring to reduce overhead. You don't need to monitor every resource on every system, but you should monitor those resources on those systems that are essential. Don't check things too often, since you can introduce overhead that way. Lots of information can be gathered from the `/proc` file system interface to the kernel.

BSD BOF

Host: Kirk McKusick, Author and Consultant

Summarized by Bosko Milekic

The BSD Birds of a Feather session at USENIX started off with five quick and informative talks and ended with an exciting discussion of licensing issues.

Christos Zoulas for the NetBSD Project went over the primary goals of the NetBSD Project (namely, portability, a clean architecture, and security) and then proceeded to briefly discuss some of the challenges that the project has encountered. The successes and areas for improvement of 1.6 were then examined. NetBSD has recently seen various improvements in its cross-build system, packaging system, and third-party tool support. For what concerns the kernel, a zero-copy TCP/UDP implementation has been integrated, and a new pmap

code for arm32 has been introduced, as have some other rather major changes to various subsystems. More information is available in Christos's slides, which are now available at <http://www.netbsd.org/gallery/events/usenix2002/>.

Theo de Raadt of the OpenBSD Project presented a rundown of various new things that the OpenBSD and OpenSSH teams have been looking at. Theo's talk included an amusing description (and pictures!) of some of the events that transpired during OpenBSD's hack-a-thon, which occurred the week before USENIX '02. Finally, Theo mentioned that, following complications with the IPFilter license, the OpenBSD team had done a fairly extensive license audit.

Robert Watson of the FreeBSD Project brought up various examples of FreeBSD being used in the real world. He went on to describe a wide variety of changes that will surface with the upcoming FreeBSD release, 5.0. Notably, 5.0 will introduce a re-architecting of the kernel aimed at providing much more scalable support for SMP machines. 5.0 will also include an early implementation of KSE, FreeBSD's version of scheduler activations, large improvements to pccard, and significant framework bits from the TrustedBSD project.

Mike Karels from Wind River Systems presented an overview of how BSD/OS has been evolving following the Wind River Systems acquisition of BSDi's software assets. Ernie Prabhakar from Apple discussed Apple's OS X and its success on the desktop. He explained how OS X aims to bring BSD to the desktop while striving to maintain a strong and stable core, one that is heavily based on BSD technology.

The BoF finished with a discussion on licensing; specifically, some folks questioned the impact of Apple's licensing for code from their core OS (the Darwin Project) on the BSD community as a whole.

CLOSING SESSION

HOW FLIES FLY

Michael H. Dickinson, University of California, Berkeley

Summarized by J.D. Welch

In this lively and offbeat talk, Dickinson discussed his research into the mechanisms of flight in the fruit fly (*Drosophila melanogaster*) and related the autonomic behavior to that of a technological system. Insects are the most successful organisms on earth, due in no small part to their early adoption of flight. Flies travel through space on straight trajectories interrupted by saccades, jumpy turning motions somewhat analogous to the fast, jumpy movements of the human eye. Using a variety of monitoring techniques, including high-speed video in a "virtual reality flight arena," Dickinson and his colleagues have observed that flies respond to visual cues to decide when to saccade during flight. For example, more visual texture makes flies saccade earlier (i.e., further away from the arena wall), described by Dickinson as a "collision control algorithm."

Through a combination of sensory inputs, flies can make decisions about their flight path. Flies possess a visual "expansion detector," which, at a certain threshold, causes the animal to turn a certain direction. However, expansion in front of the fly sometimes causes it to land. How does the fly decide? Using the virtual reality device to replicate various visual scenarios, Dickinson observed that the flies fixate on vertical objects that move back and forth across the field. Expansion on the left side of the animal causes it to turn right, and vice versa, while expansion directly in front of the animal triggers the legs to flare out in preparation for landing.

If the eyes detect these changes, how are the responses implemented? The flies' wings have "power muscles," controlled by mechanical resonance (as opposed to the nervous system), which drive the

wings, combined with neurally activated “steering muscles,” which change the configuration of the wing joints. Subtle variations in the timing of impulses correspond to changes in wing movement, controlled by sensors in the “wing-pit.” A small, wing-like structure, the haltere, controls equilibrium by beating constantly.

Voluntary control is accomplished by the halteres, whose signals can interfere with the autonomic control of the stroke cycle. The halteres have “steering muscles” as well, and information derived from the visual system can turn off the haltere or trick it into a “virtual” problem requiring a response.

Dickinson has also studied the aerodynamics of insect flight, using a device called the “Robo Fly,” an oversized mechanical insect wing suspended in a large tank of mineral oil. Interestingly, Dickinson observed that the average lift generated by the flies’ wings is under its body weight; the flies use three mechanisms to overcome this, including rotating the wings (rotational lift).

Insects are extraordinarily robust creatures, and because Dickinson analyzed the problem in a systems-oriented way, these observations and analysis are immediately applicable to technology, the response system can be used as an efficient search algorithm for control systems in autonomous vehicles, for example.

THE AFS WORKSHOP

Summarized by Garry Zacheiss, MIT
Love Hornquist-Astrand of the Arla development team presented the Arla status report. Arla 0.35.8 has been released. Scheduled for release soon are improved support for Tru64 UNIX, MacOS X, and FreeBSD, improved volume handling, and implementation of more of the vos/pts subcommands. It was stressed that MacOS X is considered an important platform, and that a GUI configuration manager for the cache

manager, a GUI ACL manager for the finder, and a graphical login that obtains Kerberos tickets and AFS tokens at login time were all under development.

Future goals planned for the underlying AFS protocols include GSSAPI/SPNEGO support for Rx, performance improvements to Rx, an enhanced disconnected mode, and IPv6 support for Rx; an experimental patch is already available for the latter. Future Arla-specific goals include improved performance, partial file reading, and increased stability for several platforms. Work is also in progress for the RXGSS protocol extensions (integrating GSSAPI into Rx). A partial implementation exists, and work continues as developers find time.

Derrick Brashear of the OpenAFS development team presented the OpenAFS status report. OpenAFS was released immediately prior to the conference; OpenAFS 1.2.5 fixed a remotely exploitable denial-of-service attack in several OpenAFS platforms, most notably IRIX and AIX. Future work planned for OpenAFS includes better support for MacOS X, including working around Finder interaction issues. Better support for the BSDs is also planned; FreeBSD has a partially implemented client; NetBSD and OpenBSD have only server programs available right now. AIX 5, Tru64 5.1A, and MacOS X 10.2 (a.k.a. Jaguar) are all planned for the future. Other planned enhancements include nested groups in the ptserver (code donated by the University of Michigan awaits integration), disconnected AFS, and further work on a native Windows client. Derrick stated that the guiding principles of OpenAFS were to maintain compatibility with IBM AFS, support new platforms, ease the administrative burden, and add new functionality.

AFS performance was discussed. The openafs.org cell consists of two file servers, one in Stockholm, Sweden, and one in Pittsburgh, PA. AFS works reasonably well over trans-Atlantic links,

although OpenAFS clients don’t determine which file server to talk to very effectively. Arla clients use RTTs to the server to determine the optimal file server to fetch replicated data from. Modifications to OpenAFS to support this behavior in the future are desired.

Jimmy Engelbrecht and Harald Barth of KTH discussed their AFSCrawler script, written to determine how many AFS cells and clients were in the world, what implementations/versions they were (Arla vs. IBM, AFS vs. OpenAFS), and how much data was in AFS. The script unfortunately triggered a bug in IBM AFS 3.6–derived code, causing some clients to panic while handling a specific RPC. This has since been fixed in OpenAFS 1.2.5 and the most recent IBM AFS patch level, and all AFS users are strongly encouraged to upgrade. No release of Arla is vulnerable to this particular denial-of-service attack. There was an extended discussion of the usefulness of this exploration. Many sites believed this was useful information and such scanning should continue in the future, but only on an opt-in basis.

Many sites face the problem of managing Kerberos/AFS credentials for batch-scheduled jobs. Specifically, most batch processing software needs to be modified to forward tickets as part of the batch submission process, renew tickets and tokens while the job is in the queue and for the lifetime of the job, and properly destroy credentials when the job completes. Ken Hornstein of NRL was able to pay a commercial vendor to support Kerberos 4/5 credential management in their product, although they did not implement AFS token management. MIT has implemented some of the desired functionality in OpenPBS, and might be able to make it available to other interested sites.

Tools to simplify AFS administration were discussed, including:

- AFS Balancer. A tool written by CMU to automate the process of balancing disk usage across all

servers in a cell. Available from <ftp://ftp.andrew.cmu.edu/pub/AFS-Tools/balance-1.1b.tar.gz>.

- **Themis.** Themis is KTH's enhanced version of the AFS tool "package," for updating files on local disk from a central AFS image. KTH's enhancements include allowing the deletion of files, simplifying the process of adding a file, and allowing the merging of multiple rule sets for determining which files are updated. Themis is available from the Arla CVS repository.

Stanford was presented as an example of a large AFS site. Stanford's AFS usage consists of approximately 1.4TB of data in AFS, in the form of approximately 100,000 volumes. 3.3TB of storage is available in their primary cell, ir.stanford.edu. Their file servers consist entirely of Solaris machines running a combination of Transarc 3.6 patch-level 3 and OpenAFS 1.2.x, while their database servers run OpenAFS 1.2.2. Their cell consists of 25 file servers, using a combination of EMC and Sun StorEdge hardware. Stanford continues to use the kserver for their authentication infrastructure, with future plans to migrate entirely to an MIT Kerberos 5 KDC.

Stanford has approximately 3400 clients on their campus, not including SLAC (Stanford Linear Accelerator); approximately 2100 AFS clients from outside Stanford contact their cell every month. Their supported clients are almost entirely IBM AFS 3.6 clients, although they plan to release OpenAFS clients soon. Stanford currently supports only UNIX clients. There is some on-campus presence of Windows clients, but Stanford has never publicly released or supported it. They do intend to release and support the MacOS X client in the near future.

All Stanford students, faculty, and staff are assigned AFS home directories with a default quota of 50MB, for a total of approximately 550GB of user home directories. Other significant uses of AFS

storage are data volumes for workstation software (400GB) and volumes for course Web sites and assignments (100GB).

AFS usage at Intel was also presented. Intel has been an AFS site since 1994. They had bad experiences with the IBM 3.5 Linux client; their experience with OpenAFS on Linux 2.4.x kernels has been much better. They use and are satisfied with the OpenAFS IA64 Linux port. Intel has hundreds of OpenAFS 1.2.3 and 1.2.4 clients in many production cells, accessing data stored on IBM AFS file servers. They have not encountered any interoperability issues. Intel has some concerns about OpenAFS; they would like to purchase commercial support for OpenAFS and to see OpenAFS support for HP-UX on both PA-RISC and Itanium hardware. HP-UX support is currently unavailable due to a specific HP-UX header file being unavailable from HP; this may be available soon. Intel has not yet committed to migrating their file servers to OpenAFS and are unsure if they will do so without commercial support.

Backups are a traditional topic of discussion at AFS workshops, and this time was no exception. Many users complain that the traditional AFS backup tools ("backup" and "butc") are complex and difficult to automate, requiring many home-grown scripts and much user intervention for error recovery. An additional complaint was that the traditional AFS tools do not support file- or directory-level backups and restores; data must be backed up and restored at the volume level.

Mitch Collinsworth of Cornell presented work to make AMANDA, the free backup software from the University of Maryland, suitable for AFS backups. Using AMANDA for AFS backups allows one to share AFS backup tapes with non-AFS backups, easily run multiple backups in parallel, automate error recovery, and provide a robust degraded mode that prevents tape errors from

stopping backups altogether. Their implementation allows for full volume restores as well as individual directory and file restores. They have finished coding this work and are in the process of testing and documenting it.

Peter Honeyman of CITI at the University of Michigan spoke about work he has proposed to replace Rx with RPCSEC GSS in OpenAFS; this would allow AFS to use a TCP-based transport mechanism, rather than the UDP-based Rx, and possibly gain better congestion control, dynamic adaptation, and fragmentation avoidance as a result. RPCSEC GSS uses the GSSAPI to authenticate SUN ONC RPC. RPCSEC GSS is transport agnostic, provides strong security, is a developing Internet standard, and has multiple open source implementations. Backward compatibility with existing AFS servers and clients is an important goal of this project.

2002 Linux Kernel Developers Summit

JUNE 24–25, OTTAWA, CANADA

Summarized by David B. Sharp

This year's two-day summit featured participants armed with laptops of every size, feature, and brand. The summit was followed by the four-day Ottawa Linux Symposium, with many of the talks and presenters following up on topics discussed during the summit.

REQUIREMENTS FOR CARRIER GRADE LINUX

Timothy D. Witham, OSDL

Timothy presented the requirements for carrier grade Linux. Carrier grade service, as required by the telecom sector, differs from enterprise grade service in that enterprise networks typically service networks with nodes that number in the thousands. Carrier grade networks, however, service nodes that may number in the millions. Thus, an interruption of a carrier grade network can be very costly, especially when service agreements with clients include penalties. Ser-

vice levels usually demand 5 9s of reliability.

OSDL outlined a list of priorities they felt were required to achieve carrier grade performance from a Linux-based network. Some of these include:

- Compliance with LSB, POSIX, Ipv6, IPsec MIPv6, and all three SNMP versions.
- SCTP: this is a new streaming protocol built on top of IP and sister to UDP and TCP.
- Boot-cycle detection to sense frequent reboots.
- Hot-swapability.
- Alternate boot selection; if not A, then B.
- Six 9s availability.
- A hardened driver specification.
- Application heartbeat monitors.
- A watchdog timer pre-interrupt. This would allow systems and applications to be notified prior to a timeout by a watchdog timer.
- Live upgrades and rollbacks. This would allow the state of a system to be captured and, in the event of failure of an upgrade, to be rolled back to its former stable version.
- Better serviceability. This would allow a system in the field to be serviced remotely with tools such as dynamic debug and probe insertion.
- Device enumeration of hot-plugged devices.
- The ability to service a component without having to take the whole system offline.
- Forced umount.
- Better user-level tools. This would allow users without significant kernel knowledge and experience to perform triage.
- Kernel profiling and system tools.
- Kernel preemption.
- A selectable scheduling policy framework and O(1) performance to better service soft realtime applications.

- Support for a high number of timers, threads, and processes. This item was an acknowledged problem, but such support would allow for easier porting of existing applications and models.

Many of the items listed above are currently in progress.

AMD HAMMER PORT (AMD)

Wayne Meretsky and Vojtech Pavlik, AMD

Wayne and Vojtech presented AMD's new Hammer series of 64-bit processors and outlined what would be required to implement Linux. Unlike Intel's implementation of the Itanium, with its entirely new architecture, the Hammer chip extends the IA-32 and adds two new instructions. The features for the Hammer include:

- Larger L1 and L2 caches and better branch prediction characteristics.
- Double the number of general purpose registers.
- A greater number of integer and floating point functional units. This and the previous two features allow, according to AMD's preliminary tests, 32-bit applications to run up to 20% faster.
- A "Hyper-Transport" (see Figure 1) high-bandwidth point-to-point interconnect built into the chip. This allows the Hammer chip to connect to other chips and devices. AMD stressed that this was not a replacement for PCI.
- Four page levels for memory access and a larger Translation Look-Aside Buffer (TLB). This differs from the current architecture, which uses three page levels. The first implementation will allow 512GB of 48-bit addressable memory.
- Memory controller support for up to 333MHz

DDR memory and future support for DDR2 memory.

KERNEL PARAMETERS

Rusty Russell and Patrick Mochel, OSDL

This discussion looked at some of the issues with boot parameters; these are used at boot time or during module loading to specify various kernel behaviors.

Rusty presented the problems with the MODULE_PARAM, one of which is there is no type checking. This would be replaced with a macro, PARAM(), which defines a callback for new variable types.

Patrick discussed the driverfs/. This is a virtual file system, similar to the /devices file system, used for debugging purposes, and allows the export of kernel device attributes in user space. The driverfs/ differs from the /proc tree in that there is one value per entry. One problem noted with the /proc tree is that it is not centrally maintained. The driverfs/ would enforce a strict interface, which should eliminate some of the anarchy in the 9 directory.

It was suggested that the driverfs/ could be used for event notification and that select polling would be easy to build in. Linus countered that this does not map to a lot of events you would want to have and that it would not be a convenient interface.

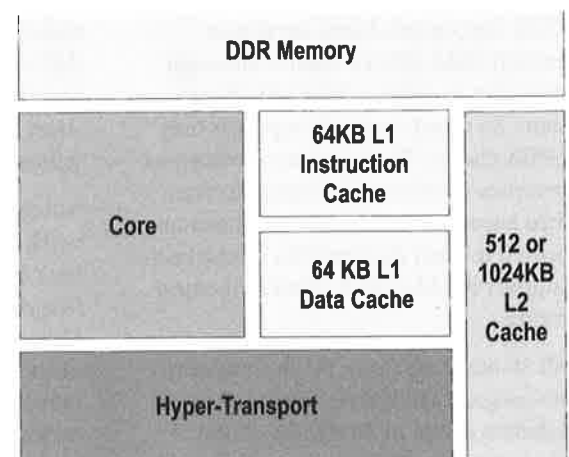


Figure 1: AMD Hammer Architecture

The `driverfs/` prompted a lot of discussion, in particular regarding what belongs in the `driverfs/`, what doesn't, and where the latter should be exported.

LIVING WITH MODULES

RUSTY RUSSELL

Rusty presented a candid and animated discussion on the problems with loadable modules. He started by stating that a module's only purpose is to add hardware.

The first problem he noted is with module implementation. A poorly written interface is almost impossible to use; the intended function needs to be clear. As an example of what constitutes a good interface, Rusty posed the problem of a module failing during `init`. He suggested that a kernel module linker be used instead of the `insmod` user space routine.

Initialization problems were then discussed. Most code assumes it is run at boot time; if there is a problem at this point, error code is often nonexistent. Someone asked about dropping module versioning for 2.5, but it was pointed out that this would be a nightmare for the vendor. Module versioning also deals with issues such as SMP and non-SMP modules.

One proposed initialization solution was to break the kernel registration into two parts. The first part would involve *reserving* the resources required by the module. If this stage were successful, the second stage would involve committing these resources with a `use` function call. This would solve the problem of bad entries in the `proc/` directory.

Another problem is that there is no way to force the removal of a module; for example, the module may always be in use. A solution for this was another two-stage clear-and-destroy approach. The first stage would involve a reference count for each module. When we want to remove a module, no new users will be allowed to reference the device. When

the reference count reaches zero, then the module can be destroyed.

VIRTUAL MEMORY

Andrea Archangeli, Daniel Phillips, and Rik van Riel

The first topic discussed in this presentation was reverse mapping of memory with a new function call to `rmap()`. This function would allow mapping of physical pages to page tables. There was some discussion about whether the overhead of `rmap()` would be worth it. With terabytes of memory, VM management is too complex and takes too much CPU time to manage page tables. By including reverse mapping, it was pointed out, VM balancing would be possible.

There was some discussion about how to benchmark systems with such large memory and whether `dbench` was a fair measurement tool. We need a tool to quantify measurements and evaluate changes against different workloads. Further, we need to agree on the different benchmarks and tests. If the "right" set of tests existed, then the process could be automated.

Another issue presented was accounting and out-of-memory conditions. We still cannot accurately measure an out-of-memory condition, i.e., how many pages are free. There is a need for a heuristic algorithm to provide proper memory balancing and an intelligent way to kill a task to free up memory. This algorithm must efficiently keep track of how much memory is currently pinned. The problem now is that failing to do so properly can lead to system deadlock.

Miscellaneous VM issues included:

- NUMA scalability. A suggestion to reduce the overhead was to put pages into groups of four, producing one-fourth of the page structs.
- Dealing with very large page tables. By using page coloring, an increase in performance of 300%–400% in page allocation and a complexity of $O(1)$ can be achieved.

BLOCK I/O

Jens Axboe

Block I/O is a big topic in 2.5. The first topic discussed dealt with ordered writes and barriers. Ordered writes using barriers are required to maintain the integrity of a journaling file system such as `ext3`. Write barriers can be achieved with IDE by setting a barrier bit to force cache flushing. SCSI can achieve the same thing using ordered tags. These need to be implemented at the device-driver level.

I/O scheduling was another topic discussed. The current method is a single-direction elevator algorithm. A new elevator algorithm with a bounded delay using a 1-second deadline resulted in a 3% decrease in throughput. With a 100ms deadline there is an 8% decrease in throughput. The current I/O schedule does not implement any priority to increase deterministic behavior. This needs to be looked at in the future.

BoF SESSIONS

The Birds of a Feather Sessions ran in three rooms and included VM, Crypto, `ext2` Htree, memory management, swapped-to-RAM, and swapped-to-disk.

SCALABILITY FOR DATABASE PERFORMANCE (IBM)

Ken Rozenthal

Ken gave a presentation on IBM's database requirements on Linux. The focus was on performance, scalability, and throughput; some of the requirements discussed included:

- Reduced capacity effects. The database is cached in user address space, and the goal is to have no page faults.
- Larger page size. One of the benefits of a larger page size is reduced TLB misses.
- Parallelizing I/O across many spindles of many devices.
- Large sequential-block I/O to avoid breaking up I/O requests.

- The ability to do raw I/O and avoid data buffering. The databases understand I/O and do not want to do it over a file system.
- Asynchronous I/O. There is a need for non-blocking completion detection.
- DMA support to high physical memory addresses to avoid copying.
- Reduced lock contention, with finer granularity device locks at the driver level.
- Numa effects. This can be accomplished with memory affinity, i.e., intelligent page allocation to the closest processor.

Again, it should be noted that many of the items are currently under construction (e.g., asynchronous I/O, by Ben LaHaise).

HP KERNEL WISH LIST

Bdale Garbee

Bdale's presentation of HP's wish-list requirements for Linux focused on:

- SCSI subsystem support for a LUN count of >1024, better error handling, and support for failover and load balancing.
- Large system support, with >256 PCI busses.
- Hot-plug support. Make PCMCIA work like other devices.
- Device naming. For example, would like to have an interface such as eth0 not move when a new device is plugged in.
- Unionfs support to transparently add other memory devices, such as docking stations to an Ipac.

During the discussion there was concern raised about multiple PCI bus handling. Bdale, on behalf of his Debian alter ego, discussed the problem of proprietary firmware being distributed with device drivers in the kernel distribution. The concern that this might violate the GPL prompted some discussion.

LOADABLE SECURITY MODULE

Chris Wright

In the classic UNIX security model, the root user has access to everything. Other users are assigned privileges based on individual or group access. There are many custom security patches available. The Loadable Security Model (LSM) solution looks at a standardized security model for Linux. The LSM does not, however, support auditing for Common Criteria.

The LSM, shown in Figure 2, is an intrusive patch that consists of approximately 150 hooks distributed in the kernel and mediates 15 core kernel objects; each hook controls the function of an individual process. The total size of LSM is 3000. Policies are registered one at a time, but it is possible to register a composition policy. The reported overhead for the LSM security, using lmbench tests, was 0%–2%; the exception noted

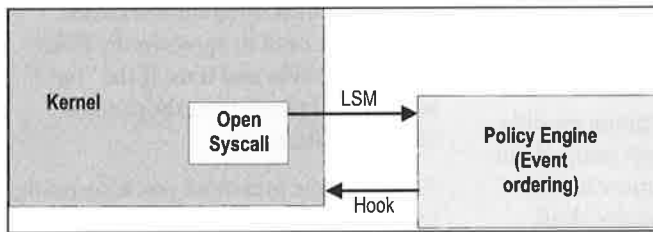


Figure 2

was GB Ethernet at 20%. LSM is available for both the 2.4 and 2.5 kernels.

ASYNCHRONOUS I/O

Ben LaHaise

Asynchronous I/O potentially has some far-reaching effects in the kernel. Most of the new syscalls in the fs/aio.c have not yet been implemented, but those that have been tested have worked well.

One of the issues raised was support for two streams of I/O, one for synchronous and one for asynchronous. This would result in a lot of duplicate code. The suggestion was made to build the synchronous I/O on top of a complete implementation of asynchronous I/O. This would result in a lot of things breaking in the kernel. Linus was sup-

portive and his philosophy in this area was to use the KISS Principle, doing only a minimal implementation, adding features as required, and fixing the problems as they come up.

SCSI

James Bottomley

This session provided an overview of the SCSI stack subsystem and some issues outstanding. Three of the areas that James focused on were the error-handling component, “cruft,” and reducing or eliminating the SCSI mid-layer, moving most of it to the upper layer.

The discussion started with a back-grounder on the SCSI stack. He noted that the error-handling component of the SCSI stack is inadequate and in need of a major rework. James discussed some of details regarding this component. He would like to replace the error-handler API with a more generic

message-passing API.

The next area he focused on was “cruft.” This is the result of a great deal of obsolete legacy code that needs to be reworked or eliminated. The SCSI subsystem is an impor-

tant component, and emphasis was placed on a careful but deliberate reworking.

Some of the miscellaneous items noted were that the SCSI needs to be able to deal with hot-pluggable devices; that does not exist at this point. The subject of SAN devices was brought up; with the current system, the entire network can be probed for devices. It was recommended that a “lazy,” or as-needed, allocation method be used. Overall this session was quite topical and provoked a good deal of constructive discussion.

GOALS FOR 2.6 – IMPROVING THE RELEASE-ENGINEERING PROCESS

Theodore Ts'o

Ted started out the closing session by asking for any suggestions on how to improve the next summit. These comments can be emailed. He noted that the size of the group was, at this point, quite manageable and productive and about as big as they would want to go.

The next discussion was on how to get 2.6 out. Currently, when a freeze is announced, there is a rush to get in all the last-minute patches. The result can often be that patches are not tested with other patches and things break. When things break, delays occur. There was some discussion about when the feature-freeze for the 2.6 release should occur, and Halloween (of this year!) was decided on.

There was some discussion of offloading from Linus the responsibility of maintaining stable releases, but no capable and willing volunteers found. (The paradox cited was that the person needed to be smart enough to do the job and stupid enough to take on the job.)

The session closed with a warm thanks to Ted for his work in organizing the summit.

BOF SESSIONS

The Birds of a Feather sessions discussed ACPI, SAN attachments, fbdev, and boot code issues and changes for faster boots.

2nd Java™ Virtual Machine Research and Technology Symposium

SAN FRANCISCO, CA
AUGUST 1–2, 2002

Summarized by Jose F. Osorio

OPENING REMARKS

JVM '02 Program Chair Sam Midkiff of IBM's T.J. Watson Research Center welcomed symposium attendees and provided an overview of the topics that were going to be presented in the intensive two-day event. These included

research work in the areas of JVM memory management, advanced JVM architectures, just-in-time compilers, method inlining, realtime JVMs, embedded JVMs for portable, personal, mobile devices, and hardware-based Java machines for stack-based microprocessors.

The Best Student Paper Award went to "Supporting Binary Compatibility with Static Compilation," by Dachuan Yu, Zhong Shao, and Valery Trifonov, Yale University. The project's Web site is <http://flint.cs.yale.edu/flint/publications/bincomp.html>. The Best Paper Award was given to "An Empirical Study of Method In-lining for a Java Just-in-Time Compiler," by Toshio Suganama, Toshinki Yasue, and Toshio Nakatani, IBM Tokyo Research Laboratory.

KEYNOTE ADDRESS: STOP THINKING OUTSIDE THE BOX. THERE IS NO BOX.

Robert Berry, Distinguished Engineer, IBM Centre for Java Technology

"Virtual Machine technology is here to stay as a fundamental technology supporting real business computing," said Robert Berry during his remarks.

As an absolute core of much of IBM's software and hardware business, JVMs are a proven technology for real computing: they form an essential component of WebSphere's Application Server runtime environment, are integrated with the DB2 database management system, and are a key enabler for Web services infrastructure. Proof of this is also Microsoft's .NET framework and its common language runtime (CLR) for managed code, which is a VM-based technology. Furthermore, JVMs are at the core of business computing. Java researchers are also evaluating CLR and looking into performance compared to JVMs.

Customer requirements such as viability and legacy expectation, competitive pressure such as functionality and performance, hardware evolution, technology changes, and new ideas are some of

the principal motivating drivers for JVM innovation, explained Berry.

IBM's focus on innovation reflects the growing maturity of JVMs, evolving from an inward client-first, server-next focus to virtual machines in the context of middleware such as database, transaction, and applications servers.

Berry presented a history trail of IBM's innovations in such key areas as JVM architecture, just-in-time compilation, middleware, and autonomic systems.

Innovations in JVM memory allocation and garbage collection were explained. Different garbage collection strategies were compared, including mark-sweep-compact, compaction avoidance, parallel marking, parallel marking plus sweeping, and concurrent marking. The impact of full vs. incremental compaction was discussed.

In the area of JIT compilation, Berry explained method inlining and escape analysis strategies (where the detection of objects that do not survive a particular method invocation scope is done to allocate them on the stack so as to reduce heap and memory synchronization overhead).

The high-performing JVMs' need for execution of short and repetitive transactions and strict isolation requirements between transactions, plus the constraints of a 31-bit addressing mode, were key factors that led to the invention of IBM's Persistent Reusable JVMs, which reduce initializing and startup JVM overhead, are equipped with efficient garbage collectors, and support transaction isolation among multiple JVMs.

In autonomic computing, the drive to increase reliability, availability, serviceability (RAS) resulted in the creation of two well-known Java specification requests (JSRs): JSR 174 – jvmmi for monitoring, and JSR 163 – jvmsi/ti for performance.

IBM has innovated successfully with JVM for four reasons: (1) they listen to their customers (whether external or internal) and to their users; (2) they provide significant investment in research and development for key JVM infrastructures; (3) they deploy novel technology in the field and in the community early enough for experimentation; and (4) they do not give up on any ideas.

Berry explained that work and innovation is needed in various areas, including JVM footprint, very large heaps (VLH) on the order of 500GB and beyond, object pooling, object sharing, decimal arithmetic, RAS, Web services, JVM execution performance, simplification and integration of resources monitoring, control and management, and the impact on JVM technology from aspect-oriented software development (the next generation of modularity beyond the object-oriented paradigm). Visit <http://www.aosd.org> for more information on aspect-oriented software development.

ADAPTIVE GARBAGE COLLECTION FOR BATTERY-OPERATED ENVIRONMENTS

G. Chen, M. Kandemir, N. Vijaykrishnan and M.J. Irwin, The Pennsylvania State University; M. Wolczko, Sun Microsystems

Narayanan presented an adaptive garbage collection strategy for embedded devices in which object creation and memory allocation history on a time interval is used to tune the frequency at which the garbage collector is invoked to reduce energy consumption from Java applications. He also showed how energy leakage is reduced by exploiting supply-gated power management mechanisms which shut down energy supply to memory banks that do not hold useful data. Using KVM – Sun Microsystems' JVM designed for embedded and battery-operated environments – and a set of nine applications typical for handheld devices, the adaptive garbage collection strategy incurred fewer performance penalties than other approaches.

CONCURRENT REMEMBERED SET REFINEMENT IN GENERATIONAL GARBAGE COLLECTION

David Detlefs and Ross Knippel, Sun Microsystems; William D. Clinger, Northeastern University; Matthias Jacob, Princeton University

With generational garbage collection, garbage collection latency can be decreased, increasing throughput and reducing execution pauses. It divides the heap memory into multiple layers of allocation generations. In this strategy the youngest generation is the one most frequently collected, and a complex data structure is used to represent the generation layers as remembered sets in order to identify links from older generations to younger ones.

The refined strategy focuses on the challenge of efficiently maintaining the accuracy of remembered sets in a concurrent environment in a way that reduces the cost penalties from write/update barriers involved in maintenance of the complex data structure and that scales well as the size of the old generation increases. The strategy uses a write barrier implementation whose cost is not significantly greater than card marking techniques.

Two remembered-set organizations were considered in the implementation of the proposed strategy: a card table augmented with a summary table, and a refined version of the first as a two-level card table (a coarse-grained summary table, and a fine-grained detail table with a 2N ratio between each table for some large N). Two write-barrier refinement techniques were considered as well: direct refinement on one-level and two-level card tables, and log-base refinement with mutator and refinement threads.

An attendee asked why hashtables were not used in the strategy implementation. Detlefs responded that separate benchmarks had shown that their two-level card table approach is as good as hashtable direct access during scanning of the coarse-grained table.

TO COLLECT OR NOT TO COLLECT? MACHINE LEARNING FOR MEMORY MANAGEMENT

Eva Andreasson, BEA/Appeal Virtual Machines; Frank Hoffmann, Royal Institute of Technology; Olof Lindholm, BEA/Appeal Virtual Machines

Andreasson explained how reinforcement learning contributes to JVM autonomous decision-making to perform garbage collection. Given the elements of system environment – State, Action, Reward, new State, new Action (SARSA) – Eva explained how after an iterative and systematic process of decision-making exploration and prediction, the SARSA approach yields an optimal learning scheme for adaptive garbage collection. Benchmarks demonstrated how the RLS (reinforcement learning system)-based JVM outperformed JRockit, a conventional JVM, in a dynamic environment in which memory allocation behavior changes more rapidly.

OPTIMIZING PRECISION OVERHEAD FOR X86 PROCESSORS

Takeshi Ogasawara, Hideaki Komatsu, and Toshio Nakatani, IBM Japan

A novel approach to optimize floating-point operations in JVMs for x86 target processors was presented by Takeshi Ogasawara. The strategy involves tracking floating-point precision-type code blocks in a Java program, performing region analysis, and also tracking precision-aware method invocations.

By transforming the original bytecode and generating code blocks with the appropriate floating-point precision, the default precision mode can be ignored. Region analysis investigates code blocks to find strategic points where single- and double-precision mode-switch instructions can be inlined to reduce rounding and store-reloads overhead. After transformation, the just-in-time compiled code calls the native target code with the same floating-point precision type, therefore eliminating any redundant mode switches across method boundaries of precision-aware invocations.

Ogasawara emphasized that their strategy does not sacrifice strictness of floating-point precision; it actually reduces the x86-specific overhead of preserving strictness.

A MODULAR AND EXTENSIBLE JVM INFRASTRUCTURE

Patrick Doyle and Tarek S. Abdelrahman, University of Toronto

The design of Jupiter, a modular scalable JVM framework, was outlined by Patrick Doyle. The Jupiter project investigates JVM architectures for high performance on large-scale parallel systems with 128+ microprocessor clusters and facilitates research and future enhancements to their JVM. Jupiter has been constructed out of a multitude of discrete units with small and simple interfaces in a manner similar to how shells in UNIX build complex command pipelines out of discrete programs. Numerous aspects of JVM system architecture (memory allocation, metadata, method dispatching, object manipulation, call stack, bytecode interpretation, just-in-time compilation multithreading, synchronization, and many others) are covered in this paper in a clear and simple manner.

A LIGHTWEIGHT JAVA VIRTUAL MACHINE FOR A STACK-BASED MICROPROCESSOR

Mirko Raner, PTSC

The Symposium ended with a presentation of a hybrid implementation of a JVM. The Ignite microprocessor is a stack-based bytecode processor with a 32-bit dual-stack architecture, an 18-word operand stack, and a separate 16-word stack for call-stack frames. It was originally designed as an embedded computing platform for efficient execution of C and Fortran. Ignite is not a Java technology-enabled microprocessor, but its architecture is very similar to the JVM, and Java bytecode translation is easy and efficient. Ported from the original LVM (Lightweight Virtual Machine), the main challenges were the separation from the underlying operating system by means of JELLO, an abstraction layer; optimization of ahead-of-time (ATO) compilation; lazy class

resolution; and optimization of method invocation.

Raner gave a detailed explanation of the operand stack, local stack, and global/general register-set structure, the differences between the stack model of a typical JVM and Ignite's stack model, and Ignite's instruction-set format.

Raner also explained that the differences between the argument-passing mechanisms of a standard JVM and Ignite's made it necessary to move method arguments from the Ignite's operand stack to Ignite's local variable stack.

The LVM is still a work in progress. Plans for its future include placement in very small devices, cellular phones, and mobile devices.

To view the presentation slides, visit <http://java.sh/lvm.html>.



**NordU2003 -
The fifth NordU/USENIX
Conference
February 10-14, 2003
Aros Congress Center
Västerås, Sweden**

Announcement

We would like to welcome everyone to Västerås (100 km east of Stockholm) and to the NordU2003 - The fifth NordU/USENIX Conference.

As UNIX and Free Software becomes more and more widely adopted in corporations and academia we see a need to stay on top of current trends.

The NordU/USENIX Conference offers a venue for developers, administrators and users of UNIX and UNIX like operating systems, to meet, talk and learn.

Jon "maddog" Hall will deliver one of the keynotes. Mr Hall has a long history of being involved in our community. We will take a special look at how Open Source Software, such as GNU/Linux, *BSD, Open Office and GNOME, is being used in government. For developers we have some exciting invited presentations on Java Technology as well as GNU Compiler Collection family of tools

We look forward to meeting you all in Västerås!

Programme

Monday - Wednesday

February 10-12

Tutorial Programme

Thursday - Friday

February 13-14

Technical Programme

Papers

Exhibition

Sponsors Presentations

Topics that will be covered

Security

Operating Systems

Desktop

Tools

Backup Solutions

Applications

Open Source/Free UNIX

Interoperability



NordU2003





MobiSys 2003

The First International Conference on Mobile Systems, Applications, and Services

May 5-8, 2003, Renaissance Parc 55 Hotel, San Francisco, CA, USA

<http://www.sigmobile.org/mobisys/2003/>

<http://www.usenix.org/events/mobisys03/>

Jointly sponsored by ACM SIGMOBILE and The USENIX Association, In cooperation with ACM SIGOPS

General Chair:

Dan Siewiorek
CMU

Program Committee:

Co-chair: Mary Baker
Stanford University

Co-chair: Robert T. Morris
MIT

Gaetano Borriello
University of Washington and
Intel Research Seattle

Claude Castelluccia
INRIA Rhone-Alpes

Hao Chu
DoCoMo USA Labs

David Culler
University of California,
Berkeley, and Intel Research
Berkeley

Nigel Davies
Lancaster University and
University of Arizona

Maria Ebling
IBM T. J. Watson Research
Center

Peter Honeyman
CITI, University of Michigan

Cristina Hristea
Stanford University

Tim Kindberg
HP Labs Palo Alto

David Kotz
Dartmouth College

Kevin Lai
University of California,
Berkeley

Adrian Perrig
Carnegie Mellon University

Doug Terry
Microsoft Research Silicon
Valley

Marvin Theimer
Microsoft Research

Henry Tirri
University of Helsinki

Amin Vahdat
Duke University

Deborah A. Wallach
HP Labs Palo Alto

Tutorial Chair:

Songwu Lu
UCLA

Research Demo Chair:

Tom Martin
Virginia Tech

Steering Committee

Chair:
Victor Bahl
Microsoft Research

IMPORTANT DATES:

Paper submissions due:	October 7, 2002
Notification of acceptance:	December 18, 2002
Camera-ready final papers due:	March 4, 2003

OVERVIEW: MobiSys 2003 seeks to present innovative, significant research in the area of mobile systems. This will be a 2.5-day conference, featuring refereed paper presentations, demos, poster sessions, and Birds-of-a-Feather sessions.

TOPICS: The conference addresses systems issues in mobile computing. Areas of interest include, but are not limited to:

- Design, implementation, and evaluation of mobile systems
- Experience with mobile systems
- Operating systems for small devices
- System-level energy management for mobile devices
- Middleware and service architectures for mobile applications
- Systems for location awareness and determination
- Data management for mobile applications
- Personal-area networks and systems
- Resource discovery of mobile services
- Applications and services supporting the mobile user
- Personal mobility
- Disconnected and weakly connected operation
- Security, privacy, authorization, and billing
- Infrastructure support for mobility
- Proxies and data adaptation
- Wearable and handheld devices
- Social and economic aspects of mobility
- Mobile agents
- User interfaces, programming interfaces, and applications for small devices

MobiSys is not an appropriate forum for research focused narrowly on the routing, link, or physical network layers, although papers describing larger mobile architectures that include work on these areas may be appropriate. Please feel free to contact the Program Chairs at mobisyschairs@mit.edu to determine appropriateness.

BEST PAPER AWARD: An award will be given at the conference for the best paper overall.

WHAT TO SUBMIT: Submissions should be full papers, 12-14 single-spaced 8.5" x 11" pages, including figures, tables, and references, two-column format, using 10-point type on 12-point (single-spaced) leading.

Submissions will be judged on originality, significance, interest, clarity, relevance, and correctness. Please see the conference Web site for complete submission instructions and author guidelines.

MobiSys, like most conferences and journals, requires that papers must not be submitted simultaneously to any other conference or publication, that submissions must not be previously published, and that accepted papers must not be subsequently published elsewhere. Papers accompanied by non-disclosure agreement forms are not acceptable and will be returned to the author(s) unread. All submissions will be held in the highest confidentiality prior to publication in the Proceedings, both as a matter of policy and in accord with the U.S. Copyright Act of 1976.

HOW TO SUBMIT: Authors are required to submit full papers by October 7, 2002. All submissions to MobiSys 2003 must be electronic, in PDF or PostScript. Please see the conference Web site for the submission form. Authors will be notified of receipt of submission via e-mail. If you do not receive notification, contact the Program Chairs at mobisyschairs@mit.edu.

POSTER SESSIONS: Do you have interesting work you would like to share, or a cool idea that is not ready to be published? Poster sessions are for you! Poster sessions, scheduled during the technical sessions, introduce new or ongoing work. The MobiSys audience provides valuable discussion and feedback. We are particularly interested in presentations of student work. To submit a poster, please send a proposal, one page or less, by February 1, 2003, to the poster session coordinator at mobisysposter@mit.edu.

BIRDS-OF-A-FEATHER SESSIONS: Birds-of-a-Feather sessions (BoFs) are very informal gatherings organized by attendees interested in a particular topic. BoFs will be held in the evenings. BoFs may be scheduled in advance by phoning the Conference Office at +1.510.528.8649 or via email to conference@usenix.org. BoFs may also be scheduled at the conference.

REGISTRATION MATERIALS: Complete program and registration information will be available in February 2003 on the conference Web site. The information will be in both HTML and 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.

NEW MEMBER BENEFIT

eLearning from DigitalThink

NOW AVAILABLE FOR OUR MEMBERSHIP – AT A 25% DISCOUNT
ALL YOU NEED IS A BROWSER!

USENIX and SAGE members can now register for DigitalThink's Web Based Training courses at a 25% discount. Through DigitalThink, you can **learn online, 24 hours a day, anytime, anywhere!** If you want to improve your skills, get certified, get promoted, learn about management, finance or other key skills, these courses can help get to where you want to go.

The DigitalThink courses were selected by a committee of USENIX and SAGE members after a comprehensive review of all the major eLearning providers. Courses cover a wide variety of topics including: UNIX, Linux, Java, Databases, Networking & Project Management. DigitalThink also offers unique, **integrated tutor support** to give you personal feedback and assist you with questions.

If you, your colleagues or subordinates need an educationally effective AND cost effective way to learn a specific skill, DigitalThink might be the best solution. Try one today!

The outline below offers a glimpse of the over 200 courses offered.

- Certification Database Concepts
- IT Management
- Java Programming
- Microsoft Certification
- MCDBA
- MCSD
- MCSE
- Networking
- Oracle
- Programming
- Red Hat Linux
- System and Network Administration
- UNIX
- Web Development
- Business Fundamentals
- Compliance
- Project Management
- Desktop Operating Systems

For a complete course catalogue and to register please visit:
<http://www.usenix.org/membership/digitalthink/index.html>



**“That’s
Mr. SmartyPants
to you.”**

Be the first to know what’s happening across your Web systems with SiteScope®

When it comes to the status of your Web systems, being a “Know-It-All” is a good thing. In fact, it’s a critical part of your job. SiteScope from Freshwater Software lets you proactively monitor your network services, system resources, and applications. You’re alerted to availability issues and other problems before they can rage out of control. Best of all, **it takes just 20 minutes to install, configure and begin using SiteScope**—no matter how many servers you’re monitoring. Looks like people will just have to get used to that look of supreme confidence on your face.

See how easy it can be to proactively monitor your Web systems:

**Download a FREE trial of SiteScope today
at www.freshwater.com/KnowIt2**

©2002 Freshwater Software, Inc., a Mercury Interactive company. All rights reserved. Freshwater and SiteScope are trademarks of Freshwater Software. All other trademarks or names are the property of their respective holders.



Freshwater Software®
a Mercury Interactive company

CONNECT WITH USENIX



MEMBERSHIP AND PUBLICATIONS

USENIX ASSOCIATION
2560 NINTH STREET, SUITE 215
BERKELEY, CA 94710
PHONE: 1+ 510 528 8649
FAX: 1+ 510 548 5738
EMAIL: office@usenix.org
login@usenix.org
conference@usenix.org



WEB SITES

<http://www.usenix.org>
<http://www.sage.org>



EMAIL

login@usenix.org



COMMENTS? SUGGESTIONS?

send email to ah@usenix.org

CONTRIBUTIONS SOLICITED

You 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 its entirety or in part requires the permission of the Association and the author(s).

USENIX & SAGE

The Advanced Computing Systems Association &
The System Administrators Guild

;login:

USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710

POSTMASTER
Send Address Changes to *;login:*
2560 Ninth Street, Suite 215
Berkeley, CA 94710

PERIODICALS POSTAGE
PAID
AT BERKELEY, CALIFORNIA
AND ADDITIONAL OFFICES