conference reports

This issue's report focusses on LISA XVI

Our thanks to the Summarizers:

for LISA '02
Josh Simon, who organized the collecting of the summaries
Abiodun A. Alao
Paul Anderson
David Berg
Robert Beverly
Kuzman Ganchev
Jim Hickstein
Rob Kolstad
Martin Krafft
Renuka Nayak
James O'Kane
Will Partain
Peg Schafer
J. D. Welch
Steve Wormley
Garry Zacheiss

LISA XVI

Sixteenth Systems Administration Conference

Philadelphia, Pennsylvania, USA

November 3–8, 2002

KEYNOTE

Scaling the Web: An Overview of Google (a Linux Cluster for Fun and Profit)

Jim Reese, Chief Operations Engineer, Google

Summarized by J.D. Welch

We all know, use, and love Google, but how do they make it work? In this engaging talk, Jim Reese explained how custom software, massive replication and expendable, commodity hardware have allowed Google to answer 150 million Web search queries a day.

The core technology that separates Google from other search services is the PageRank system developed by founders Larry Page and Sergey Brin while graduate students at Stanford University. This system aims to objectively rank Web content by popularity; according to Reese, "a page's importance is the sum of the aggregate importance of the pages linking to it," so a page linked to from the *New York Times* is given more weight than one linked to by a high-school newspaper. In addition to assessing popularity, hypertext analysis is used to quantify the importance of elements on a page (e.g., larger text is probably more important).

To get a sense of scale of Google's challenge – there are 3.8 billion pages and 256 million Web users, and 85% of them use search services. Given this, any single machine will always be too small for the task, so index and page data is divided up into pieces, called "shards," which are distributed across many machines and multiple data centers. Thus, traffic is scalable by replication; the index is read-only, so single failures aren't fatal but only reduce capacity.

To answer a query, the Web server (a custom package called gws) queries index servers, document servers (cached pages), and ad servers, in parallel, and keeps trying until it gets a response. Each query may involve a dozen or more servers, using whichever reply comes in the fastest (the average query time is .23 seconds). Before the query reaches a Web server, however, it passes several load balancers, both global and local, which use various methods (including round-robin and least connections) to choose which servers to query.

"El Cheapo" PCs are used to maximize reliability through replication. Fault tolerance is kept very simple; timeouts are in the milliseconds, and machines are restarted automatically and regularly polled for their status. Racks of machines are very dense, with 80 half-depth 1U boxes in each, along with paired switches, load balancers, and Gigabit uplinks to the routers. All disks are local (100–120Gb/machine); large fans are mounted atop the rack and heat is drawn from the space between the machines in the center of the rack. All the machines run a "Googlized" distribution of RedHat Linux as well as proprietary tools for serving content and system monitoring.

For comparison with the new, very organized racks, Reese showed photos of historical configurations, including a custom-built 1U machine with four motherboards, eight disks, eight NICs, and one power supply, which was configured with the disks mounted over the processors separated by a sheet of Plexiglas (!).

INVITED TALKS

SECURITY ON MACOS X

John Hurley, Apple

Summarized by J.D. Welch

Hurley began by saying that Apple is in an interesting position to deal with security issues, as they manufacture the hardware, firmware, operating system, and often the end application, so a great degree of integration is possible in OS X security features.

Since OS X is based on BSD, many of the OS X security tools are ports of standard UNIX tools, oftentimes GUIfied with a Cocoa (native OS X Objective-C framework) front end. For example, the Sharing and Firewall control panels are a front end to ipfw. OS X also offers Kerberos, OpenSSH, OpenSSL, and other familiar UNIX tools in its default installation. Obviously, the use of familiar, often open source, packages is a departure from (and significant improvement over) OS 9.

A primary goal in designing the OS X security architecture was to make it easy to use these important features. Additionally, although many tools are presented plainly for users, they are configurable beyond what most users would bother with – good news for longtime UNIX users and security types. Also, Software Update encourages users to keep up-to-date with patches, as it automatically polls for and delivers updates directly to end users.

OS X implements a Common Data Security Architecture API, which provides an expandable set of crypto algorithms to various applications, including the Keychain (encrypted user information store) and Disk Copy (which can encrypt disk volumes). These "layered services" include file signing and certificate management as well as APIs for adding plug-in modules for additional services. With this modular architecture, developers can make use of security services without having to know a great deal about the specific component or service at work.

OS X makes a point of separating authorization from authentication, a move designed for next-generation applications, including smart card access, for which they are developing an SDK (called Smart Card Services) in collaboration with HP, Intel, and other vendors.

Out of the box, OS X is reasonably well locked down: Services like SSH, HTTP are off by default (but are easy to enable – from GUI or command line – if you know what you're doing), no ports are open, and the root account is disabled (sudo is used for administrative access). OS X honors UNIX user/group/file permissions and is designed to be a multiuser OS.

The Keychain is a cornerstone application, and was given much play in this talk. Accessible to all Cocoa, Carbon, and UNIX applications running under OS X, the Keychain provides an encrypted environment to store passwords for Web sites and file servers, encrypted disk volumes, and the like. Users "unlock" the Keychain with a master password, and applications can store and read data from the Keychain. Additionally, all Keychain items include an access control list for fine-grained control.

Another highlighted technology was the ability of Disk Copy, a utility available on all installations of OS X, to create encrypted disk images. Once the image is created, it can be mounted (with successful authentication) read/write, burned on a CD for transfer, etc.

The physical security of Apple hardware has also been considered. The XServe 1U rack mount server, for example, sends messages to the console when physical security of the rack is compromised; other Mac models can be "locked" with the Open Firmware Password, which prevents booting the machine without first authenticating, and prevents the use of startup commands (which can make the machine act like a FireWire disk or be booted into UNIX-permissions-free OS 9, for example).

This talk was a little marketing-heavy and didn't delve into technical details of the various systems implemented in OS X beyond their GUI expression, but it did provide a good introduction to the various services available to the OS X user or administrator.

ETHICS FOR SYSTEM ADMINISTRATORS: DILEMMAS FOR LISA 2002 ATTENDEES

Lee Damon and Rob Kolstad

Summarized by Steve Wormley

Unlike the medical profession, which has had thousands of years to develop ethical standards, system administration ethics are new. The mapping of conventional communications such as paper mail and the telephone do not work in the realm of email and Instant Messaging. The quantity of sensitive data online and issues such as identity theft contribute to awareness of the need for ethics and privacy guidelines in new technology.

Since computer ethics is a new area, novel situations and their attendant problems now happen at "Internet speed." We as system administrators need to have knowledge of ethics, privacy, and security so that we can protect rights and still get work done.

One definition of a professional is a person who conforms to the technical and ethical standards of a profession. For system administration to be regarded as a profession by the outside world, therefore, ethical standards need to be addressed.

A distinction should be made between ethics and policies. Since policies are well defined and generally not open to interpretation, establishing a site policy will often eliminate many ethical problems.

Ethics in the context of computer networks pertain to all privileged users, including anyone with access to others' information, even if that access is accidental; even help desk personnel, for example, need to be included.

Lee and Rob went on to present five scenarios involving ethical dilemmas for system administrators:

1. A project you worked on at a previous client had a flaw which could kill people if not corrected, but you only realized the flaw while at your current client, working on something similar, and could lose your job if you disclosed the flaw.
2. Your boss asked you to read the CTO's email to look for evidence of wrongdoing. You found a problem, reported it to your boss, and nothing was done. Now what do you do?
3. The third scenario was the often-repeated case where the boss wants the root password but is not competent on the system. How would you handle it?
4. In the course of routine administration you discover that your boss is discussing doing something evil, such as going to a competitor with customer lists. Now what?
5. You are providing network connectivity to a neighbor with children, and the children receive pornographic email. What do you do?

The Constitutional and Financial Arguments against Spam

Daniel V. Klein

Summarized by Martin Krafft

A trip to Dan Klein's home page (*http://www.klein.com*) reveals that he's a geek leaning toward the humorous. In his talk on the constitutional and financial argument against spam, he used exactly that tack. "Spam steals my time" could be seen as the motto as Dan proceeded to unroll his theories on preventing spam, keeping his audience focused while he delivered facts and ideas that, if nothing else, were entertaining.

He isn't a lawyer – he stressed this fact several times – so he approached his topic from a "common-sense" angle. Spam, or Monty Python's breakfast delicacy, is all those emails you never asked for – commercial mails advertising get-rich-quick schemes, mortgage loans, advertisements for penis enlargement devices, and other breathtaking new technology you wouldn't lack before or after the spam hit you. Dan started out by presenting a short history of his involvement with the Net and his exposure to spam, and then proceeded to lay out the numbers of an 80-day research period, in which he received one spam every 29 seconds. Even using a fairly restrictive set of anti-spam techniques, he claimed the ratio of ham to spam he receives is about the same as Earth's mass to Jupiter's. But to place his figures into relation to the real world, Dan quotes hotmail.com as being burdened by one billion spam messages per day.

He attacks the problem from two sides, starting with the constitutional. Freedom of speech seems to be commonly misunderstood and extended to argue for spam. Yet freedom of speech has exceptions (e.g., screaming "fire" in a public theater for no reason). You can say what you want, Dan pointed out, "but I don't have to listen to it, I can disagree, and [most importantly] you cannot make me pay for something I don't want to hear." Taken together with freedom of the press (I can print or refuse to print whatever I want, and so can you), and the constitutional argument against spam is right there: you are forcing your spam to be printed on my press, and I have no choice but to receive it. What Dan criticizes is that spammers seem to misinterpret freedom of speech as a guarantee of an audience, and freedom of press as a free method to print.

His financial argument against spam claims that spam costs the American people in the vicinity of 165 billion (!) dollars per year. In contrast to the 15 billion dollars made available to NASA every year and the 300 million the RIAA loses to "piracy" per year, this figure clearly indicates there's something severely wrong. Advertising isn't evil, it's necessary. Rather, the lack of regulation and control is what constitutes the problem. Spam is the cheapest method of advertising since it mostly raises costs for the recipients. It is marketing with a bullhorn, as Dan put it. He wants to "take back the Net."

Current anti-spam methods almost all come in the form of a filter-and-block setup on the recipient side. As effective spam filters are becoming more and more of a marketing technique of big ISPs like AOL and gmx.net, the voices around the "censorship" buzzword seem to be getting louder and louder. Censorship, in Dan's view, is ubiquitous rather than evil. "Abolish Censorship" may sound good but it reveals how little is known about the topic. People tolerate censorship more than they are willing to acknowledge, and yet scream at the idea of having someone filter their mail.

Dan sees little use in current methods, such as whitelists and confirmation systems. He wants a legal solution, and if not on a global level, then at least within the United States as a starter. However, he couldn't lay out a strategy for how such a law would be enacted and controlled, for which he isn't to blame — anti-spam is a challenge to the entire infrastructure and requires a lot of cooperation, from the MTA author to the ISP, from the government to the end user. He wants a global opt-out mechanism rather than one focused on individual advertisers.

Dan's talk, albeit very amusing, did not really offer anything new. Some audience members came to the talk to be comforted about their spam problems, others to get an idea of what spam is about. As such, Dan succeeded in reviving the subject and making it a prominent one, for a number of

anti-spam-related topics and discussions were evident throughout the remainder of the conference.

Risk-Taking vs. Management

Paul Evans

Summarized by Jim Hickstein

Paul gave a post-mortem of a dot-com company, Webvan, extrapolating from that experience to the broader view that social misperceptions of risk skewed business decisions and contributed to the dot-com bubble. He also looked at why our profession did not have enough credibility with management to influence those decisions.

The essence of capitalism is putting assets at risk in the service of profit. Professional financial managers get paid to balance the equation of assets, risk, and profit. But managers are people, and people tend to underestimate familiar risks and overestimate unfamiliar ones.

Unfamiliar risk abounded in the dot-com world, but it was asymmetrically distributed. Taking Webvan as an example, the grocery business is pretty well understood: Financial, operations, and even software development risks, were familiar. But risks in IT were unfamiliar to management. The "prevailing doctrine of risk" changed: In 1999, it was about not appearing above the fold of the *Wall Street Journal*; in 2000, the slogan became "five nines," whether merited or not.

The result was a business that overspent on redundancy (the larger perceived risk), while making fatal errors about the fundamental business model. Some people like to shop online, from a list and with a two-day lead time, but many others have a different, opportunistic style that only works in an actual store. Bad acquisitions, over-aggressive growth targets, and bad marketing decisions sank the company.

Paul gave several other examples of this misperception of risk, calling it peculiar to American society: the Challenger accident, the Persian Gulf war, the battle of Mogadishu. Yet several in the audience, from outside the US, thought it was not just an American trait.

Making Backups Easier with Disk

W. Curtis Preston, The Storage Group

Summarized by Renuka Nayak

The take-home message of W. Curtis Preston's talk was that system administrators should back up to inexpensive disks frequently while duplicating disk backups to tape. Doing anything else might lead to situations that SUCK (a mantra that was chanted throughout the interactive talk). The presentation was well-delivered and peppered with real-life examples that Preston had encountered throughout his career.

Preston first outlined some of the advantages, disadvantages, and challenges associated with tape drives. Tapes and tape drives are high speed and low cost, which makes them good archival solutions. But tape backups take a long time, and newer, higher-speed drives are becoming more expensive. Furthermore, it is difficult to make off-site tape copies with a stand-alone drive, which needs to swap tapes. When trying to access the tape in the drive, one might run into the problem of not having the desired tape in the drive. Challenges to using tape as the only backup medium include the time it takes to make tape-to-tape copies, the rigors of regularly perfuming full backups, the limitations on writing to a single tape drive from two shared servers, and the inability to know whether a tape is in good condition until you actually need to use it.

Using inexpensive disk arrays as a primary tool in backups in addition to using tape is an excellent way to address some of the challenges presented above, Preston suggests. There are IDE/ATA-based disk arrays that are addressable via Fibre Channel, SCSI, Firewire, NFS, and CIFS and which can use RAID configurations. These units are as low as $5,000 for off-shelf varieties, and it costs as low as $2,000 to build your own. Preston recommends buying enough disk for two full backups and many incremental backups. Then connect arrays to clients or backup servers and make backups. Finally, make duplicates (note that duplicating is different from backing up) of what is on your disk to tape. One might even want to place another disk unit off-site and replicate to it. Except in catastrophic disasters, one can easily restore from disk.

Preston then went on to say why using disk is better than using tape. Disk does not require a constant stream of data and neither is there the need to multiplex, as is the case for some tape drives. He claims that if disk backups are multiplexed, then the tape copies can be easily de-multiplexed without a performance penalty. Furthermore, since disk arrays can be protected via monitored RAID, the loss of a single disk would be monitored and repaired. Making disk-to-tape copies are easier than making tape-to-tape copies, and full backups can be performed less often, saving network and CPU utilization.

So, why should we even use tape at all? Preston argues that tapes are still good for archiving purposes so that older backups can be available. Tapes are also much cheaper than disk, allowing for multiple, stable copies to be stored "on the shelf" or off-site. Furthermore, tapes are not susceptible to file-system corruption, as disks may be.

To find out more information, email Curtis Preston at *curtis@thestoragemountain.com*.

"Who Are These People?" Internet Governance, Peering, and Legislation

Paul Vixie, Internet Software Consortium

Summarized by Robert Beverly

Mr. Vixie, a self-professed "graybeard" and "member of the loyal opposition," is a long-time programmer and maintainer of BIND (a software implementa-

tion of the Internet's domain name service). Mr. Vixie's talk explored some of the changing dynamics as the Internet metamorphoses from research network to commercial network to a component of national security. The talk was timely given the recent denial-of-service attacks on the root name servers.

Because of the academic nature of the early Internet, resources were given away freely, as needed, by a loose collection of individuals. Today many of these resources have become valuable commodities. Examples include IP address space, domain names, top-level DNS domains, autonomous system numbers, and protocol numbers. These shifts have produced a variety of stakeholders, all with different motives. The talk focused repeatedly on ICANN (Internet Corporation for Assigned Names and Numbers), a government-sponsored entity. An example of a current area of conflict is ICANN's control over the top-level DNS domains. Many Internet users feel that ICANN's policies toward new top-level domains is unjust. In fact, Mr. Vixie's contention was that because ICANN is a government-sponsored entity, it tries to be all things to all people and thus fails to serve anyone.

"The Government is coming and they want to take our toys." The operation of the root servers, so-called RSOs (Root Server Operators), is a clear example of a loosely organized resource that has become part of the critical infrastructure. How one becomes an RSO is a question with no answer today. Until Dr. Jon Postel's death, he alone made the determination. The original intent was to distribute the root name servers among commercial, research, and educational entities in different countries, such that there was enough natural distrust between operators to prevent a problem. No single entity should control, or be able to control, the entire system. Specifically, no single government should be able to take over the whole system. Today the root name servers are physically in the United States, England, Japan, and Sweden. As one of the long-term participants in the health of the global DNS system, Mr. Vixie is very concerned with current politics that may circumvent the original "graybeard" policies. Despite stating that everyone should be very concerned with recent policy directions, a general sense of pessimism emerged that those with "guns and money" would eventually prevail.

A second resource at stake is IP addresses. IP address space, once abundant, is now a valuable resource. An organization may have either provider-assigned space or provider-independent space. Smaller organizations requiring fewer addresses generally must obtain addresses from their providers. Provider-assigned space allows larger service providers to aggregate the routing announcements of their customers into a single aggregate. Limiting the total number of routes in the global Internet helps maintain its health and stability. The downside to provider-assigned space is that if an organization wishes to change providers, it must forfeit its current address space (which belongs to the provider) and obtain new space from its new provider. Obtaining new space requires renumbering the IP addresses for all of the machines the organization owns. Therefore, there is a large disincentive to switch providers, giving the existing large providers a distinct competitive advantage. Currently, obtaining provider-independent addresses requires providing justification to a regional registry for a minimum-sized block. The size of this minimum allocation is determined by the members of the registry who themselves are often network operators, creating an inherent conflict of interest.

Mr. Vixie noted that while current address-space policy prevents competition in many respects, the worst-case scenario is that government would take over the registries' duties. One would then have to go to the government to obtain IP address space, much as one would go to the government to get a business license. Mr. Vixie concluded the talk by warning of the dangers of people in decision-making positions who "don't understand the impact of those decisions." He urged attendees to get involved.

Nobody Notices Until It's Broken: Self-Marketing for Sysadmins

Moderator: Lee Damon, University of Washington

Panel: Karen Ken, Dan Klein, Strata Rose Chalup

Summarized by Abiodun A. Alao

The session was devoted to why system administrators are not very popular with other staff and why their point of view is difficult to convey. They are generally perceived as overpaid with unclear job schedules. It was noted that sysadmins make the first error in introducing their role. How do you explain what you do to someone who does not have any idea what the term "sysadmin" stands for? Here are some responses: "I work with computers"; "I make the Internet run"; "I manage computers."

It is essential that you are seen as a person, a member of the team and one whose contributions are valuable in the realization of the goals of the organization. We must also make people understand what we do in concise terms and in ways that they see how we contribute to their ability to meet their tasks. Avoiding all the techno jargons will go a long way toward making us understandable and more acceptable. In the alternative we may have to teach people to speak our language (even at the risk of training them to take over from us). Finally, we must also learn the language of business since we serve the business world.

Many of us complain we do not get any respect from coworkers. Can we do a better job at marketing ourselves? While it may not always be possible to make others see things from our point of view or even understand our role, a sysadmin

who takes time to explain what we do and why will be doing a good job and will help improve our image. If people are not savvy, make use of pictures. Often we are seen as capable of providing the silver bullet to all problems. When the system fails to meet the "expectations" of the customer, we are seen as incapable. When people do not understand or lack the capability to effectively use the IT solutions provided, their tendency is to blame the IT expert, especially the system administrator.

Granted, no matter how hard you try, you can't get everyone to cooperate or appreciate what you do. For instance, how do you deal with a marketing department that has the attitude, "We've sold this product, you design it," or with your fear that the marketer is misrepresenting what you are developing?

Or what if your manager is technically savvy and brought back a piece of IT equipment from a trip. It is acceptable to tell him, "I don't try to do your job; you shouldn't try to do mine." Our response to these and similar issues is careful education so that our colleagues see the error in trespassing on territory related to IT.

Attitude is very important. Be cooperative and courteous. You can do it right and keep people around or do it wrong. Any unpleasant situation can be made even less pleasant by a negative attitude.

This is about marketing ourselves, so here are some helpful hints:

1. Marketing is about educating. Educate people around you to help them understand how our jobs are interrelated. Cultivating relationships helps in achieving this. Reveal yourself in ways other than technical; hobbies and other general interests can help us connect to other people. In addition to books, mouse and toolkit, place pictures of families, pets, etc. in your workspace.
2. Let management understand what you do, that you are not just "staring at computers." Take the manager around; draw analogies. Send in periodic status reports of major accomplishments. Even if management does not demand this, it's a good idea.
3. Get your users adequately informed by warning them about changes. Do you let them know ahead of time when bringing the system down? Do you warn them about the database server? Send mass mailings; send enough, not too much. Package your regular suggestions for users in "Useful Tips." Give users a chance for feedback. It is legitimate to occasionally ask, "Is it useful?"
4. Create opportunities and look for the next problem to be solved. Look for opportunities to make yourself valuable. That's part of self-marketing! Find a niche for yourself. When people know who you are and what you do, they will come to you. How else can you keep your reputation as a miracle worker? Do something different: for instance, publish an article.
5. Keep a good problem-tracking system. Return phone calls and reply to emails. If a problem is not resolved quickly, acknowledge and give feedback. If you cannot get a problem solved, do not blame anyone; and if you don't know, say so. "I will research it" is a good response. Be honest about your capabilities.
6. Take a vacation. Here is a cool suggestion: have some toys on your desk that could relieve tensions or put them in a box with a label that reads, "Five-minute stress relief box. Feel free to use." When you are able to get away for vacation, put things in place that make the system work while you're gone. NO "I'm going for a week; let's see how they cope" attitude.

In closing, the panel members were in agreement on the following: "Make yourself a light. Be the illuminator" (Karen Ken); "Whatever you do, own it" (Dan Klein); "You make things, that's the goal. Work as a team with a sense of duty" (Strata Rose Chalup); "Be not ashamed, but be ye not arrogant either" (Karen Ken).

Sysadmin, Stories, and Signing: Learning from Communication Experts

David Blank-Edelman, Northeastern University

Summarized by Jim Hickstein

Sysadmins have to talk to each other, and to "other species," in other fields, especially when diagnosing system and user problems. The speaker brought perspectives from two other disciplines: storytelling and interpretation (specifically in American Sign Language).

Storytelling has a long tradition in the sysadmin community, but it has an academic underpinning that most sysadmins aren't aware of. Its mastery requires application, study, and practice. Yet in 20 minutes, the speaker gave a veritable short course in storytelling, which anyone would do well to take. He used the various methods along the way, repeating the part about repetition, using silence for effect in the part about silence. (The slides are good, but they don't do justice to this performance.)

Stories are good for sequential or related events; making diverse information coherent; passing on lessons (either overtly or implicitly). They fulfill social roles, as in establishing one's membership in a community. Stories make experience reproducible and reusable, and they do so safely (i.e., with a happy ending). Stories are good for constructing layers, which the listeners can then follow, especially in complex technical situations.

I won't try to reproduce the whole course here, but the lesson was clear: If you learn to tell stories better, you will be a more effective sysadmin.

He told a story about a difficult network problem escalating through a front-line

technician via online chat. It was an oft-repeated scene of a failure to communicate. But the technology was not really the problem: The parties seemed to speak different languages (though both spoke English) and had different backgrounds and mindsets.

What was needed? An interpreter!

The speaker then went on to talk about interpretation, in general as well as how it differs from translation. Interpretation is live, and the interpreter can't go over the "source text" more than once. Generally defined, interpretation creates in the mind of the "target" the same idea that exists in the head of the "source." It is subtle and difficult, especially when no direct translations exist: The interpreter must be able to move in two cultures and make the necessary mappings between them, accurately and in real time.

In ASL, for instance, pronouns are spatial: One doesn't say, "And then he said X, and she said Y." One creates people in space, in front of the speaker, (Z is here and T is over here), and then the "saying" happens in that particular place. Another one of the many challenging aspects is shown by the difference between "leave the party" and "leave the car at home." Polysemous words, density and context mismatches, preserving register...the list goes on. Affect and intent must be conveyed: The way something is said is very important to its meaning. It is what the listener hears that matters.

Mapping this to sysadmin communication doesn't take much imagination. The potential for misinterpretation is large; SA has many "rich points." One must use feedback to detect a snag, then go back and find the knot.

He finished with a taxonomy of useless support email requests, one of which read in its entirety: "Something is wrong and I have know idea what." (The speaker's first reaction, before seeing the subject line, "Printer help," was, "Yes, I have days like that myself, sometimes.")

Perl 6: The Science of Perl, aka Studies in the Ballistic Arts

Larry Wall, Creator of Perl

Summarized by Steve Wormley

Larry started off with a brief overview of where Perl came from. Perl has roots in linguistics, computer science, art, and common sense. In addition he discussed how Perl draws from ecology, math, and golf, among other things.

Perl was described as initially a way to combine the "manipulexity" of C and the "whipuptitude" of shell in one language. And Perl was designed to continue evolving into both. One major feature of Perl is that it is designed to hide the fancy stuff. In addition Perl behaves as a natural language. Some of these aspects of the language include: you can learn as you go, you can learn something once and use it many times, there are many acceptable levels of competence, and there are multiple ways to say something.

Another important part of Perl is the culture. The Perl culture, like some others, accepts newcomers, is okay with subtribes, encourages sharing, captures knowledge, encourages cooperation, and has fun. Perl 6 started by placing a request for comments for the new language. They received 361 comments. The Perl 6 team decided to take the Winnie-the-Pooh Approach: Think Things Through Slowly. They wanted to keep everything good and throw out everything bad. The final goals for Perl 6 were simplification, power, better OO programming, better functional programming, and better pattern matching.

Some of the new features and changes include: no more double parsing, comments work better in patterns, simpler precedence rules, removal of special variables, no more parentheses on conditionals (now whitespace dependent), and blocks are now closures. Full type signatures will exist, there is a new aliasing operator, and there will be vector operators.

One new aspect of Perl is that variables will have properties, such as a compile-time property of "constant" as well as runtime properties. These properties can also be accessed as methods. There are also new smart match and smart switch statements. Explicit exception handlers will now exist (try, catch, throw). New OO support will include opaque data which must be accessed by methods, and also the possibility for multimethod dispatch (the functions called depend on the types used).

The new pattern-matching support means that patterns are no longer interpolated as strings, the use of brackets is consistent, there are no postfix modifiers (all prefix or defaults). Other features will be new modifiers, meta-syntax, full grammar support, easy parse-tree generation, and grammar inheritance.

Finally, Larry mentioned that Perl 6 will be able to use Perl 5 modules. And to create Perl 6 code there will be a Perl 5 to Perl 6 translator.

How To Write a Book with Someone You Don't Know: Internet Collaboration for the Truly Geeky

Tom Limoncelli, Lumeta; Christine Hogan, Independent Consultant

Summarized by Kuzman Ganchev

Tom and Chris started the presentation by comparing the process of writing a book to that of managing a system administration project. To write their book, they used familiar tools such as SSH, CVS, and make, and had to deal with common system administration problems: security and data integrity. Their job was more difficult in that they lived two (and later, five) time zones apart and didn't know each other. The book, *The Practice of System and Network Administration*, is divided into four parts, 32 chapters, and three appendices. The presentation focused on how they had gone about writing their book.

First, they decided on a set of standards for formatting, tools they would use, and terminology (such as "customer" vs.

"user"). Then they used a top-down approach to plan out the rest of the book.

They divided the work by splitting up the chapters between them, and specified an explicit development cycle. They used their scarce meetings to do high-interaction brainstorming, used the phone for problem-solving sessions, and organized the logistics via email. They highly recommended automating as much as possible. For example, they used CVS to automate synchronization, Perl to generate the tables, and make for pretty much everything. They used open protocols, such as SSH and LaTeX, so that they could work from any platform.

The presentation ended with some comments about writing a book. They warned that the financial rewards are not likely to be great – minimum wage is above average – and that it takes a lot of work; they devoted two years of their lives to the task. Finally, they gave some advice for aspiring authors: Interview your publisher as you would an employer, negotiate hard on contracts, and retain a lawyer.

REFEREED PAPERS

SERVICE, RISK, AND SCALE

Summarized by Jim Hickstein

Application-Aware Management of Internet Data Center Software

Alain Mayer, CenterRun

The speaker described a new product that can help manage large groups of Web servers and their related applications. The product guides the user to "capture" the essence of an existing application (for instance IIS, all relevant content, ASPs, configuration files, etc.) from a "baseline" server into a central repository. Then it can be pushed onto new servers.

The master server contains the repository and certain engines, remote agents on baseline, and "managed" servers. Any server can be baseline and/or managed. The product encourages a workflow: deploy on baseline first, tune it using existing tools, then capture.

The system embeds version control. Replacing a managed server can be done in minutes. Objects inside the system include resources of various resource types, each type having a resource handler. The handlers are deployed on the remote agents to do the capturing (pulling) or management (pushing) of applications and state. The system is extensible by adding resource types and handlers.

The field is wide open for new research in modeling, config generation, rollback, policy-based management, among other areas.

Geographically Distributed System for Catastrophic Recovery

Kevin Adams, Naval Surface Warfare Center

The speaker described a disaster-recovery system that continuously copies data to a backup data center over a wide area network, at a steady rate that is just fast enough to meet the requirement to not lose more than N hours/days of data. Constant network utilization maximizes the cost-effectiveness of using a switched WAN rather than a private line. IP quality-of-service (QoS) guarantees adequate total throughput. The minimum and maximum data rates are nearly equal.

When you can eliminate all small "single" points of failure, the entire data center becomes the new single point. High-availability (HA) solutions like local, shared resources; disaster recovery (DR) wants things separated; HA eats bandwidth; DR wants distance – bandwidth is a problem.

They wanted to copy an HA system – migrate process, data, network identity, "heartbeat" – but tried to minimize the bandwidth required. A relatively low-bandwidth pipe would also minimize the impact on the primary site. A private circuit was considered, but distance and other factors argued for using a packet-switched WAN. In this case, they wanted to maximize the link utilization, for best cost-effectiveness.

The basis of the system is point-in-time imaging (snapshot). You dribble a copy of a recent snapshot to the remote site, constantly. The snapshot interval, and thus the data rate (given a fixed size) depends on the question, "How much (new) data can we afford to lose?" If it's 24 hours, that's the cycle; you need to copy N GB per day, depending on the size of the data set for each application.

The copy uses traffic-shaping to limit the transmission rate to a fixed upper bound, and IP QoS to guarantee minimum bandwidth equal to the maximum bit-rate, to ensure completion within the cycle.

Embracing and Extending Windows 2000

Jon Finke, Rensselaer Polytechnic Institute

The speaker described a meta-directory integration project that provides all students, faculty, and staff with a single username/password for all computer system access. Rather than modifying all authentication clients to use a central server, the username and chosen password are pushed out from a central system to several different client systems, including Active Directory (AD).

The institution needed AD for students, faculty, and staff; Exchange email for staff; and password and account synchronization across all platforms. Each person should have exactly one username/password. Certain Web services would tie into it. They also wanted this mechanism to manage email addresses, so the *user@rpi.edu* alias could be directed to any of numerous internal mail systems.

Existing administrative structures are not always along department or division lines. Some groups go their own way.

But DNS is centralized; no delegation, ever.

Currently, of 2000 employees total there are about 520 who use Exchange exclusively and 150 who are "casual" Exchange users. Windows 2000 authentication spans 400 public workstations and a number of administrative Web applications; a ticket system; and so on.

The speaker presented a graph of the distribution of mail systems by division: Exchange exclusively, other mail (department), central system (POP, *user@rpi.edu*). Administrative departments are mostly on Exchange; academic departments mostly not, yet.

He then showed a diagram explaining how systems are linked during a password change. The user interacts via HTTPS; that Web server encrypts the password in a public key and stores it in a change queue in a database. The encrypted password is shortly pulled from the queue, decrypted on the password-change server using the private key, and distributed to the several authentication systems, including NT domain servers. It does not happen instantly, but password propagation times were charted. Most were under 90 seconds.

PRACTICAL THEORY

Summarized by Kuzman Ganchev

Stem: The System Administration Enabler

Uri Guttman, Stem Systems

Uri presented Stem, a framework for creating tools to automate system administration. The Stem building block is called a cell. These are written in a custom-made declarative configuration language, and are executed by a runtime daemon called a hub. Uri presented a few example Stem programs. Of course, the first one was the obligatory "Hello world," which in this case conducts a conversation by replying with a greeting. He then went on to demonstrate more complex but trivially written examples, including a remote log-monitoring program. This appends data to a remote log and log file status changes (such as creation, deletion, and truncation) to a separate file.

Stem configuration files contain the definitions of cells, hubs, and portals through which hubs communicate with each other. Stem is implemented in Perl using an entirely peer-to-peer architecture, supports modules (that administrators can write to create more complex cells), and allows encryption using SSL.

Pan: A High-Level Configuration Language

Lionel Cons and Piotr Poznanski, CERN, European Organization for Nuclear Research

Lionel Cons started his presentation by introducing the Large Hadron Collider (LHC), what will become the world's largest particle accelerator, being built by the European Organization for Nuclear Research. This facility will produce enormous amounts of data. After on-site filtering, 10 petabytes will need to be stored to tape per year. The project will require 2 petabytes of disk storage, and over 100,000 processors. Pan is designed as part of an approach to solving the incredible system administration requirements of a cluster-computing project of that size.

After this introduction, Lionel presented some principles of the system administration project, such as automation, abstraction, and the use of configuration policies. The overall structure of the system would be a loop containing four components: the cluster, a monitoring database, an "operator," and a configuration database. The monitoring database collects information about the cluster, which is then examined by the "operator" – probably a combination of automated tasks and human administrators. This then modifies Pan source code, which is compiled into XML and stored in the configuration database, from where the clients retrieve it.

The language they required needed to be high level, be declarative, avoid duplication, support powerful validation and distributed administration, and be domain neutral.

Pan stores information in a tree structure and supports template manipulation and strong validation of data. It is licensed under the "European Data Grid License," an open license. It was designed to be portable but has not yet been ported beyond its original platform – Linux on i386. Finally, the project is in its early stages; Pan is not yet being used in production.

Why Order Matters: Turing Equivalence in Automated System Administration

Steve Traugott, TerraLuna; Lance Brown, National Institute of Environmental Health Sciences

Steve Traugott presented what he calls a "theory paper." He did not go into any formalism in the presentation but instead focused more on the paper's conclusions.

Traugott argues that in many production systems, there is a tendency for system administrators to make changes by hand instead of using automation tools. He calls the resulting system state "divergent," meaning that the difference between the baseline machine state and the current state is greater than expected, making rebuilds complicated, or requiring backups for the entire operating system. A different situation that he calls "convergent" involves an automated tool synchronizing changes between hosts (hence making them closer to each other). He claims that this is an ongoing procedure, since the multiple hosts are never quite identical. Finally, a "congruent" system is one where all the hosts start out identical and all changes are performed on them using a deterministic automated and repeatable process.

Traugott concludes that maintaining a congruent system is the least-cost

method to guarantee that a host can always be restored to its working state, especially if multiple identical hosts need to be kept (for example a Web-server farm). In particular in the long run it pays to reinstall systems from scratch to bring them to an identical state rather than to work with the disparate systems. Traugott recommends a tool he helped write called isconf to deterministically automate changes to different hosts.

LOGGING AND MONITORING

Summarized by James O'Kane

A New Architecture for Managing Log Data

Adam Sah, Addamark Technologies

When you have as much log data as Yahoo! does, you need new methods to store and query it. That's why Adam Sah presented a Log Management System (LMS) called Addamark. Some of the goals of this LMS were to handle 10's and sometimes 100's of GB of data per day, parse and query arbitrary log formats, be highly available and be able to keep the original files available in compressed format. Addamark achieves this by using a cluster of machines, and an extensible SQL-like query l anguage.

MieLog: Interactive Visual Log Browser for Inspecting Log Information

Tetsuji Takada and Hideki Koike, University of Electro-Communications

When you are looking through logfile after logfile, having a tool like MieLog, c an be helpful. MieLog, presented by Tetsuji Takada and Hideki Koike, gives an in teractive visual tool that can highlight key data. An administrator can see keyw ords, periods of high log activity, or high word frequencies. Everything is color-coded so you can see at a glance if there is a problem.

Process Monitor: Detecting Events That Didn't Happen

Jon Finke, Rensselaer Polytechnic Institute

But what happens when a service doesn't run, and therefore nothing is logged? Jon Finke addresses this problem with a tool called Simon. With Simon, services l og when they last run into a database, and when a service has not checked in wit hin its configured window, a notification is sent to the administrators. For exa mple, if a service should run every 24 hours and the last time it reported in wa s more than 25 hours ago, notification is sent.

SERVICE AND NETWORK UPGRADES

Summarized by Jim Hickstein

Defining and Monitoring Service Level Agreements for Dynamic e-Business

Alexander Keller and Heiko Ludwig, IBM T.J. Watson Research Center

Alexander Keller outlined a software system that manages service level agreements, defined in such detail that a computer can automatically evaluate needs against offered services, and actual performance against guarantees, all in aid of permitting e-business suppliers and consumers to find each other dynamically. Dynamic e-business is created and dissolved on demand. For instance, a Web site's inventory, cart, and payment services might be distributed among several providers. With a dynamic system, the Web server could select, for example, payment providers on demand, based on a cost bid.

What do SLAs have to do with the daily chores of the sysadmin? In fact the sysadmin is constantly making such evaluations: What is the cost to guarantee a response time of less than one second? How much should we bill a customer for throughput of 1000 transactions per second? How much revenue is lost per hour of downtime? Can you accommodate another customer and more workload? How would this impact SLAs with other customers? SAs will become involved in this, because they have the knowledge underlying it. The speaker outlined the structure of the system, composed of SLA parameters, metrics, and functions. Some are resource metrics, others composite metrics; for instance, a function might define the peak value of a metric over a given time period. Various services (measurement, evaluation) might be delegated to third parties. The specification is flexible, using a formal language. The software package, WSTK 3.2 with SLA-compliance monitor, can be downloaded.

HotSwap – Transparent Server Fail-Over for Linux

Noel Burton-Krahn, HotSwap Network Solutions

Several techniques exist for adding fail-over capability to certain parts of a computer system, with certain limitations. But most of them don't address the problem of a failing server which has a live application state and, especially, open, long-lived TCP connections.

The speaker presented a solution for transparent fail-over of Linux servers, which preserves internal state and connections. It does this by running entire virtual servers on separate hosts, sharing a virtual IP address, synchronized in near real time over a local network. A diagram showed a typical high-availability Web application, with network load balancers distributing HTTP requests to Web servers, and these talking to a back-end database. The HTTP connections are quickly over, but the database connections tend to be long-lived, and the database server itself becomes a single point of failure. Commercial database solutions exist to make this part fault-tolerant, but they tend to be expensive, and even the front ends will occasionally show a failure to a user, when, for example, a Web browser times out. A fail-over system should never lose data; the clients should never be aware of a failure; no connections should be broken; the cost

should be low; and it should avoid forcing a rewrite of existing server processes.

Naturally, there are trade-offs. Cheap backups mean long recovery times, whereas full replication and quick recovery is expensive. The goal of this system is to replicate a server on another box, without a rewrite. It replicates the network, TCP, and internal program state, even memory, by knowing and duplicating all external stimuli coming in through trappable system calls. This assumes the server processes are deterministic, which is often true, though OpenSSL had trouble until an uninitialized memory bug was fixed. Timing-related code and direct hardware access may also break this assumption. Performance may be an issue, of course, and the network traffic between master and slave may be large. But tests so far show a reasonably good result. The author's Master's thesis was a demonstration of the system serving streaming video. On HTTPS downloads, there was about 9% degradation compared to a single server.

Over-Zealous Security Administrators Are Breaking the Internet

Richard van den Berg, Trust Factory; Phil Dibowitz, University of Southern California

Path MTU Discovery (PMTUD) is used by many TCP implementations, usually to good effect. But a growing number of sites on the Internet have overly restrictive firewall rules that block certain critical ICMP packets, resulting in whole classes of users who simply cannot see these sites. They create self-inflicted PMTUD "black holes." More than a few are security-related sites run by people who ought to know better. The authors are calling for better education on this issue and running a Web service that users can check to see if a given site is in a known black hole. Certain ICMP packets have been an avenue for some attacks, so security administrators tend to decide that all ICMP packets are dangerous and none strictly necessary. They are wrong about that: ICMP is not an optional extra. It is an essential part of IP, and filtering it out entirely will break an IP network. Some ICMP types are just more important than others. The MTU (maximum transmission unit) is the longest IP packet that will cross a given network link. For best bulk-transfer performance, two IP hosts should send each other packets that are as large as possible for the end-to-end network, but no larger. IP can fragment packets, if they exceed the local MTU at any point along the path. But the sender can set a bit, called "don't fragment" (DF), to say that a packet needing fragmentation should instead be dropped, and an error returned to the sender. This error is ICMP type 3 (unreachable) code 4 (fragmentation needed and DF set).

Path MTU Discovery works by setting DF on all the packets in a connection; if the ICMP error comes back, the MTU is reduced and a shorter packet sent. If no error comes back, the sender assumes the path MTU was large enough to accommodate packets of this size, and it proceeds. If a firewall blocks all ICMP packets returning to the sending host, such a connection will not work: The sender will time out and re-send the same, too large packet, and eventually give up. For a Web site, the user sees the connection established, but nothing ever comes out. The users most affected are those with a slightly constricted MTU, typically because their Internet connection requires a tunneling method such as GRE, PPTP, or PPPoE. Many consumer-broadband users are in this group. Their number is growing quickly. The authors have started the MSS (maximum segment size) Initiative, to try to educate those responsible for creating PMTUD black holes and to offer help in fixing them. They also list their successes and failures.

NETWORKING TRACK

Large-Scale 802.11

Tim Pozar, Late Night Software

Summarized by David Berg

Tim actually titled his presentation "Long Distance Wireless Networking Using Non-Licensed Radios." This session presented a top-down view of wireless networking on a scale larger than your average single-access-point LAN. Tim began with an overview of topologies, applications, and pros and cons of 802.11.

After laying out the basics, he moved into a more practical arena. Discussing the design of networks, he mentioned both "site surveying" and "engineering the link." "Engineering the link" covered signal loss/gain and attenuation – topics that segued nicely into his comments on hardware. Tim presented various examples, including pictures, of classes of antenna and access points. One of the more fascinating access points was the home-brew model, for which he, unfortunately, didn't provide instructions.

Pozar continued his speech with several brief remarks on security, including the forthcoming 802.11i standard. He concluded with a round-up of what we can look forward to in the 802.11 family and a list of books and Web sites of particular interest to the aspiring large-scale wireless guru.

SECURITY TRACK

Internet Security: Beyond Firewalls, Passwords, and Crypto

Peter H. Salus, Matrix NetSystems

Summarized by David Berg

Salus' presentation clued in the audience on the myriad threats that the Internet faces and that lie beyond the control of any local administrator. Peter presented the information using the analogy of a medieval fortress under siege and a wealth of graphs depicting reachablility and packet loss on the entire Internet. The discussion started with several slides on the history of worldwide Internet growth and the general state of the Internet at present.

Emphasizing the siege theme, Peter proceeded to demonstrate the effect of some of the recent viruses (April Fool's Virus) and DDoS attacks on the overall

flow of traffic across the matrix. He continued with other, perhaps less obvious, threats to IP traffic, including the severing of one of the oceanic fiber lines connecting China to the world, the 9/11 terror attack, and the bankruptcy of WorldCom. He finished the slides with the October 3, 2002, DDoS attack in which the 13 root DNS servers were attacked.

The session ended with a discussion with the audience on the possible solutions to these types of disruptions. Peter suggested that DDoS attacks might one day be prevented with an IP "early warning system." Until that day, as Peter's answer to one participant's query highlighted, the solution is active monitoring.

The Promise of Privacy

Len Sassaman, Consultant

Summarized by Martin Krafft

Len Sassaman has been involved with PGP from the early days, which puts him in a role to analyze the position of PGP and its relatives today. To sum up his talk, everyone is screaming for privacy, and yet nobody uses the tools available. Topics ranged from basic crypto to why PGP and similar products are failing.

Privacy comes in various forms: financial privacy, communication privacy, privacy of stored data. The need for privacy in all these areas is high. Modern technology poses new risks in the form of credit card fraud, ID theft, and general trust in the law to protect oneself. One of the answers to the general problem of protecting privacy is cryptography, which has seen great successes. PGP ("Pretty Good Privacy") was released in 1991, and technologies like SSL/TLS, S-MIME, and anonymizers are also still in widespread use. Consumers understand the threats, and the technologies are there, but privacy aspects of the current Internet are frightening.

The problems that Len isolates touch on almost every aspect of cryptography and related technologies. From user apathy to developer incompetence, from politically influenced decisions to the intractable problem of usability, cryptography is experiencing a number of problems as it tries to be accepted into everyday use. As an important point to back up his arguments, Len mentioned various fields in which cryptography has improved: where the user interface is simple, where there is a real need, and, last but not least, where it's actually used.

But cryptography is suffering from the problem of weak links in a chain. Unless everybody uses it, it is not going to be useful on a broad scale. A vast number of people don't use cryptography because it's not standardized, not readily available, or simply too confusing. Len questions whether it would help if the entire theory around encryption could be reduced to processes similar to sealing a letter and sending it off. He points to various attempts at making crypto easier, including PGP and TLS, as well as more high-level services like Hushmail, Zendit, and Lokmail. Most of these try to reduce the user interface to the bare minimum, with TLS being "the best" because it is opportunistic and invisible.

In conclusion, Len wants to see the technology simplified for the user. He wants friendly user interfaces, better integration, no room for individual error, and everything to be open-hooded. He wants cryptography as a standard, with the proper usage being the only usage. You are not alone, Len. Who's going to do something about it?

My Years with the NSA Red Team

Tim Nagle, TRW Systems

Summarized by Robert Beverly

Nagle spoke to a capacity crowd, underscoring the interest people have in one of the government's most secret organizations. The NSA Red Team is a group of specialized individuals whose charter is to protect information security, including voice, data, and encryption. Typically, they attempt to compromise the security of a network and the hosts within the network. Contrary to prevailing opinion, the NSA only offers this service (popularly referred to as "Red Teaming" systems) to US government networks and the networks of government contractors. Further, the NSA will probe only with the explicit request of the organization. In some cases, parts of the network that were considered critical were off-limits to the Red Team. For example, the air traffic communications at an air base were not probed. A number of people were skeptical, believing that this prepared the organization, in effect, for a not very rigorous NSA "attack." Nagle emphasized that they were working together with the organization they probed, not against them. The team never tried to exploit social engineering to compromise systems.

The NSA Red Team grew out of the 1987 Computer Security Act that divided the responsibility between NIST and the NSA. "Eligible Reviewer" was the code name for the summer 1987 DoD exercise to improve the war-readiness of government computer systems. The exercise evaluated vulnerabilities of the systems and scripted out what might have happened in the event of a malicious compromise. For instance, sending troops to the wrong location, disrupting supply chains, and so on.

Despite the prior warnings, the Red Team invariably found security holes. In order to prove compromise, the team generally left a file or some other evidence of the security hole. Every keystroke was logged to aid forensics and reproducibility. Often, the team was required to prove what they did and did not do.

Much to the chagrin of at least a few members of the audience, the talk did not discuss any technical specifics of how the Red Team compromised networks. However, Nagle provided interesting insight into the policies and procedures the NSA follows.

The Intrusion Detection Timeline

Paul Proctor, Network Flight Recorder

Summarized by Abiodun A. Alao

We are slammed on all sides – viruses, rogue insiders, employee error, software bugs, corporate spies, Web defacements, script kiddies, password crackers, network vulnerability, worms, Trojans – the list seems endless. The economic impact of malicious codes has grown exponentially to over $13 billion a year.

The number of attacks in the first three quarters of 2001 rose by over 60% compared with the entire year 2000, representing a loss of almost $380 million by corporations, government agencies, financial institutions, medical institutions and universities! And it's going to get much much worse.

The paper focuses on knowledge for selecting and employing information security technologies that are appropriate, meet organizational needs, are able to contain known risks and stated requirements, and pass a cost-benefit analysis.

Most intrusions are the result of known vulnerabilities or configuration errors where countermeasures are available; 99% of intrusions could have been prevented with patches, updated servers, etc. A direct reaction to vulnerability would be to close the window to exposures, but it is important to identify all such windows as they emerge. Making everything secured stops business and drives administrative costs through the roof. This returns us to the issues of the cost-benefit analysis of available solutions. For instance some threats may not materialize or their effects may be muffled and not as significant as anticipated. Investing huge sums to prevent such attacks may not be economical or efficient.

How then can you defend your organization? There are six major steps:

1. Analyze risk and classify resources: You have to set your enterprise-specific requirements. Identify all the source of risks and their costs in terms of potential damage to systems, loss of opportunities to do business, etc. Also estimate the value of each resource and the implications of the breach of any of them for the organization. The most critical and vital resources should get the best protection. Some questions to consider include:
   - What threats are most relevant to your business?
   - How critical is the data?
   - Where does it reside? What is its value?
   - How do you define an attack?
   - What are the technology value propositions?
2. Anticipate: It is important to anticipate potential problems by creating effective policies in the areas of security, auditing, configuration, detection, access, boundary, and application design.
3. Protect: Protect computers to reduce the threat of compromise from the inside or outside. The strength of a network or system is determined by its weakest link. Therefore, ensuring adequate protection of *all* systems on the network is essential. The following specific steps should be taken: assess computers for vulnerabilities; install latest patches regularly; use best industry practice; keep anti-virus software updated; disable Java, JavaScript in browsers; turn off macros in applications; and back up servers and workstations.
4. Detect: Prevent attacks that are known, detect attackers probing for weakness, and direct attackers into honeypots. This will make hacking more difficult and less rewarding, and may reduce the incidence of attacks. Detect network probes as attackers search for vulnerability to exploit network scans, port scans, and systematic activities. This is usually accomplished with IDS technologies for log analysis.
5. Respond: Well into the attack or shortly after an attack, forensics, and correlation will help determine what has happened or what is currently happening. Response must be timely and appropriate; that is more than enough to solve the problem and deter further attacks.

Check constantly the integrity of all files and fix problems as soon as they are detected to minimize the cost of such attacks. Review logs to reveal patterns of likely attacks. Gather evidence and apply trending and long-term analysis to determine further activity. This makes it possible for firms to anticipate attacks. Proactive firms are able to beat the attackers. Finally, it is important to report and log all attacks and attempted attacks to ensure that the organization has in place adequate data to plan with and to use for prevention.

Various technologies were examined, including system call trapping technology (Intercept, OKENA, Trojan Trap), honeypots/decoy technologies, network IDS, HIDS-Log analysis, and file integrity checkers.

Security is a process, not a destination; use the right technology for the right problem.

Slides and other security resources are available at *http://www.practicalsecurity.com*.

GURU SESSIONS

NAS: Network Attached Storage

W. Curtis Preston, The Storage Group

Summarized by Kuzman Ganchev

When I came in, the discussion had already started, and NetAppliance filers were being discussed. Essentially, the problem with these is that you have to keep the NetApp filer around as long as you want your data. Curtis gave a few examples (without names) of companies who still have to keep around archaic technology, because it's the only thing that will read their old backups, which they still use from time to time.

The discussion then moved to non-tape storage. Curtis mentioned a service at e-vault.com, for backing up a small amount of data over the Internet; this is probably best for personal data – configuration files and other compressible information. For a small office, disk-based backups can be a better solution than traditional tape. Curtis cited a backup failure of up to 40% in small office environments, because of failure to insert the next day's tape after a tape is ejected. Though taking media off-site for disaster recovery is not possible with a disk-based solution, at least the data is being backed up.

Alacritus Software, a Livermore-based company writes storage software that enables a disk-based system to act as one or more virtual tape libraries. They do not actually provide out-of-the-box solutions directly but have partnerships with third-party vendors to do so. Curtis suggested backing up to a disk-based device and then periodically duplicating those to actual tape to be taken off-site or stored in archives. This is better than backing up the device to tape, since a restore from tape only requires one operation, as opposed to two in the case of backing up the backup device.

Discussion then moved to the Quantum DX30, which are disk arrays used as backups for quick restore. According to Curtis, these are not quite as small as originally intended due to unresolved cooling issues.

Perl/Scripting Gurus

Daniel V. Klein, LoneWolf Systems; Mark-Jason Dominus, Plover Systems

Summarized by Abiodun A. Alao

Larry Wall once said, "Most of the programming out there is not done by Perl experts...they learn by experience to do better over time, and eventually they become experts." We took a step in that direction with Perl scripting gurus Dan Klein and Mark-Jason Dominus. Just type these guys' names into Google and you'll get more than you ever need to know about Perl. Doesn't get any better than that does it? Ah, but it did. A few unannounced guest gurus showed up: Matthew Barr and Larry Wall. Where else but at LISA?

We were treated to a guided tour through the coding of Larry's own home automation and monitoring setup, accompanied by many fascinating side trips into his life and personal interests: X10 problems, techniques, war stories, human-readable code. Are we asleep or awake when the thing goes bump in the night? It does make a difference, at least in Larry's home. I pity the poor mice. He didn't touch on mousetraps; perhaps that's a question for next year.

Dan Klein could not be outdone, leading to displays of several such systems. Water-flow monitoring, and why you should care. A graphical display of furnace operation related to temperature inside and outside the home. At the cabin on the lake, the water temperature at the surface and underwater. Ways not to waterproof a temperature sensor, complete with graphic descriptions of failure modes, and at least one method that works.

Mark-Jason Dominus happened to mention his Perl quiz of the week. Check it out at *http://perl.plover.com/qotw/*. A new quiz every week followed later by sample solutions. No better way to learn (except to get paid for it). This week's entry: Find all the anagrams in a list of words. And, they were off! Amazing how much can be done with a single line of Perl. Oh, forgot to mention they were to be sorted alphabetically . . . no problem. Oh, and if there are more then two words . . . and this isn't even the "expert" quiz.

Some tidbits we grabbed out of the air:

> "You can deal with unreliability in automation . . . a little bit."
>
> "How does one become a Perl guru? Volunteer a lot, try hard things, fail a lot, and learn."
>
> "Tired of chomping and putting \n at the end of every print statement? Try perl -l."

How do you top all that? Perhaps a Perl script to generate unique pattern sets for a quilt, then having your wife sew it. And convincing her it's a gift! Hmmm, don't try that at home. I guess that's why these guys are the gurus.

Email/MTAs

Eric Allman, Sendmail

Summarized by Martin Krafft

This year's guru session on MTAs and email was well attended, led by Eric Allman, author of the infamous Sendmail and current CTO of Sendmail, Inc. It wasn't a big surprise that the first questions were about spam. Eric talked about the simple anti-spam methods in Sendmail (which are still more advanced than most other MTAs), like per-host connection throttling, tweaking rule sets, milters (mail filters) and RBL, and he referenced Spamassassin. The next question concerned remaking SMTP, cleaning up its fundamental flaws as part of the anti-spam war. Eric agreed, but he stressed the extensibility of SMTP and argued against a new protocol on a new port – port 25 is the mail standard, he argued, and changing standards is near impossible: If you splinter the Net by trying to introduce a new standard, you not only create chaos for email for some period, but you also make it possible for a company that would prefer that the Net run on their proprietary standards to get a foothold. He also addressed the single fax machine problem – either everyone employs the "new SMTP" or it is as useless as a single fax machine. The next question on spam dealt with a buffer/moderation queue in Sendmail, which would allow a postmaster to intervene in case of a spam flood. Finally, a couple of technical questions about MTAs and the RFCs yielded closer inspection of RFCs 2821 and 2142 about the type of email addresses one must accept. Even though <abuse> and

<postmaster> are listed in 2142, nobody really forces users to implement them. The empty address (<>) is accepted nearly everywhere, though. Rfc-ignorant.org was mentioned.

The discussion moved to the roles of SMTP and instant messaging. Eric doesn't seem to see their technologies fusing in the future, but he recognizes that users perceive them more and more as one and the same. Eric wished that he had actually implemented SEND, SAML, and SOML (which are forms of instant messaging) in Sendmail because this would have possibly standardized IM systems from the start. The audience noted that jabber (one of the later and more successful open source IM systems) is starting to implement queueing, so maybe the technologies aren't too far apart after all.

Eric then talked a bit about the forthcoming version (8.13) of Sendmail. Among many new features, it will include LDAP support and milters per socket, but it won't interface with Berkeley DB 4.1 (even though that's being worked on with the Berkeley folks). 8.13 still has some problems with the latest Linux implementation of flock(), which doesn't behave as expected. Eric announced the "Bat Book" (O'Reilly's Sendmail book) on version 8.12 for the end of the year and said he will release 8.13 before 2003 only over his dead body – he wants the book to be current for at least a while.

Performance and scaling comparisons between various MTAs came up next. Oracle's new mail product (which uses Sendmail) is not their first attempt at this market, but previous attempts were not commercial successes, which Eric attributes to the inadequate speed of the Oracle back end for a real-time mailer application. Performance comparisons between the big UNIX mailers Sendmail, postfix, and qmail cannot really be instituted. Eric believes that qmail does way too much sync-I/O. Following up on performance and I/O, a postfix admin asked if Sendmail suffered from the same problem as postfix when it came to journaling file systems. Eric carefully tried to answer by saying that Sendmail has had good luck with journaling file systems in the past. He does not know of serious implications or dangers when running the spool on a JFS. People also asked about a mail queue residing on a solid-state disk, with which Eric has had some success. Nevertheless, he suggests not putting the entire queue on it, just the hotspots.

The last set of questions was about queue consistency and lifetime, and the ability to back up and restore the queue on a live system. While other mailers have various kinds of problems with manual intervention of the queue, Eric notes that Sendmail's queueing strategy has been reworked to avoid collision for 60 years, so even injection of restored data into a live system would not mess up the consistency of the queue. However, Eric specifically does not recommend this on production systems.

Performance Tuning

Jeff Allen, Tellme Networks

Led by Jeff Allen, author of the Cricket SNMP monitoring tool, the performance tuning session was loosely organized and consisted of specific questions as well as general problem-solving methodologies.

Allen emphasized that one should always understand data and statistics in context. As an example, WebTV engineers could not immediately offer an explanation for a drastic dip in network usage for a particular day until they discovered it coincided with the broadcast of the Super Bowl. In general, one should always form a scientific hypothesis and test that hypothesis. When analyzing statistics, averages are generally of little use since the most interesting events (and those that cause issues) are outliers. Many distributions have strong modalities or heavy tails that negate the conclusions pure averages may find. Instead, use box plots, which display the mean, minimum, maximum, and quartiles for a given data set. Box plots graphically provide much more information about the data and reveal any hidden peculiarities.

Questions focused on the typical culprits of resource contention: network interfaces and hard disks. The discussion turned to Gigabit Ethernet interfaces on Sun equipment, where the performance was sub-optimal. Many factors may contribute to this, including the packet-size distribution and various TCP parameters. Allen explained the notion of the bandwidth delay product, the ideal number of unacknowledged packets in flight. The TCP window size provides receiver-initiated congestion control. To achieve maximum link utilization, the window size must be large enough to accommodate the bandwidth delay product. Sun also has the notion of TCP high-water marks, which control the rate in which user space applications may access kernel network resources.

Finally, questions about disk performance surfaced. First, one should determine whether the problem is in fact due to an I/O-bound device. The iostat command is ideal to observe disk and controller performance. If the disks are in fact the bottleneck, data should be stripped across as many disks as necessary (often five or more). In this manner, a single datafile is divided so that a piece of the file exists on each disk. Because disk read performance is the limiting factor, each disk can now read their portion of the file in parallel and fully utilize the controller bandwidth. Even though this will waste disk space, disks are relatively inexpensive today and this technique will yield much higher performance.

WORKSHOP SERIES

SYSTEM CONFIGURATION WORKSHOP

Summarized by Will Partain and Paul Anderson

The system configuration workshop, with 22 participants herded by Paul Anderson (University of Edinburgh), built upon the cfengine workshop at LISA 2001 (*http://www.cfengine.org/Workshop/*) and the Large-Scale System Configuration workshop in Scotland (*http://homepages.inf.ed.ac.uk/dcspaul/publications/wshop/*).

Anderson led off with an introduction to system configuration: Given a large computing infrastructure (dozens to thousands of hosts), how can we describe its desired state in a humanly tractable form? How can tools (better) use such a description to control the infrastructure?

System configuration tasks span an infrastructure's whole life, including pre-installation (e.g., BIOS configuration), operating system and software install, configuration of that software, managing changes to the infrastructure over time, and taking in feedback information about the infrastructure and recovering from faults.

Problems that arise in the design of system configuration tools include handling scale, diversity, and/or change; supporting modular management so that different people can control individual aspects of an infrastructure separately; providing an explicit representation of components (separate from the components themselves); providing higher-level views of an infrastructure (e.g., viewing a cluster as a single entity); making possible the desired level of consistency across systems; and security (of course).

Solutions to these problems have to choose between static vs. dynamic configuration (e.g., JumpStart vs. cfengine); getting to a "good" state by cloning vs. by scripting; declarative vs. procedural language (more below); centralized vs. distributed control; and synchronous vs. asynchronous operation.

Though the rest of the workshop talks described particular system configuration tools, the purpose of this workshop was to study tool-independent configuration principles.

Much discussion arose from the notion (raised by Luke Kanies) that existing tools comprise an unholy *mix* of model, language (to express an instance of the model), and implementation (of the language). We will do better when these concerns can be understood independently. Points raised in this session included:

- Ideally, the language should express *what* is true in a model of a configuration (a declarative approach), not *how* to make it true (a procedural approach).
- A model should be able to represent inter-machine relationships and be independent of implementation details.
- The model needs to represent dependencies between components, including runtime temporal dependencies. ("Service X must be started before client Y tries to use it.") Temporal constraints are even more fun. ("Kernel upgrades can only be deployed across lab machines on Saturday nights, except in exam week.")
- The "truths" expressed in a model and the "truth-checking" of a monitoring system need to be closely coupled (more below).
- The model and language must support devolved management. If more than one person is specifying configuration details, how do we know the total infrastructure still "makes sense"?
- The conversation continued about the importance (or not) of "ordering" in a model; see the Traugott/Brown LISA paper for one side of the story.
- There was some discussion of whether or not a good model needs practical backing by a CPAN-like Infrastructure Framework Library (suggested by Mark Roth).

A surprising issue that emerged in the discussions was usability: System configuration tools often fail to make headway because they are too hard to use. Possible reasons for this:

- Configuration tools are complex, with a steep learning curve, especially for small sites. Better user interfaces are needed (for both GUI tools and languages).
- A tool embeds its author's notion of sysadmin policy, which proves inappropriate at any other site.
- A configuration tool is most useful when it has complete control of the system. This is a big culture change for many sysadmins.
- Existing tools are a diverse, fragmentary bunch, each one covering just a part of the problem; learning enough tools to do the whole job is a daunting task.

An idea that gained immediate acceptance by the group was that there must be a strong connection between configuration, testing, and monitoring. Specific points raised:

- Monitoring and feedback of the *actual state* are crucial. "We have to embrace failure." (Andrew Hume)
- What do we mean by "testing" a configuration? How can we test configurations before deploying them?

Our sketch of this workshop should make clear that system configuration is an intellectual and practical challenge. The conversation will continue at LISA 2003 – interested configurationists take note! Until then, details about a configuration mailing list are at *http://homepages.inf.ed.ac.uk/group/lssconf/config2002/*, along with all of the materials (e.g., slides) from this workshop.

Education and Book of Knowledge Combined Workshop

Coordinators: Geoff Halprin, Rob Kolstad, SAGE; John Sechrest

Summarized by Rob Kolstad

This year, the Education and Book of Knowledge (aka sysadmin taxonomy) groups merged their workshops in order to learn each other's working style and interests. About 18 people attended, including organizers and individual contributors from both groups. The Education group included several people who were trying to run 10- to 18-week courses and a fellow from NYU who is implementing a five-semester Master's degree course in system administration (!).

Geoff Halprin presented his brilliant BoK history and motivation. We're all solving similar problems, and we need to work and develop from the same foundation. Let's address areas of personal ygrowth, organizational maturity, and a framework upon which we can capture "best practices."

Sysadmin is about "intricacy," the interplay of components that come into play when dealing with complex environments and a continuous stream of microscopic changes. Thus, there is no such thing as "a best practice." There are a *number of best practices* that we can capture.

System administrators ensure integrity of computing systems and assist users in maximizing effectiveness of their computing environment. System administrator roles include: troubleshooter, walking encyclopedia/user manual, toolsmith, researcher and student, tech writer, both strategist and tactician (today and in the future), doctor, and counselor.

Administrator tasks, challenges, and difficulty combine with availability and hidden costs to present issues with many details. Professional development proceeds along half-a-dozen paths. Standardization is needed since people change jobs frequently (every 1.5 to 3 years), and it takes six months to adapt to a new job. We must understand the nature of a problem space by breaking the problem into its components and understanding them.

Geoff covered several related programs and listed unique features of sysadmin. He also discussed professional development and "key areas of responsibility."

The BoK seeks to define a sysadmin and development maturity model. Several examples were given. The BoK is a reference framework that supports best practices, enables effective training, and feeds certification, education, and job descriptions by listing the core skills, knowledge, and disciplines of the profession.

Rob Kolstad echoed many of these same sentiments and discussed the over 2000 elements now present in the current BoK matrix (of tasks/knowledge and the various factors that affect those tasks). Creating the document is the next step, given this list.

John Sechrest talked about the Education Committee's work. "Last year, we had a workshop, and I asked a lot of demographic questions in an effort to learn how best to serve people and enable sharing of teaching ideas." He showed his goals and discussed accreditation at his university. He gave a list of about a dozen topics and sub-topics.

Hours of lively discussion ensued, with lots of time spent delving into topics like risk assessment and change management. Teaching techniques and paradigms were discussed, including the creation of virtual laboratories. A curriculum discussion group was formed for the purposes of creating a four-year curriculum (from whence other curricula will evolve). Attendees rated the day a general "thumbs up."

AFS Workshop

Coordinators: Esther Filderman, Pittsburgh Supercomputing Center; Derrick Brashear, Carnegie Mellon University

Summarized by Garry Zacheiss

The workshop began with status reports from representatives of both Arla and OpenAFS. The current released version of Arla is 0.35.10, which supports all *BSD UNIX variants, including MacOS and Linux. An 0.36 release is expected to branch before the end of the year. This release will include Themis, their `package` utility replacement, which includes features and extensions not found in the traditional AFS `package` utility. Themis should be a drop-in replacement for `package` . Improvements in Arla 0.36 include support for incremental open and support for UUID-based callbacks (via the WhoAreYou RPC). Additionally, the afs3-callback port used by Arla will change from `7111/udp` to `7001/udp`, and XFS will be renamed to NNPFS. Windows support will also be present in this release, along with a GUI ACL manager for MacOS X that integrates with the Finder. The MacOS X ACL manager will also work with the OpenAFS MacOS X client. Future goals include implementation of a cleaner and faster kernel/userland interface, and the addition of IPv6 support for AFS. Work on integrating Kerberos 5 and GSSAPI into Rx continues.

OpenAFS recently celebrated its two-year anniversary. Recent progress in OpenAFS includes the addition of fakestat; with this feature enabled, the AFS client will provide stat information for volume mountpoints not yet traversed without contacting remote file servers. This allows the use of graphical file managers to browse `/afs` without causing excessive hangs and timeouts. This feature is present in OpenAFS 1.2.7; OpenAFS 1.2.8 will include a further refinement to only present this behavior for mountpoints to volumes in foreign cells. Other recent features include ports

to MacOS X 10.2 and an experimental port to FreeBSD, further Linux client tuning, and modifications to the file server to use Rx pings to determine if clients are reachable before allocating threads to them; this prevents asymmetric clients from consuming all available file-server threads. Issues that OpenAFS is currently facing include recent RedHat Linux kernels (which break the OpenAFS client by no longer exporting the symbol sys_call_table), the minimal RedHat AFS client, and a forthcoming HP-UX 11 port. rxkad 2b, which will add Kerberos 5 support to Rx while still using fcrypt for encryption, will appear in a future OpenAFS release, most likely OpenAFS 1.2.8.

Other discussions included:

- Porting OpenAFS to HP-UX for the Itanium and to AIX 5.1 and later
- CERT's transition from Transarc AFS to OpenAFS/Kerberos 5
- Using AFS through a firewall
- Using AFS with Kerberos 5 and a Kerberos migration kit
- Performance benchmarks and tuning
- Common user errors – Backing up AFS cells
- AFS on MacOS X 10.2
- IBM's end-of-life announcement for their AFS implementation

The workshop closed with a roundtable discussion on what AFS needs to do to gain more market share. Support for files larger than 2GB, byte-range file locking, better support for Windows clients, and more training opportunities and documentation were all cited as being desirable for AFS to gain additional market share.

Advanced Topics Workshop

Coordinators: Adam Moskowitz, Consultant; Rob Kolstad, SAGE

Summarized by Josh Simon (with help from Rob Kolstad)

The Advanced Topics workshop was once again ably hosted by Adam Moskowitz. We first discussed what percentage of our time is spent on reactive versus proactive tasks, which varied relative to how close to the end user or customer our roles were.

We next talked about the various barriers to fixing problems, including technical ones, economic problems, problems of management not understanding, and so on. Many of these issues are discussed in the forthcoming SAGE *Short Topics* booklet on budgeting.

Our next discussion was on why we reinvent the wheel by recreating the tools for the same task again and again. Reasons include ignorance of preexisting tools, political factors influencing the decision (the "not invented here" syndrome), taste, where the tool falls in the issue of specific vs. general, and changing needs.

After our discussions, we went around the room to list our favorite URLs that might be unusual as information and humor. We went through system administration aphorisms — pithy sayings such as "Never send email in anger." SAGE will be making a poster of these. (Send your favorites to *kolstad@sage.org*.) Finally, we talked about things we learned in the past year and, as usual, made our annual predictions.

WORK-IN-PROGRESS REPORTS

Summarized by Peg Schafer

The LISA '02 WIP session went very well. We had some interesting submissions! Amr Awadallah created a lot of excitement with his vMatrix presentation. However, the crowd gave the LISA '02 WIP Whip to Jeremy Mates for his "Improving Productivity" (by reading your daily cartoons) presentation.

In order of presentation here are the submitters' own descriptions.

When the Trouble Is People, Not Technology

Chuck Pervo

cpervo@jonesday.com

Sysadmins of the world unite! Are you tired of being stepped on by others? Have you ever been in a situation where there was a serious problem and you were out-shouted in the problem resolution process by an unknowledgeable person? Or when the process was directed by politics rather than solutions based on causality, data, or reason? If the answer to these or similar questions is YES, you are not alone! Alva Couch has encouraged me to do a paper on this topic, including case studies and a manual on formal problem resolution practices, which will include a Robert's Rules-style set of guidelines that should preempt such time-wasting, stressful activity.

Tivo & MacOS X

Matthew Barr

mbarr@mbarr.net

After being encouraged by some seemingly nameless party, I've been conned into doing this. So, you get to hear about it. This WIP will focus on a MacOS X machine being the recipient of a copy of data from a Tivo. It includes information on connecting a Tivo to a TCP/IP network, enabling external control of the Tivo via HTTP and Web browser, as well as how the heck to export data from a Tivo to a Mac/UNIX system. I am also involved with a collaborator (who just happens to work at Apple :) on designing a GUI system for all of this.

The vMatrix

Amr A. Awadallah

aaa@cs.stanford.edu

The vMatrix is a network of virtual machine monitors allowing for fluid server mobility between real machines. By building the servers inside of virtual

machines, we can easily move them around. The applications that we are targeting are dynamic content distribution, server switching, and warm standbys. This is research work that I am doing with Prof. Mendel Rosenblum at Stanford. More info (papers, presentations) is at *http://www.thevmatrix.com*

Taking SysNav Open Source

Christian L Pearce
pearcec@commnav.com
sysnav.commnav.com

SysNav started out as a closed source project for managing servers via a portal infrastructure. It consists of storing configuration information about machines and what components they would like managed. This information is held in LDAP and translated into cfengine files and configuration files by the middle layer. Then the back end takes these configuration files and executes them via cfengine. This framework will install, upgrade, and configure components automatically based on the information stored in LDAP. SysNav is going through a transition. It is CommNav's goal to take the back end and the middle layer and form an open source meta-project. We, at CommNav, feel the community will benefit from the project and other sub-projects that will be generated out of taking SysNav open source. Collaboration has already begun internally and will be released in 2003. Please see *http://sysnav.commnav.com* for more information.

The Configuration Monitoring and Reporting Environment

Xev Gittler
usenix-lisa@schore.org

The Configuration Monitoring and Reporting Environment (CMRE) is designed to collect configuration data from all our systems and then correlate and report on the information. This allows us to understand exactly the state of our systems, from OS levels and hardware, to software installed and patches, to security and audit problems, to standards conformance. We collect this data and save it for historical data collection (via CVS), as well as upload a significant portion to a database to do reporting across the company at various levels of detail. We also combine this with our performance monitoring to identify the most over- and under-utilized systems.

Software for Optimal Time to Patch

Adam Shostack
adam@homeport.org

Following on research presented in the refereed papers track, Adam has founded a company to build decision support software for IT departments to find the optimal time to install patches, maximizing their uptime and reliability. Adam is interested in talking to IT managers who measure uptime and security.

What?? Another &%#!'ing Backup Package?

James O'Kane
jo2y@midnightlinux.com

I'll talk briefly about why I'm writing yet another backup application and why this one will be newer, better, different. So cool, that hopefully you'll forget why you thought digital watches were a pretty neat idea.

Return of the Son of the Bride of Conserver (aka Conserver 8.0.0)

Bryan Stansell

The conserver application was developed by Tom Fine in 1990 to allow multiple users to watch a serial console at the same time. Despite its indispensability, many sysadmins aren't aware of it. Conserver can log console output, allows users to take write access of a console (one at a time), and has a variety of bells and whistles to accentuate that basic functionality. The idea is that conserver will log all your serial traffic so you can go back and review why something crashed, look at changes (if done on the console), or tie the console logs into a monitoring system. With multiuser capabilities you can work on equipment remotely, collaborate with others, and mentor junior admins. (See Fine and Romig, LISA IV Conference Proceedings, 97-100.)

Since then, many enhancements have been added. The current conserver.com version (7.2.4) also includes basic SSL support so that, assuming you have a network connection, you can securely interact with any of the equipment from home or wherever. The next version will have yet another slew of enhancements, including complete SSL support and a new config file format. In this WIP, I'll give you the scoop on the latest features and solicit you for additional cool ideas for the code and a possible future paper.

rcs.mgr

John Rowan Littell
littejo@earlham.edu
www.earlham.edu/~littejo/

rcs.mgr is a basic, self-contained configuration manager that wraps the RCS process for textual configuration files and manages their installation, including setting ownerships and permissions and running any post-installation commands necessary to activate the changes. The script has been in production for 1.5 years. Future developments will include better handling of unauthorized changes and support for per-file editors, allowing the management of non-textual files.

Infinite Scalability Distribution

Doug Hughes
doug@gblx.net

I have a multicast distribution program that has been "under development" for about two years now. It puts a sequence number on each datagram and uses selective retransmission from the receiver to the sender to get missing sequence numbers. It also uses PGP signatures on each whole "package" for authenticity and for integrity; this also allows building of a web of trust. The "file" program is used to determine how to process the received item on each

receiving host. Each "distrib item" is signed with PGP and multicasted to all listening clients on a well-defined port. Responses can be collated in many different ways: syslog, mail, tcp socket, file, etc. The software provides distribution and an extensible framework upon which to build. A distribution server can also be used as a generic request repository. A peer-to-peer network of senders and requestors can thus be built easily.

Improving Productivity

Jeremy Mates
jmates@sial.org

http://www.sial.org/code/perl/modules/

Sial::Apache::ImageShow (1.2)

The talk is available at *http://www.sial.org/talks/productivity/* with pointers to the script.

Peg's Notes: Jeremy showed true WIP spirit by developing this presentation moments before he was to go on the stage! His HUGE contribution to productivity allows users to see all their favorite daily comics on ONE page!

Administering Self-Secure Devices

A. Chris Long
aclong@ece.cmu.edu

Suppose you had a host-based and a network-based IDS on every computer in your enterprise. How would you manage them? The "self-secure devices" we are developing are disk drives and NICs that include security measures, such as monitoring for changes to system files and virus traffic. I am in the early stages of designing the user interface for a system administrator to configure, monitor, and control self-secure devices.