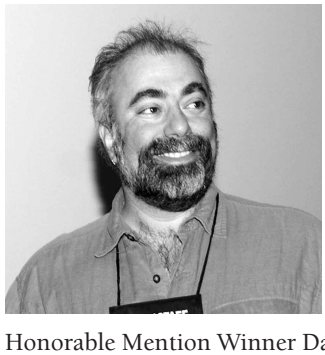
conference reports

THANKS TO OUR SUMMARIZERS

Leah Cardaci
Marc Chiarini
Chris Cooke
Beth-Lynn Eicher
Raj Kumar Gurung
Nathaniel Husted
Sumeet Jain
Kevin L. James
Robert Marmorstein
Sanjai Narain
Will Nowak
Alex Polvi
Josh Simon
Gautam Sinjaraju
Ning Wu

Best Paper Winner Melanie Rieback receiving her award from Program Chair William LeFebvre



Honorable Mention Winner Daniel Klein

LISA '06: 20th Large Installation System Administration Conference

Washington, D.C.
December 3–8, 2006

KEYNOTE ADDRESS

Hollywood's Secret War on Your NOC

Cory Doctorow, science fiction writer, co-editor of Boing Boing, and former Director of European Affairs for the EFF

Summarized by Alex Polvi (alex@polvi.net)

Cory Doctorow discussed many startling issues in his keynote about contemporary digital rights. The talk emphasized how the PC, the Internet, and crypto were once tools that helped us. Now, those tools are used to control us. The ex-EFF cohort and science fiction writer went on to discuss many issues, including the spyware in Amazon Unbox, the DVR disabling broadcast flag, and the crippling of TiVo.

The session brought out many idiosyncracies. For example, if you download music for free from Kazaa you will not get a rootkit; instead, you have to pay $15 for Sony to give you one. Or in the case of a Sony PSP, where the user hacker community is providing extra value to the device for free, Sony continues to obfuscate and defect the device.

The disheartening continued with consideration of End User License Agreements (EULAs). Today a EULA is often accepted without the user even doing anything. A concerned audience member was quick to ask what could be done as an individual and as a company. Cory responded that all EULAs must be addressed as a policy in the organization. Such policy will help raise awareness about the issue.

Cory concluded by reminding all sysadmins to make the right choice for digital freedom.

ELECTRONIC MAIL

Summarized by Will Nowak (wan@ccs.neu.edu)

■ ***Privilege Messaging: An Authorization Framework over Email Infrastructure***

Brent ByungHoon Kang, Gautam Singaraju, and Sumeet Jain, University of North Carolina at Charlotte

Brent Kang said it was his belief that email infrastructure is restricted by disk, CPU, memory, policy, and a large set of security concerns. He touched on how much time users spend sorting through email they do not want and identifying

falsified emails, such as phishing attacks. Dr Kang drew the audience's attention to the lack of an infrastructure to guarantee where a message is coming from and filtering based on that system. He drew an analogy between unauthorized email and unsigned software; allowing anyone to create a message that will appear in anyone's mailbox is akin to allowing anyone to write software and have it be trusted on any system.

The talk was not based on a new problem; rather, the idea of creating an email authorization framework has been around for some time. The solution that Dr Kang and his team created is called "Privilege Messaging" or "P-Messaging" for short. The system seems to be a cross between message authentication and a tagging structure. It layers on top of already existing email systems by adding a header to the email message. Utilizing these P-Messaging signatures, one can use them as one would tags in the "Web 2.0" sense, creating email filters and classification based on the sender's identity and "privilege-tag" information. The privilege-tag can be associated with a group, a department, or an individual, allowing for complex filtering schemes, which provide the ability to accept email on a white-list basis.

See http://isr.uncc.edu/pmessaging.

- ***Securing Electronic Mail on the National Research and Academic Network of Italy***

Roberto Cecchini, INFN, Florence; Fulvia Costa, INFN, Padua; Alberto D'Ambrosio, INFN, Turin; Domenico Diacono, INFN, Bari; Giacomo Fazio, INAF, Palermo; Antonio Forte, INFN, Rome; Matteo Genghini, IASF, Bologna; Michele Michelotto, INFN, Padua; Ombretta Pinazza, INFN, Bologna; Alfonso Sparano, University of Salerno

Alberto D'Ambrosio came to LISA on behalf of GARR, the National Research and Academic Network of Italy. As with other large sites, the GARR network faces the challenge of dealing with spam. Following a proposal in November of 2003, the SEC-MAIL team was formed to find a solution to dealing with the spike in spam delivery. The SEC-MAIL team started with a common base, Spam Assassin. The team worked to tune Spam Assassin's scoring, but it moved to work with a central Bayesian classifier. The SEC-MAIL team was seeing up to a 95% success rate with the Bayesian filtering. Mr. D'Ambrosio discussed SEC-MAIL's success by using many unique SpamAssassin plug-ins and a network of DCC servers in Italy.

Overall, the GARR SEC-MAIL team noted that the most effective tools for combating spam were Bayesian filtering, powerful DCC server networks, and greylisting. A combination of these tools, with other standard tools and best practices, led the GARR network to a steady decrease in delivered spam.

- ***A Forensic Analysis of a Distributed Two-Stage Web-Based Spam Attack***

Daniel V. Klein, LoneWolf Systems

Awarded Honorable Mention

Dan Klein presented a detailed analysis of a typical way a spammer can abuse a CGI-to-email gateway. The most interesting part of the talk involved the methods used to identify that such an attack was taking place. A series of RRDtool graphs illustrated trends that were not visible through on-demand statistics. A simple thing such as a sudden drop in spam, owing to an increase in overall volume and a high percentage of nonspam (compared to normal operation), can lead to the discovery of a small underlying issue. Klein's approach was twofold: to alert the system administration community to possible vulnerabilities that can lead to spam and to demonstrate how good reporting can lead to the discovery of such issues.

Dan emphasized that looking at your reporting technology is the most valuable way to detect these kinds of attacks. Simply looking at one log or graph would not have tipped him off as to what was happening or not happening; it was a combination of information fed to him through the comprehensive coverage of reporting. Because of this daily review he was able to notice a small (5000-message) spam attack, through his exploited Web form. There was a question from the audience of whether or not watching for short, sporadic hits to your Web form could help prevent this type of attack, to which he replied that it would not, on its own, indicate that an attack was happening. This is because short, sporadic hits to your Web site exhibit exactly the kind of behavior you expect. If you look at that data in conjunction with email volume, you can paint yourself a better picture of what is happening. Dan sent us off with an exhortation: We should check our scripts and our reporting tools.

INVITED TALK

- ***Teaching Problem Solving: You Can and You Should***

Elizabeth Zwicky, Acuitus

Summarized by Marc Chiarini (marc.chiarini@tufts.edu)

Zwicky gave an interesting and genuinely funny talk about why and how to teach problem-solving skills to system administrators. She began by speaking about the "noncontroversy" surrounding

the teaching of such skills. The general academic consensus is that excellent tutors are usually able to improve a student's skills by up to two standard deviations. This implies the possibility of pushing mediocre administrators into the highly employable "very good" category. There are of course a few barriers: People need to believe they can be taught this stuff; the system administration business selects for natural talents (i.e., not many admins get formal instruction); and, in general, although school systems know that problem solving can be taught, they don't tend to teach it. Once these hurdles are overcome, there are at least two good reasons to teach problem solving: First, people who cannot do it well tend to behave like bozos (the definition of which is left to the reader's fertile imagination!). Second, being able to solve problems naturally improves people's lives.

Throughout the talk, Zwicky offered examples of how a little bit of problem-solving skill can avoid (or could have avoided) disasters. For example, after three days of rain, a windowless third-floor machine room fills with muddy water via a blocked A/C drain. The first shift of A/C maintenance techs stem the flood by putting a plunger in the drain while a team of people work in a bucket line to empty the room out of a first-floor window. When the A/C repair shifts change at 8 a.m., a new A/C tech sees the water and the plunger and promptly removes the plunger with no forethought, creating a gigantic plume that causes damage to the ceiling and at least one machine. A bit more problem-solving skill would have led to the question, "Have I ever before seen a plunger jammed into an A/C drain, and if not, is there a good reason for it to be there?"

For most of the rest of her talk, Zwicky expounded on general problem-solving approaches as well as techniques for being a good tutor. In general, there are six steps (give or take) involved: identify the problem, analyze it, find solutions, choose a solution, implement it, and verify it.

Identifying a problem is sometimes more difficult than it sounds; one receives complaints such as "the elevators are slow" or "the Internet is broken." These are *symptoms* of a problem, not the problem itself. In the first case, once mechanical problems are ruled out, the key becomes recognizing that elevator riders have absolutely nothing to do while waiting. The *problem* is that riders are bored, and this has many good solutions. Likewise, in the second case, the symptom may be that the user cannot access cnn.com. As it turns out, the Web site itself is down, and nothing can be done from your end. No matter. The *problem* is that the users need their news (or, really, they're just bored), which again admits of many satisfactory solutions.

Analyzing a problem involves knowing what you are allowed or not allowed to do to investigate (do you have root?), and also how things look when they are working. Diagrams and guided questions are important in this phase. Don't be afraid to think "outside the computer." When you (and others) are satisfied with your analysis, it is helpful to come up with more than one solution and weigh them against each other and analyze their side-effects and long-term consequences. Again, don't hesitate to consider less costly solutions (by whatever measure) that fundamentally alter or eliminate altogether the process that led to the problem. Finally, verify your solution. Did the problem go away, and was it your considered solution that made that happen? Is the fix permanent? What would you have done differently if you knew then what you know now?

Now we know about general problem-solving approaches, so how do we teach these? Zwicky suggested several best practices: *Scaffolding* is doing the absolute minimum to allow somebody to reach a higher level than is possible without help; if done correctly, the student doesn't even notice the help. *Spotting* is the practice of being unobtrusive and letting the student make some mistakes, but catching any errors that would lead to disaster. Providing *conceptual focus* is another key; looking for repeated errors (having the wrong model or no model) that indicate misunderstanding and asking the student to verbally explain the concepts helps to keep this focus. *Praise and support* are also essential, but don't overdo it; people usually smell perfunctory back-patting. Finally, make sure to have learning environments that are as safe as possible; where practice is encouraged, mistakes go unpunished, and the teaching machines can be easily reset to a golden state. Zwicky offered the caveat that this is considerably harder to do in a workplace context.

Zwicky's references for tutoring and puzzle and problem solving can be found at the end of her online presentation at http://www.usenix.org/events/lisa06/tech/slides/zwicky.pdf.

INVITED TALK

Sysadmins, Network Managers, and Wiretap Law

Alex Muentz, Geek and Corporate Counsel, Cornerstone IT
Summarized by Nathaniel Husted (nhusted@iupui.edu)

Alex Muentz currently works as Project Manager for Onsite3. His talk covered how the various

wiretapping laws affect a system's administrator during his or her time on the job. The various laws include the 4th Amendment of the U.S. Constitution, the Wiretap and Electronic Communications Privacy Act, the Stored Communications Act, and various state laws. CALEA was also an important part of this talk, as were the pen register and trap and trace devices. He also made it be clearly known that this talk was not legal advice, that it was U.S. federal law and not state law, and that the precedents he would be discussing are not set in stone.

The first subject of the presentation was the 4th Amendment. Basically, the 4th Amendment protects against unreasonable search and seizure. This also does not affect any private actors, only public actors. In the private sector, certain states have "intrusion into seclusion" laws that can be the basis for a civil lawsuit. The current 4th Amendment view was decided in the Katz v. United States court case in 1967. This case stated that there is a privacy right. A person has the right to privacy when two conditions are met. The first condition is that society at large decides if the person has a right to privacy during an activity. The second condition is that the person thinks he or she is being private during the activity. In terms of communications, this means that a person is protected in whatever communications that he or she receives, but not in what he or she transmits. Also, information given to third parties for a delivery is not protected except in certain situations.

The second subject discussed was the Wiretap and Electronic Communications Privacy Act. This was originally enacted as Title 3 of the Omnibus Crime Control Act of 1968, then updated in 1986, and finally in 2001 by the Patriot Act. Alex also mentioned that FISA does not modify this, but he did state that nothing should interfere with the president's right to gather intelligence about foreign powers. The Wiretap and Electronic Communications Privacy Act states that interception is the acquisition of the contents of oral communications through the use of any device. The term device is not specific and can include sniffer software, laptops, or other mechanisms. Interception only happens if the information is read on the wire. Penalties include five years in prison and fines greater then $10,000 per incident. One exception to this law is that recipients may intercept their own messages. Another exception is that server provider agents, employees, and contractors may intercept communications to protect the rights or facilities of the service providers, to comply with a court order, to troubleshoot customer problems, or with the permission of the user.

The third subject discussed was the Stored Communications Act. According to the Stored Communications Act, accessing stored communications without permission or exceeding the granted permissions results in criminal penalties. This also includes someone who alters or prevents authorized access to the stored information. If the law is violated for profit, the penalties include five years in prison for the first offense and ten years for each subsequent offense. If the law is violated without a profit motive, then a person can receive up to five years in prison. Fines can also be issued by the courts. There are various exceptions, however. One exception is that the owner of a service, for any reason, can view stored material. Another exception is that providers may divulge content to the information's recipient or forward it to a third party if certain conditions are met. These conditions include a court order, inadvertent discovery of criminal evidence, or a reasonable belief that death or physical harm may fall on someone mentioned in the letters. The term *provider* is not clearly defined by the law. A basic definition is a maintainer or owner of some system that transmits electronic communications.

The fourth subject discussed was pen register devices, trap and trace devices, and customer records. Pen register devices traditionally are devices that list all phone numbers dialed along with the duration of calls from a single phone. Trap and trace devices are traditionally devices that list all the numbers that have dialed a single phone number as well as the duration of those calls. Customer records consist of names, dates, times, payment methods, and addresses (real or IP). What constitutes customer records is defined vaguely except for phone records. The only restrictions on getting pen register or trap and trace warrants are that law enforcement must show the courts that there is a legitimate need. Providers must only use informed consent, or the usage must be for billing, maintenance, testing, protection of services, or compliance with an issued wiretap order.

CALEA comes into play when discussing network design. According to CALEA, any network must allow the federal government to tap in. Alex said that the Department of Justice has stated that it must be able to do these tapes remotely without the NOC having knowledge and without the help of the network administrator.

Other interesting cases that were talked about include Steve Jackson Games v. U.S. Secret Service (1995), Garrity v. John Hancock, Muick v. Glenayre, Konop v. Hawaiian Air, IAC v. Citrin, and Councilmen v. United States. The most notable is this last case, in which the judges decided that sniffing is more than what is on the wire, but they did not define what sniffing is. Alex stated that, just to be safe, businesses should operate under the wiretapping law when reading information in their own storage. This only affects individuals in the First Federal District (New England and Puerto Rico, minus Vermont and Connecticut).

Alex said that the best way to protect yourselves is to make sure you have the user's consent in writing. He also said that it would be best to have a sniffer policy and that said policy should be clearly laid out in the employee handbook. Also, businesses need to make sure they are aware of any application that is sniffing networking traffic.

A number of questions were asked during this session, and most revolved around what is acceptable under Councilman v. United States. Other questioners also asked about CALEA. Alex continued to stress his point that businesses need to ensure that the customers are notified about what is going on. One thing Alex mentioned about CALEA is that if you do not own the keys to any encrypted data that needs to be accessed, you are not responsible for providing those keys to the authorities.

BOUNDARIES

Summarized by Ning Wu (ningwu@cs.tufts.edu)

Firewall Analysis with Policy-based Host Classification

Robert Marmorstein and Phil Kearns, The College of William and Mary

Firewall policies are susceptible to many configuration errors. The difficulty of analyzing these policies prevents their rigorous analysis. Active testing of policies may not catch all possible errors; passive testing produces output that is too unstructured to be useful.

One analysis method that avoids these problems is to classify hosts into groups by deriving equivalence classes from firewall policy. One first converts the policy into a multiway decision diagram: a directed acyclic graph in which each path through the graph represents a firewall rule. From this graph, equivalence classes can be computed that represent mail server, workstations, Web servers, etc.

Reviewing these classes for possible configuration errors allows the user to catch typos, shadowed rules, out-of-order rules, and other configuration errors. This analysis method is available via the tool ITVal at http://itval.sourceforge.net.

Secure Mobile Code Execution Service

Lap-chung Lam, Yang Yu, and Tzi-cker Chiueh, Rether Networks, Inc.

Yang Yu presented SEES (Secure Email Execution Service), a commercial system that secures the execution of mobile code that arrives at a host as an email attachment or as a downloaded Web document. Because signature scanning cannot detect any new malicious code, to protect the user the mobile code is moved to an isolated playground machine containing no valuable data and executed there.

The document containing mobile code is intercepted from Web browsers and email clients. The local version of the document is saved with a special mark, and local read operations are intercepted. After the document is sent to the playground server, a terminal is started by using remote desktop activeX control. The GDI events and bitmaps are sent back to the user's computer. The accounts on the playground machine are autogenerated, and the profiles are reset. There are also firewalls isolating the playground server.

Yang also discussed the limitations: Currently, mobile codes are intercepted from email and Web browsers only; terminal servers may be not the best choice (because of scalability and license issues).

FLAIM: A Multi-level Anonymization Framework for Computer and Network Logs

Adam Slagell, Kiran Lakkaraju, and Katherine Luo, NCSA, University of Illinois at Urbana-Champaign

Adam Slagell presented FLAIM, a multi-level anonymization framework for computer and network logs. There is an urgent demand from the research and education community to obtain real-world logs of various kinds of system events, but data owners are concerned about exposing sensitive information contained in the logs. Anonymization, defined as the process of removing information about personal and organizational identity from the logs, is absolutely necessary to protect the data owners who are willing to share.

FLAIM attempts to bridge this gap by anonymizing the data but keeping enough information in the data to make it useful to analysts. A key part of this is to define a diverse set of anonymization

algorithms and to separate format parsing modules from the core anonymization module, so that one can make policy decisions about how to anonymize without having to understand how data is parsed. FLAIM software is available at http://flaim.ncsa.uiuc.edu.

INVITED TALK

Site Reliability at Google/My First Year at Google

Tom Limoncelli, Google

Summarized by Alex Polvi (alex@polvi.net)

Tom Limoncelli gave an overview of Google's production infrastructure. Tom covered how Google does upgrades, gave a simple view of the network, and showed some of the impressive benefits of the distributed system. Upgrades are done in a multistep process. Every feature in a Google application has a corresponding flag. When a new flag is ready to be turned on, the upgraded application is put into production, with the feature flag off. The site is then tested for regressions. After the upgraded site is proven to be equivalent to the original, the feature is turned on. Emphasizing the benefits of this approach, Tom noted, "once you have good failure management, upgrades are free."

Tom also brought light to the Google WAN. Google will distribute applications across many different data centers, but not all of them. For that reason there is often a choice between sending user requests to the closest data center and using the Google backbone or simply having the user send all requests directly to the application-hosting data center. Tom also discussed some of the stats associated with distributing all data via the Google File System. For example, in 2004 their infrastructure was able to grep 1 TB of data in 100 seconds using 1800 machines. Concluding, Tom mentioned that lots of machines, lots of data, and lots of services make for a sysadmin's playpen.

INVITED TALK

Leveraging the IT Community

Patrick McGovern, VP Community and Services, Splunk

Summarized by Gautam Singaraju (gsingara@uncc.edu)

Patrick McGovern has been the director of the successful Souceforge.net Web site for five years. McGovern introduced the idea of building an IT troubleshooting community Web site. The vision behind Splunk is to develop a centralized repository for logs and IT data information. The goals as described were to provide logs and IT data for availability, security, and compliance. With the ability to share the logs, Splunk builds a community where users share and solve each other's problems. Spunk would be a centralized repository for information about all IT troubleshooting information. McGovern demonstrated that combining the viral nature of Wikipedia with the technical knowledge base from SourceForge.net would bring forth a community-based IT troubleshooting Web site for system administrators.

Started in 2001, Wikipedia, ranked as the 12th most popular Web site according to Alexia, was based on Nupedia. Wikipedia decentralized the content by using a simple wiki for uploading the contents. Following its tremendous growth, Wikipedia today holds about 1.5 million articles. The growth was attributed to the fact that Wikipedia simplified the content review process by allowing the community to collaborate.

SourceForge.net has about one-million registered users, with about one hundred thousand projects. Ranked 81st by Alexia, it has the world's largest user base and code repository. SourceForge has provided support for community-based open source development by allowing team building, collaboration, and fast development cycles. Each project can build a micro-community with a bug-tracker, ticket tracker, feature requests, mailing list, cvs, file release, and mirrors. Some of the successful projects developed at SourceForge are SugarCRM, Mailman, JBoss, SquirrelMail, and Gaim.

McGovern introduced Splunk as a technology and not as a Web site. Data centers generate a lot of logs; as a software stack, services on the machines at a data center have considerable interdependency. It is important to locate and fix the source of a problem as soon as possible. Finding the source of a problem requires many users to come together to identify the problem in the system, with 30–70% of the time spent looking through the logs. The problem in the Linux distribution is not obvious, as it installs and runs about 50 services at startup. McGovern pointed out that it is difficult to write tools to parse the data, as there are multiple log formats that change for each release of the software. The events need to index the events in real-time.

Splunk is an Ajax-based search engine for machine data. Splunk's powerful algorithm dynamically indexes data and discovers relationships among them to troubleshoot data across multiple applications, systems, and technologies. Splunk also provides all facilities for collaborative devel-

opment, just as in SourceForge. McGovern introduced Splunkbase so that the community can edit based on a wiki. Splunkbase provides an interface where people will enrich the data by introducing tagged words.

To a question about anonymization of the logs, McGovern stated that Splunk provided tools to anonymize the IP address. McGovern suggested that the logs to be put up on Splunkbase should be small, so that the community could go through them to help in troubleshooting the problem.

SECURITY

Summarized by Robert Marmorstein (rmmarm@cs.wm.edu)

Centralized Security Policy Support for Virtual Machine

Nguyen Anh Quynh, Ruo Ando, and Yoshiyasu Takefuji, Keio University

Nguyen Anh Quynh presented an architecture for implementing Mandatory Access Control (MAC) on a collection of virtual machines. This is work his team has done at Keio University in Japan. After a quick explanation of MAC and a summary of how it differs from Discretionary Access Control, Nguyen pointed out that virtual machines pose unique problems for mandatory access security policies and described a framework for addressing these problems.

Nguyen argued that implementing MAC on a collection of virtual machines (VMs) is difficult with a traditional architecture, because the security policy must be managed separately on each VM. He also discussed the challenges of collecting and correlating security logging data in a VM environment.

To address these problems, his team created VMAC, which uses shared memory to centralize logging and policy enforcement. Their approach uses a client/server model in which one VM serves as a policy manager and the other VMs retrieve the MAC policy from the server, which also coordinates security logging data. The VMAC architecture is designed to support any kind of VM and any MAC scheme. To prove the concept, they implemented VMAC using Xen with 3 MAC schemes: AppArmor, LIDS, and Trustees. In the future, they want to expand their implementation to allow other kinds of VMs and to support a broader variety of MAC policy models such as SELinux.

A Platform for RFID Security and Privacy Administration

Melanie R. Rieback, Vrije Universiteit Amsterdam; Georgi N. Gaydadjiev, Delft University of Technology; Bruno Crispo, Rutger F.H. Hofman, and Andrew S. Tanenbaum, Vrije Universiteit Amsterdam

Awarded Best Paper

The best paper of the LISA conference lived up to its high billing. According to Melanie, the proliferation of RFIDs poses several significant and dangerous privacy and security issues, ranging from identity theft to illicit tracking of passports. To address these concerns, her team has developed RFID Guardian, an embedded device for managing RFID tags and readers.

Melanie was very up-front that the Guardian could be employed for nefarious as well as benevolent uses. In addition to allowing users to identify, jam, and spoof RFID tags, the Guardian can be used to implement RFID auditing, authentication, key management, and access control. This extensive range of powerful features can be used to protect your privacy, but it could also be employed by malicious users to steal private information, launch denial of service attacks by jamming valid RFID transmissions, or circumvent anti-shoplifting measures. However, Guardian can also be used to protect your personal privacy and improve the security of RFID transactions. These benefits can be extended by employing Guardian-aware RFID readers.

The device is portable (battery-powered) and has much greater range than RFID tags. It actively transmits on the side-bands, rather than relying on power received on the carrier frequency. By generating carefully timed collisions, the Guardian provides selective RFID jamming, leaving some RFID tags accessible while blocking others.

Future versions of the Guardian will reduce the cost of the components, improve the range of the device, and potentially include a Bluetooth interface for communicating with cell phones. The question and answer period led to some lively discussion. When asked by Brian Trammel whether multiple Guardian units can interfere with each other, Melanie admitted that no such testing has been tried, but such tests are planned once the unit goes into mass production. With careful design, they may be able to make Guardians play nicely with each other. Cory Doctorow pointed out that perhaps a better solution to the problem is to get rid of RFIDs altogether. Melanie replied

that there doesn't seem to be a way to stop the proliferation of RFIDs. She argued that making Guardian widely available will make people more aware of the privacy issues surrounding the use of RFIDs.

Guardian supports both of the ISO standards for RFID. More information on the Guardian is available at http://www.rfidguardian.org.

INVITED TALK

■ ***Open Source Software and Its Role in Space Exploration***

DJ Byrne, Software Engineer, Jet Propulsion Laboratory
Summarized by Alex Polvi (alex@polvi.net)

DJ Bryne talked about open source at NASA's Jet Propulsion Lab (JPL) at Caltech. The JPL is dedicated to expanding knowledge, observing the universe, and analyzing data. DJ discussed many reasons why open source software is fit for such purposes. First, community-based bug fixes and feature additions are generally much better than those of proprietary vendors. Also, there is much higher confidence in the future of a particular software if it is community-supported. Community-based open source software is used at the JPL in its OS, software management, communications, visualization, compilers, databases, and various other places.

The JPL also releases its own open source software. The CLARAty project provides a framework for researching autonomous mobile robots. Having this software released under an open source license has helped the JPL collaborate with other researchers.

INVITED TALK

■ ***Virtualization: The Good, the Bad, the Ugly***

Mark Baum, Splunk
Summarized by Kevin L. James (kevljame@cs.iupui.edu)

This talk was given by Mark Baum, CEO of Splunk, a startup company that produces a system administrator's "search engine for IT data." He was joined by Mike Beck of the Emerging Technologies Group, the 2006 System Administrator of the Year, and Mark Cohen, Senior Technical Support Representative at Splunk.

Baum began by giving an overall view of the good, the bad, and the ugly of the current crop of virtualization technologies. The good news is that virtualization greatly simplifies the administration of many diverse systems. Existing resources are better utilized, allowing for improved and more cost-effective scalability. The bad news, according to Baum, is that resources also become more complicated and therefore harder to manage, making virtualized systems more prone to failure and performance issues. These complications are compounded when troubleshooting must take place, as pinpointing the source of failure becomes blurred: Whom do you contact, the software vendor or virtualization technical support/open source community, when your application fails? Even before virtualized resources are in place, the decision of which type and method of virtualization in which to invest can cause headaches.

Next Baum discussed several types of virtualization. One of the more recognizable types, server virtualization, aims at masking resources of servers (physical machines, processors, and host OSes) and configuration management from users. There are two methods being promoted: hosted and hypervisor. The hosted method runs as an application, making it easy to set up, while locking it to a specific platform. According to Baum, this leads to better performance for the virtualized resources. In contrast, the hypervisor model provides a leaner kernel capable of running on different architectures. Baum has found that there is currently less support for this method and performance is actually less than that of hosted mode, but of the two, hypervisor is growing faster.

Turning to Mark Cohen and Mike Beck, he asked them their opinions of server virtualization. In Beck's company, the primary benefit their exploration has turned up is that many solutions have built-in, hands-off failover support for virtual resources. According to Cohen, Splunk uses server virtualization in development, product testing, and support, as it saves money over implementing every platform supported by their product in hardware. When called upon to troubleshoot their product, problems are solved more quickly because the many possible user platforms can be emulated on a single system.

Baum's poll of the audience revealed that about one-third have implemented server virtualization in their work.

The next virtualization type presented was storage virtualization, which allows the transparent pooling of storage resources. The immediate downside to this technology is the expense of current offerings and the noticeable lack of open source implementations in this area. Mark Cohen has found

that virtualized storage is very useful when implementing business architecture; he uses it to prevent their SANs from being overrun by greedy applications. The last type discussed was network virtualization, a technology that Baum remembers was used heavily by his previous employer, Yahoo!. When dedicated routes, channeling, and finer grained control are desired, implementing them virtually makes these easier to control. According to Baum, this promising technology is still too new.

Next the talk shifted into a description of how virtualization is used at Splunk. Virtualization solutions are implemented for all supported platforms, simplifying development, quality assurance, and customer service tasks while providing great savings from having to buy equivalent hardware. Current solutions used at Splunk run on Linux, Mac OS X, and Windows. According to Baum, his people can run 16 to 20 operating systems on a single dual-core laptop. Unfortunately, there are bad and even ugly sides to all of this great promise. One problem is that although virtualized implementations are able to be scaled easily, this can hide too much information. Users have no knowledge of where their "physical sites" are located or what resources are available to them. Because the physical infrastructure behind these virtual servers is obscured, it is difficult to determine the cause of slowing or failing systems. The increased scale can also make patch management a nightmare because of the lack of tools written to work in a virtualized environment. This proves especially troublesome at Splunk, as they must constantly update their virtual systems to keep up with the machines of their users.

Another task made difficult by virtualization is backup operations. When backing up a system running several virtual ones, it is difficult to restore a single virtual platform. According to Cohen, because of how backup software often runs, it cannot be used to back up individual files within a virtualized system; instead, the entire virtual disk image must be archived. Cohen also finds troubleshooting to be difficult, because virtualization can have unwanted effects on platforms. One problem Splunk found was that time becomes skewed within virtual environments. Because the company's products deal with various system logs and files, using its software to pinpoint problems can become impossible in a virtualization setup, since time is inconsistent across the virtual platforms.

Baum's final observation is that although virtualization has been great in Splunk's development, testing, and QA shops, it is not so great for production environments. The available management tools are not robust, and automated provisioning of resources to virtual systems under heavy load is nonexistent. Therefore virtual servers often experience high utilization rates, says Baum, even when the underlying hardware is not being taxed.

THEORY

Summarized by Marc Chiarini (marc.chiarini@tufts.edu)

Specification-Enhanced Policies for Automated Management of Changes in IT Systems

Chetan Shankar, University of Illinois at Urbana-Champaign; Vanish Talwar, Subu Iyer, Yuan Chen, and Dejan Milojicić, Hewlett-Packard Laboratories; Roy Campbell, University of Illinois at Urbana-Champaign

Shankar began his talk by describing the highly heterogeneous device makeup of many modern networks and computation grids. Most of these systems are subject to frequent changes that are managed by somewhat fragile automated scripts. This management approach does not scale well and is prone to many kinds of errors, not the least of which is the untested effects of different orders of script execution. Policy-based management, in contrast, is much more scalable and tolerant of mistakes. One traditional method of implementing such a management solution is via an Event-Condition-Action (ECA) framework. In this approach, events serve as triggers that, upon certain prespecified conditions being met, initiate some kind of action (which could be corrective or just a normal operation). An example of an ECA rule is "When checkpoint store is full (*event*), if backup store is running (*condition*), assign backup store as new checkpoint store (*action*)." The drawback to ECA is that when system changes occur, rules can trigger simultaneously, resulting in a nondeterministic ordering of actions that does not achieve the desired end state. Shankar gives a simple but very convincing example of this uncertainty when ECA is used to manage a computation cluster that has failing aggregator nodes.

To address the weaknesses of traditional ECA, Shankar proposes a specification-enhanced rule framework called ECPAP (Event-Condition-Precondition-Action-Postcondition). A *precondition* is a first-order expression representing some desired partial state of the system before any action is taken. Similarly, a *postcondition* represents a desired partial state after the action is completed. While events and conditions are normally specified by the system administrator, preconditions and postconditions are designed by the *action de-*

veloper, who may be an entirely different entity. An analogy can be found in management scripts (pre/action/post) distributed by a vendor; the sysadmin may determine why (event) and when (condition) to use them. The ECPAP framework applies algorithms that can identify dependencies between conditions and build Boolean Interpreted Petri Net (BIPN) workflows. BIPNs are very useful in modeling and reasoning about concurrent action execution. A workflow is constructed by analyzing each pair of actions to determine whether one enables the other (*enablement analysis*). Following this, an appropriate ordering of rules can be enforced by a workflow execution engine, subject to one of three different enforcement semantics: *random*, which executes rule actions in random order, is the type of workflow generated in traditional ECA; *Maximum Rule* semantics guarantees that the management system enforces rules in an order that ensures as many rules are successfully enforced as possible, provided no other errors cause enforcement to fail; *All-or-None* specifies that rule actions must be executed only if all actions can eventually execute. This is accomplished by reachability analysis on the BIPN, and it provides the strongest guarantee.

Shankar went on to briefly discuss the algorithmic complexity of ECPAP's various analyses and showed some of the research team's successes in applying the framework to managing changes to HP OpenView and Ganglia performance-monitoring infrastructures.

■ ***Experience Implementing an IP Address Closure***

Ning Wu and Alva Couch, Tufts University

Ning Wu presented a prototype "IP address closure" that provides integrated DNS and DHCP services. A *closure* is a self-managing "black box" that implements and protects the functionality of one part of IT infrastructure while making its needs known to other closures. In this case, Wu and Couch propose placing such a closure on a USB device that can be used to bootstrap a network. The procedure is fairly simple when compared to other methods for maintaining IP address assignments in larger networks: The devices communicate using a simple pull-only gossiping protocol in a peer-to-peer network and are configured via a process of *seeding*. Each device is initialized by physically plugging it into the same subnet as an already seeded device. The new device discovers the existing device, clones its configuration, and receives an idea of network topology, policy, and locations of peers. It is then moved to its final location, after which it can serve to seed other closures. This approach has many advantages, including the ability to provide automatic failover and information backups, as well as to enable quick policy change propagation.

In addition to the technical details of the research, Wu also talks about the system administrator's role when interfacing with self-managing systems. The only input to the closure is a policy file (in two parts, low-level and high-level) describing the desired relationships among IP numbers, network names, MAC addresses, and subnets. This would ideally be determined (automagically in part) by a lower-level routing closure, but such a closure has not yet been built. Still, the sysadmin is relieved of managing superfluous aspects and "incidental complexity" that has no behavioral impact (e.g., having to ensure agreement between DHCP servers on the location of routing gateways). This approach also avoids common human errors during bootstrapping by, for example, automatically replicating configurations. It's important to note that human sysadmins do not become obsolete in all this. Instead, their role moves up a level, to managing policy, ensuring that the physical architecture matches policies implemented by the closures, and intervening when certain closures discover mismatches between, for example, physical or virtual subnets and desired operating characteristics.

During the Q&A session, Cat Okita asked whether the approach was just making things more complicated. Wu answered that a lot of complexity is internalized, freeing the sysadmins to fulfill their new role. Another question concerned ideas for implementing a closure for the presumed routing architecture. The answer was taken offline. Marc Chiarini commented on the necessity of a reporting facility being built into the closure magic so that system administrators start to trust the technology.

■ ***Modeling Next Generation Configuration Management Tools***

Mark Burgess, Oslo University College; Alva Couch, Tufts University

By his own admission, Dr. Couch commits sacrilege in this paper by presenting a model of configuration management without tying it to any practical tools. He tells us about aspects, closures, and promises as three essential tool-invariant pieces of the same puzzle: Paul Anderson's *aspects* model dependencies between entities; Couch's *closures* model behaviors of entities; and Mark Burgess's *promises* model interactions. Couch attempts to enlighten the audience as to how this "grand unified theory" can be applied to practical concerns

and how future tools may be built with this in mind. He wants to dispel the myth that the tools and technologies we use define the cost of config management. In fact, most IT managers know (even if they won't admit it) that it is *the way we think about the problem* that defines the cost. We currently have two ways of thinking about config management: the prescriptive approach (BCFG2, Puppet, LCFG) and the convergent approach (cfengine). The former applies a "rule with an iron fist" methodology that makes the problem too big. The latter exhibits a more laissez-faire attitude that makes everything too small. Neither of these is sufficient to tackle the entire problem, and Couch claims we are reaching the limits of our current conceptualization.

Couch goes on to explain the three pieces of the puzzle in detail: An aspect is a set of configuration parameters whose values are interdependent (e.g., all the locations in which the hostname of a machine appears). Many people understand aspects in an intrinsic way as the coordination of parameters needed to achieve a certain effect (e.g., have a running Web server). Aspects are important because they're a tool-independent way of describing interaction and complexity and they allow some approximation to the difficulty of a management task. If one can partition parameter sets into subsets in such a way that constraints between those subsets are minimal, one can then use aspects to begin to talk about desired behavior. Closures, in a mathematical sense, are deterministic mappings between configuration and behavior. One doesn't so much build closures as *find* them. For example, we can *discover* a Web service closure by identifying and controlling all aspects that determine Web service behavior. Current prescriptive tools actually forcefully create aspects, but they don't find closures, because they don't comprehensively encapsulate desired behavior. Convergent tools also end up only managing configuration as opposed to behavior; we need some way of composing closures to describe larger and larger spheres of behavior, and for that we require promises. Promise theory can be used to describe communication between closures when they are viewed as autonomous subsystems that minimally overlap.

The most important point is that the new theory provides an efficient way to talk about system behavior within the configuration management community and also to build next-generation tools to validate behavioral models (and therefore configuration) by using closures and promises.

INVITED TALK

Everything You Know About Monitoring Is Wrong

Mazda A. Marvasti, Integrien Corporation

Summarized by Robert Marmorstein (rmmarm@cs.wm.edu)

According to Mazda, recent trends in architecture, such as the proliferation of virtual machines, have provided administrators with extremely large quantities of logging and other data. Dealing with such massive amounts of data impairs the effectiveness of traditional strategies for monitoring systems and networks. The talk discussed ways to address this problem by adopting new paradigms for monitoring. Whereas some parts of the talk seemed more appropriate to a vendor BOF than to an invited talk, much of the presentation focused on new ways to think about data and monitoring.

After discussing reasons why many admins feel they need to collect more and more data, the speaker suggested that collecting "tons of data" is the wrong goal. Instead, the focus should be on getting data that has a direct business impact.

The distinction between data (something measurable) and information (something useful) formed the basis of much of the talk. Mazda argued that collecting more data is not always helpful in identifying potential failures. Often, an overwhelming amount of data makes it difficult to distinguish important events from trivial ones.

Much of the talk focused on a white paper the speaker had written on a simulated IT environment. From the data he collected, he concluded that using 40% of the available metrics is optimal. In his analysis, he found that using this proportion of the available metrics eliminated 98% of potential problems from consideration.

The remainder of the talk focused on the benefits of integrity management. Mazda advocated the use of self-learning for determining the "normal state" of the system and for predicting faults. Although this has the drawback of requiring some time for training the system, it provides better post-mortem analysis after the first occurrence of a failure and helps reduce the duration of subsequent faults by identifying problems earlier than other techniques.

Mazda also argued that using dynamic thresholds rather than static thresholds provides more accurate measurements when discovering deviations from "normal" and allows for earlier prediction of faults. He argued that monitoring solutions must deal with change and that static thresholds are in-

adequate for this task. He also argued for increased sophistication in statistical techniques.

During the Q&A period, Mazda described the learning period of his tool. It took four weeks to achieve daily resolution of events. It took nine weeks to achieve hourly resolution. He admitted that gradual changes can sneak past the self-learning paradigm and advocated using static threshold conditions and SLAs to detect these kinds of faults.

INVITED TALK

Is Entropy Winning? Drowning in the Data Tsunami

Lee Damon, Sr. Computing Specialist, University of Washington; Evan Marcus, CTO and Founder, Aardvark Technologies, Ltd

Summarized by Sumeet Jain (jain.sumeet@yahoo.com)

According to the speakers, we're drowning under a wave of data and are oblivious to it. As data space expands we will start losing track of, and thus losing, our data. Archival backups add complexity to this already confusing situation. Then we toss in security and availability issues for some spice. Where is this going, and how can we handle it in the face of millions of gigabytes of "old cruft"?

The speakers explained existing problems in data archives and then discussed some ways of solving these problems:

- Disk is cheap but the information is expensive.
- Long-term storage is easy but retrieval is difficult.
- Time is more expensive.

Many threats exist even if we store every bit of data: The storage media can wear out; media readers may not be available or can't decrypt the data; and even if everything is present there remain difficulties in finding a small piece of information in such an ocean of data.

In ancient times data was stored on media such as papyrus or rocks, which are still readable but storing on these media was hard and expensive. Then there was an era of handmade books. Entering data in these formats was easy but it involved a high cost of ownership, and few people could read or write. Johannes Gutenberg's invention of printing made it easy to publish data in the form of books but the cost of ownership was very high. In the initial stage of evolution of computers, data was stored on punch cards, which were very bulky and had limited memory. Later on, data was stored on magnetic media, which could store whole roomsful of punch cards on a few tapes. Associated with magnetic media were some new problems such as unlabeled tapes and long-term storage. As the size of storage started increasing people started keeping a lot of data backups. Today we can get 4.5 TB+ for only $7000, but how are we going to back this much data up? And long-term storage is still a big problem. The SSLI Lab has grown from less then 1 TB to over 13 TB of backed-up storage in 5 years, along with 100s of GBs of scratch space on every disk. Most data is transitory and in limbo space.

Archives have three basic functions: ingestion, preservation, and access. With ingestion many questions need to be answered: Is this the right archive for the record? Are there duplicate records? Do records need to be stored on-site or remotely? Data preservation relates to the current condition of records, environmental needs of records, ensuring that what we store is what we retrieve, and security controls for record access. Accessibility of data relates to access policies, arrangement of records, and searching for and locating the desired piece of information.

Can we say that librarians are the best people to handle our data archives? They have thousands of years of experience in data collection and cataloging. They deal with finished goods more often than us. But they have their own problems of data finding and indexing.

Several solutions are available for libraries:

http://digital.lib.washington.edu/staff.html
http://www.lockss.org/lockss/Home
http://www.contentdm.com/

Several solutions are also available for indexing, change-tracking systems, and document management systems (e.g., Google).

Lee and Evan stated that people view their short-term "being busy" state as more important then the long-term ability to recover, restore, search, and identify data. People should decide what data is important to them and how long they should keep such data. One should keep data for six months; if you don't use it in that time, throw it away.

ANALYSIS

Summarized by Ning Wu (ningwu@cs.tufts.edu)

- ***Windows XP Kernel Crash Analysis***

  Archana Ganapathi, Viji Ganapathi, and David Patterson, University of California, Berkeley

  Archana Ganapathi presented the analysis of Windows XP kernel crash data collected from volunteers who contribute to the Berkeley Open Infrastructure for Network Computing (BOINC) project. During a year, the authors overcame the challenges of collecting user data and collected over 2500 crashes. The collected data was carefully analyzed to obtain temporal patterns in the crash history. One of the goals was to determine which organizations are responsible for causing crashes.

  Analysis shows that seven organizations caused nearly 75% of the crashes in the data set. The data is also categorized based on fault type, image name, etc. Archana also reminded the audience that this result is only derived from subscribed hosts, and no information about the installed software and their frequency of usage is available. However, it is clear from the data that Microsoft is not solely responsible for crashes.

  This paper also introduces a customer-centric kernel crash analysis framework that will help users evaluate their current practice (e.g., compared to the average crash rate) and provide useful information on how to improve.

- ***SUEZ: A Distributed Safe Execution Environment for System Administration Trials***

  Doo San Sim and V. N. Venkatakrishnan, University of Illinois, Chicago

  V. N. Venkatakrishnan presented SUEZ, a distributed safe execution environment that allows an administrator to "try" new changes before they are "committed" or "aborted." Currently, the tests for new changes are tried either in the real environment or in a testbed that is constructed to be similar to the real environment. However, it is risky to try changes in the real environment and testbeds often do not reflect the real environment. Another approach is to change the operating system itself to allow testing and commitment of changes. The authors propose a distributed safe execution environment (SEE) that implements one-way isolation between the SEE and the host OS. The processes in the SEE can access the host OS; but the host OS cannot access the processes in the SEE.

  A SUEZ environment consists of host monitors and a network redirector. Host monitors use system call interposition to provide host-level isolation, and network redirectors provide network-level isolation through static or dynamic redirection of network services. The performance impact of SUEZ is carefully analyzed with several applications to show that the performance is acceptable.

- ***WinResMon: A Tool for Discovering Software Dependencies, Configuration, and Requirements in Microsoft Windows***

  Rajiv Ramnath, National University of Singapore; Sufatrio, Temasek Laboratories, National University of Singapore; Roland H. C. Yap and Wu Yongzheng, National University of Singapore

  Often system administrators feel that we need more history or dependency information when making decisions regarding management of the Microsoft Windows platform. For example, "Can I safely remove this DLL file?" or "What programs are using this registration key?" Roland Yap introduced WinResMon, a tool that can help administrators answer these questions. WinResMon does this by tracing the history of access of resources through intercepting system calls. The trace information is then generated and stored in a database.

  Accesses to resources including file, registry, network, and synchronization objects are recorded in the log database. I/O operations are not logged because of privacy concerns. The logs can be queried with a customized query analyzer by using a customized query API à la SQL. The log database can also be maintained through log APIs. The overhead of WinResMon was analyzed using micro-benchmarks and the results show that it is comparable to that of other tools. The volume of data is also acceptable; Roland described that, in his environment, the analysis rate of raw data before compression could be as high as 18 MB per hour.

INVITED TALK

- ***Perfect Data in an Imperfect World***

  Daniel V. Klein, Consultant

  Summarized by Leah Cardaci (lcardaci@cs.iupui.edu)

  Dan Klein related the current increase in the collection and preservation of data and the implications of this change. Klein first looked at the impact of long-term data retention. He illustrated why this is a problem using the example of trying to explain his college years to his hypothetical children. One solution to this problem is personal digital rights management, which would allow control of how long the data could be used, who could use it, and how it could be used. He went

on to provide other examples of how retention and dissemination of information about someone's past could be damaging.

Klein next discussed how to handle the current state of data exposure. On an individual level, you can protect your personal information by not doing that which is illegal, not foolishly publishing incriminating information, not publishing anything you'll regret later, caring what others think, and not sabotaging the collection of data. On a societal level, you can isolate your information, legislate for privacy protection, mitigate the impact of data retention, and understand how privacy can be threatened.

In addition to the long-term retention of data, another problem is the widespread collection of data. Data is collected in credit card logs, RFIDs, and loyalty programs. In addition, such information can potentially be misused. This is compounded by the fact that people are often willing to trade privacy for convenience. Many collection practices have a potential good use, but they also can be used to violate privacy. This brings up the question of whether data must always be stored completely to serve the purpose for which it is preserved. In some cases, it may be desirable to store data completely in the long term, in some cases it is the trend in the data that is important, and in some cases it may desirable to destroy the data after a given time. However, it is not always obvious what complete, long-term data might be useful to others.

Besides the abundance of data and the lack of control that individuals have over their data, there are also problems associated with everyday attitudes toward digital information. There is a tendency to blindly believe in data, without considering the fact that it can be incorrect. There is also a tendency to abandon established social niceties when dealing with electronic information about friends, employees, and acquaintances.

People have no idea what data about them is being made public, and how it can be used to track them. The use of proprietary data formats makes it impossible to tell what data is actually being recorded and shared. This includes the embedding of camera type in images from digital cameras. One way of controlling information when you are the one publishing it is to use techniques such as not sharing the information digitally, cryptography, steganonography, and shared secrets. Another option is to use a personal privacy statement and choose what information to disclose based on the privacy statement of the third party. However, these precautions cannot mitigate the fact that a person is not the only one who controls what information about him or her is being made available. It can be difficult to use information for good purposes without inadvertently violating someone's privacy.

There is a tendency to believe a perceived authority without thought, and the Web can be perceived as an authority. In addition, people tend to trust themselves, despite their own abilities to make mistakes. These problems could be handled by verifying the correctness of data, but that is not a common practice.

Overall, technology can be used for both good and bad purposes and the abuse of technology is common. Klein suggests that information handling "makes it easy to be good" and "makes it hard to be perverse." As an example of this type of information-handling system, he mentioned an ancient contract system that involved an inner, unmodifiable contract inside a publicly visible copy of the contract. He promoted the use of open source code and open standards to ensure that both the amount and the nature of data published are visible.

INVITED TALK

QA and the System Administrator

Adam Haberlach, Google

Summarized by Will Nowak (wan@ccs.neu.edu)

Haberlach's talk focused on the questions "What is quality assurance (QA)?" and "How does QA fit in?" As "Internal Systems QA" scaled at Google a dedicated "Operations QA" team was spawned. Adam encouraged the audience to look into the question of what QA is, and what it is not, giving definition to the scope of his work.

He gave some general examples of QA at Google, then drilled down to illustrate eight use cases for the Systems Operations and Network Operations groups. Key points were in the performance testing of LDAP directory services, to ensure that global performance is up to par. Another role of the group was to test the desktop platforms, to ensure that each hardware platform globally, performed as expected with the software applications engineers need to do their daily jobs. Adam mentioned that a GUI-focus-based UI testing tool, Eggplant (http://www.redstonesoftware.com), was effective in ensuring that Windows machines behaved as expected. Simple repetitive tests help to make sure that new changes to an OS environment do not break core user functionality.

The end of the talk focused more on traditional software QA, ensuring that internal applications have unit tests and that developers use their own software. Adam had some tips for QA teams to fit into the big picture, making sure that QA is committed to making software and services for customers and clients. Getting started in QA is easiest if you can fit into a large, long-term project and sell yourself via viral marketing. In the end, the Operations Quality Assurance team at Google helps to ensure that the software and services that the Operations group oversees run smoothly on a day-to-day basis.

SYSTEMS AND NETWORK MANAGEMENT

Summarized by Will Nowak (wan@ccs.neu.edu)

LiveOps: Systems Management as a Service

Chad Verbowski, Microsoft Research; Juhan Lee and Xiaogang Liu, Microsoft MSN; Roussi Roussev, Florida Institute of Technology; Yi-Min Wang, Microsoft Research

Chad Verbowski presented several new approaches to handling the management of large networks of Windows machines. Often, with a large site, you cannot keep track of every change made on a machine to figure out which single change impacted the system. The LiveOps system inserts itself as a kernel-level driver to passively monitor what is happening on the machine, keeping logs of transactions happening on that machine. It is possible, with low overhead, to monitor what process forked another, at what time, and by what user. Chad presented the example of discovering how eMusic was installed on an MSN server. By backtracking through some intuitive Web interfaces, he could see that the eMusic installer was launched from a Winamp Media Player installer, which was launched by Microsoft Internet Explorer.

Chad highlighted another example, one of installing service packs on a large number of machines. It is hard to determine whether the service pack has done everything it was intended to do. By utilizing a "Stale Files" feature of the LiveOps service, one is easily able to see that a subset of the total affected machines did not get properly updated, owing to the lack of a reboot. The LiveOps service attempts to help administrators track the number one cause of system administration problems: unexpected user changes. The reporting features present in the LiveOps system enable the administrators to discover the relationships between changes and events on their systems.

Managing Large Networks of Virtual Machines

Kyrre Begnum, Oslo University College

Kyrre Begnum made a key point to his audience: Setting up a complex virtual machine environment for a class or lab is difficult and arduous. Kyrre introduced a tool developed at Oslo University College called MLN (Managing Large Networks). MLN allows for the use of a simple configuration syntax to build reasonably complex virtualized scenarios. These scenarios, or projects, can consist of many virtual machines and switches, and they can be used with UML or Xen. The configuration syntax allows for the creation of superclasses and is extensible through plug-ins. Illustrated was a sample plug-in, one that would autoenumerate parameters such as an IP address for a large quantity of hosts.

Kyrre gave two prerecorded demos for the audience: One demonstrated the creation of a virtual machine LAN from a configuration file, and the other was a demonstration of cold migration from one Xen host to another. The MLN tool recognizes the project as a whole and will migrate everything needed to run the project on another host, by changing only one parameter. MLN is a cool tool to work with large virtual networks, enabling one to think more about the key factors involved and less about how to accomplish them.

More information on MLN can be obtained from http://mln.sf.net/.

Directing Change Using Bcfg2

Narayan Desai, Rick Bradshaw, and Cory Lueninghoener, Argonne National Laboratory

Narayan Desai presented a interesting paper about change management. Using the bcfg2 tool, developed at Argonne National Laboratory (ANL), Narayan and his team applied supplemental features to allow for integrated change management. A Subversion repository backend was added to bcfg2, and the server was modified to allow a site to pick any revision available in the repository, not just the version available in the HEAD revision. The team also chose to modify the client side, so that they could track reporting information alongside the associated revision.

The speaker covered some theory behind how changes are made and applied. The scenario described involved coordinating changes to the configuration repository to indicate current and future applied configurations, then mapping out a schedule for those changes to be applied, because of the ordered approach that configuration takes. Change management is not a new idea, but the

implementation that ANL provides helps administrators take advantage of a tight integration between change management and configuration management.

More information on bcfg2 can be found at http://trac.mcs.anl.gov/projects/bcfg2/.

INVITED TALK

High Availability: From Luxury to Commonplace Necessity in 10 Years

Eric Hennessey, Group Technical Product Manager, Symantec Corp.

Summarized by Ning Wu (ningwu@cs.tufts.edu)

Eric Hennessey reviewed the history of high availability (HA) and how HA solutions evolved from life before HA, to server-centric HA, and then to application-centric HA. In the "old days" HA was implemented via standby servers that shared the same storage and provided failover capability. After improvements in storage technology, the "N+1 architecture" became popular: These involved N applications and one standby server. When one application failed, the application was moved to the standby server. Later, the "N-to-N architecture" emerged, in which there was no spare machine and excess capacity on each server was used to provide failover. Eric also talked about data replication (DR) technology, which has become an integral component of local HA and also provides failover across wide area networks.

Currently, HA solutions are application-centric. Applications can now run on virtually any machine with access to appropriate storage. The challenges this brings include server proliferation (more and more servers) and decreased server utilization (with some using only 15%). Applications have also become more and more complex; layered structure brings more dependencies. As IT provides more services, customers are demanding more. Customers want HA for more applications. A Gartner report shows that 58% of applications are considered critical, while an informal survey shows that 5%–10% are protected by HA. The main reason for the gap is cost.

Facing increased complexity and higher SLA requirements, server proliferation, and limited staff and budget, we need integrated solutions. In the next few years, through comprehensive data center automation (configuration management, provisioning management, and application management), high availability will become a matter of routine, not exception, and each application will get as much HA as it needs.

INVITED TALK

What Do You Mean, Identity 2.0?

Cat Okita, Earthworks

Summarized by Leah Cardaci (lcardaci@cs.iupui.edu)

Cat Okita provided an introduction to the concept of Identity 2.0, the motivations behind it, and the current state of the Identity 2.0 movement. Okita began by discussing general identity management concepts and history. A digital identity is defined as a collection of claims attached to a digital subject. She related four standard ways to treat identity management: the traditional user account method, the per-environment centralization of user accounts, the data management view, and the marketing view.

Okita discussed the different suggested properties of an identity management system and went on to relate her own recommended properties. An identity management system should be:

1. Minimal, designed for selective rather than permissive sharing of aspects of identity.
2. Verifiable, providing a means to verify assertions made by a subject.
3. Unlinkable, preventing the ability to take one aspect of a person's digital identity and link to other aspects of that person's digital identity.
4. Usable, making it something that will actually be used.

In order to provide users with control over their data, an identity management system must be designed to provide anonymity by having a default deny policy for sharing information about someone. The system can then choose who is allowed to access what information about one's digital identity. This is critical because, once it is shared, information cannot be made private again.

Problems with identity management involve the inability to know who has your information, what is being done with the information, and how it is being shared.

Identity 2.0 is an identity management scheme designed to allow individuals to control the aspects of their digital identity by limiting how it is shared. Identity 2.0 involves three key controlling entities: a digital subject, a relying party, and an identity provider. A digital subject has multiple digital identities, which are stored by an identity provider. When digital subjects want to interact with a relying party, they can choose what digital identity they want to use, based on what type of credentials the relying party accepts. They will

then send those credentials to the relying party, and the relying party will select the appropriate identity provider to confirm that the credentials are valid.

Okita then provided an overview of the current state of Identity 2.0. She mentioned important players in Identity 2.0, including standards, protocols, frameworks, and Web applications related to the movement. She then discussed the current areas of development in the movement.

Identity 2.0 looks to solve various identity management problems. Individual concerns include the management of many multiple identities, keeping track of the associated passwords, and controlling the flow of information. For those responsible for the digital identities of others, Identity 2.0 can ease the process of managing the information, sharing it within the organization and between organizations, and meeting security audit compliance. For commercial interests, it facilitates sales, helps to track habits, and promotes customer confidence. For the government, it helps reduce complexity and improve manageability.

Okita concluded that the Identity 2.0 movement is developing in several promising areas. However, progress remains to be made if it is to meet its goals.

VISUALIZATION

Summarized by Robert Marmorstein (rmmarm@cs.wm.edu)

NAF: The NetSA Aggregated Flow Tool Suite

Brian Trammell, CERT/NetSA Carnegie Mellon University; Carrie Gates, CA Labs

Brian Trammell presented a tool for aggregating and displaying network flows. The tool he and Carrie have developed is like a Swiss army knife: It has tools for handling many different kinds of inputs (including IPFIX, Argus, Silk, and even pcap!) and provides a wide variety of filtering and analysis operations.

The tool consists of three utilities which together provide a comprehensive netflow aggregation suite. The "nafilize" utility allows the user to apply aggregation and filtering expressions to a set of flows. The "nafscii" utility converts the binary output of the other utilities into a human-readable format. The binary output can also be converted into a graphical plot of the aggregation data. The "nafilter" utility is a lightweight filtering component with no aggregation capability.

In addition to aggregation and filtering, these utilities allow the user to sort on any key or value field. They also provide a "top-N" listing feature that can, for instance, show you the top 15 most common source addresses of all SSH packets sent in the last hour. The tool can manipulate either unidirectional or bidirectional flows and can even combine related unidirectional flows into bidirectional flows.

The tool is available from http://tools.netsa.cert.org.

Interactive Network Management Visualization with SVG and AJAX

Athanasios Douitsis and Dimitrios Kalogeras, National Technical University of Athens, Greece

This paper focused on ways to allow administrators to create network-related visualizations without suffering through the complexities of modern graphical APIs. The framework that Athanasios presented provides a simple but flexible API for depicting important network data. Using this framework, developers can create interactive visualizations for observing and managing the network.

The framework is designed to be modular, interactive, and reasonably secure. It provides functions for displaying and manipulating both unidirectional and bidirectional graphs. The framework uses a client/server architecture. Data is collected by a management server, which formats it as a set of XML documents and transmits it to a Javascript client, which renders it by using scalable vector graphics.

The tool is not yet available, but it will be released when it is considered stable enough for production use.

Bridging the Host-Network Divide: Survey, Taxonomy, and Solution

Glenn A. Fink and Vyas Duggirala, Virginia Polytechnic Institute and State University; Ricardo Correa, University of Pennsylvania; Chris North, Virginia Polytechnic Institute and State University

Glenn presented HoNe, a network visualization tool he developed as part of his dissertation research. Unlike existing tools, HoNe can correlate network connections with processes on the sending or receiving host. This makes the tool particularly useful for visualizing security-related information.

The main display window categorizes hosts into categories based on whether they are inside or

outside the enterprise and whether they are "hosts we manage" or "hosts someone else manages." Other windows provide connection filtering and display time-related data about currently selected connections.

Glenn also described the challenges in obtaining data about the relationships between processes and network connections. After trying to obtain this information using various userspace tools, he finally decided that modifying the kernel was the only effective and accurate solution. A usability study of the tool found that the packet-process correlation was a novel and helpful instrument for both novice and expert users.

INVITED TALK

The Last, Best Hope: Sysadmins and DBAs as the Last Guardians of Privacy

Danny O'Brien, Activism Coordinator, Electronic Frontier Foundation

Summarized by Leah Cardaci (lcardaci@cs.iupui.edu)

Danny O'Brian began by discussing his organization. He briefly covered the EFF's activities, funding, makeup, and goals. He mentioned three areas in which the EFF is involved: technical research, legal representation, and publicity and advocate work.

O'Brian then moved on to cover the need to update constitutional rights over time and the difficulties involved in this process. In particular, he focused on the 4th Amendment and the need to update the amendment to reflect new technology. To illustrate the typical slow pace and intricacies involved in such a revision, he detailed how the 4th Amendment was applied to conversations over the telephone. In 1927, the Supreme Court ruled that the 4th Amendment did not apply if no physical trespass occurred and the items gathered were not tangible. This was slowly changed, as the technology advanced and various cases began to show the problems in that decision. Eventually, the court did decide that the amendment applied to people and not places. However, it did not protect information given to a third party and then shared with the government.

The modern way of life involves private information being shared in way it would not be in the past, which requires a change to strengthen the constitutional protection of privacy. The way the data is stored should not affect the privacy rights accorded to that data, but that is the current impact of the third-party exception. To handle this problem, the EFF advises courts and judges about the need for change, advises users about how the current laws can affect them, and advises companies about the implications of their actions under the current laws.

O'Brian went on to discuss how system administrators can help this process. There are three aspects involved in the development of a civil liberties law: law, running code, and culture. One area of the culture that could be adjusted to better support privacy is the idea of logging by default. Instead, it would be preferable to consider when, what, and how much actually needs to be logged to serve the desired purpose. One current change to system administrator code is to use data storage techniques that will restrict the use of the data to its original use. An example of this is in the book *Translucent Databases,* by Peter Wayner, which describes mechanisms to protect the stored data from being used in any other way than its original purpose. This involves encryption, ignorance, minimization, misdirection, stunt data, equivalence, and quantization.

The use of these techniques can be supported by pointing out how it would be in the company's best interests. This includes the need to follow privacy policies that state that data will only be used in one way. Another factor is reducing the cost of trying to discover data when it is used in a lawsuit. These approaches can also help to avoid the issue of the company becoming a target of government agencies because of the amount of data stored. Finally, there is the ability to reduce the cost of storing the information.

The ultimate goal is a change in the law. However, the changes in code and culture are important because judges tend to look at existing practices when interpreting the law. As these changes are made in the realm of technology, the idea of the importance of privacy of electronic data will spread to the larger culture. This will help to change the law, as judges try to reflect the existing culture when deciding the meaning of the law.

O'Brian was asked about the Electronic Frontier Foundation's degree of collaboration with the ACLU and organizations such as moveon.org. He replied that they work closely with the ACLU. He added that the EFF is nonpartisan and tends to work with organizations on both sides of the political aisle.

INVITED TALK

Command and Control: System Administration at U.S. Central Command

Andrew Seely, Global Command and Control System Lead, HQUSCENTCOM-J6/Northrop Grumman Defense Mission Systems

Summarized by Kevin L. James (kevljame@cs.iupui.edu)

Andrew Seely is a system administrator for U.S. Central Command (CENTCOM), the Command and Control (C2) arm of the U.S. Military tasked with operations in the Middle East and surrounding areas. He set out to both show how his job is similar to system administration in other industries and how this unique niche sometimes requires unorthodox problem solving.

CENTCOM is responsible for managing and distributing information concerning force deployments and the resources needed to support these forces and protect them. The planning, logistics, and intelligence of missions are also its responsibility for not just war but also rescue and disaster relief efforts. To deliver these services, CENTCOM provides information to the Global Command and Control System, GCCS-J, a family of systems facilitating everything from planning and intelligence gathering to the tracking of global readiness and ballistic missiles. GCCS is also used by coalition partners and is being integrated into many homeland security organizations.

Interoperability is the system's strength, as GCCS is able to integrate raw data from many diverse sources and present it in a cohesive manner. But, according to Seely, this comes at a cost: glacial technology migration, many levels of bureaucracy to gain required approval, and months (even years) of testing. For instance, he expects that GCCS will be using Windows XP by 2007, though most likely later. Thus, CENTCOM's C2 capability is made up of systems considered by industry to be outdated, even ancient, because stability is paramount.

To accomplish these goals, C2 system administration requires constant vigilance to maintain reliability and flexibility to accommodate ever-changing requirements in order to fulfill a much broader mission. Although these requirements may sound familiar, the unique "tactical conditions" of performing these duties, which include live battlefields, present interesting challenges. Among those Seely described were power, communications, and resource shortages caused by mortars and adverse weather. These were coupled with locally hired staff who often do not speak the same language and the high turnover resulting from personnel rotation. In addition, the systems and applications that he supports are not chosen or even configured by him when he receives them. Everything comes through the Defense Information Systems Agency (DISA), a government clearinghouse that vets and tests applications and preconfigures machines to exact specifications and allowances. Many of these come from contractors working on classified contracts, so support is difficult to obtain, if even possible, and thus they are virtual black boxes that operate without regard for other applications or systems. Yet all must be integrated into this monolithic system for the continued success of the mission. All in all, Seely said, the most important requirement is that "a wide range of expertise is needed at a moment's notice: You have to be sharp and learn fast."

After describing the environment, Seely gave examples of problems he has faced and steps taken to solve them. One such problem involved an attempt to save setup time on the installation of two new machines, each of which, because of accreditation requirements, requires a specially tasked team two weeks to set up. Another wrinkle was that one machine was at CENTCOM Headquarters in Tampa and the other was in Qatar, a small country in the Arabian Gulf. An attempt to simplify the setup of one machine, make a backup, and then restore the backup on the other machine was balked when the tape containing the backup never arrived in Qatar. Communications between Tampa and Qatar were shaky, so the entire backup couldn't be sent by FTP. Consequently, everything was sent file by file, but larger files failed repeatedly. Because of the tight configuration control required by the government, simple tools such as compilers and utilities such as split are unavailable on his systems. To solve the problem, Seely decided to implement split and cat functionality himself, in Perl.

Inventive solutions such as this are often the rule in C2 environments because of the controls placed on those working in them. To solve a problem, the solution not only has to work but must be built with the few tools available to the administrator. Although he acknowledges the obvious problems with this, Andrew's approach is to assume crisis to be the norm because, as in all our jobs, often it is.

Black Ops 2006: Pattern Recognition

Dan Kaminsky, Doxpara Research

Summarized by Nathaniel Husted (nhusted@iupui.edu)

Dan has spoken at the Black Hat Briefings for the past six years and is also the coauthor of several books. He is also a member of the "blue hat hackers" Vista audit. Dan discussed various topics, including network neutrality and how to detect that it is being violated, a better way for users to recognize SSH keys, finding structures in hex dump files for use in fuzzing, visually showing file structures at a binary level, and flaws in SSL communications.

The first subject discussed was determining when network neutrality was being violated. Dan suggested that exploiting a behavior in the TCP protocol was a useful way to determine when service providers slow various packets down. The specific TCP behavior being exploited is the protocol dropping extra packets when the channel is saturated with information. Based on this behavior, Dan suggested that if someone sends 100 KB on one channel, one can tell what link is causing a drop in speed based on dropped packets. By spoofing source IPs, a person can then determine which providers are acceptable, and which providers are not. One can also spoof various payloads to determine what content is acceptable, and what is not. Other protocols useful for this purpose include RTP and RTCP.

The second subject Dan discussed is the weakness in SSH key validation by the user. Every time a user sees an SSH key she or he doesn't recognize, it should not be accepted. Generally, this doesn't happen. To improve the user's ability to recognize SSH keys that belong to the user's servers, Dan suggested replacing the hex values with names from the U.S. Census report. Five names provide the same amount of entropy as the hex keys that are normally used. Dan also mentioned some other methods that have been thought of, including using abstract art and people's faces.

The third subject discussed was how to make hex dumps more usable for fuzzing by finding structures in the files. Dan stated that hex is generally hard for people to read. This makes it especially troublesome during a process called fuzzing. This is when a user tries to input various types of data to make a program do something that isn't desired. The two types of fuzzing include smart fuzzing and dumb fuzzing. Smart fuzzing is when a user requires knowledge of the underlying structure of a system and then has a specific attack to exploit that system. Dumb fuzzing is when the user will input random data into areas that he or she thinks might cause the system to stop working. To improve the dumb fuzzing process, Dan suggested using the Sequitur function to determine and highlight any structures in a hex dump. This technique was based on a paper by Craig Neville-Manning written during his Ph.D. research. The major benefit to using this technique is that it scales very nicely, even if it is not the best way to generate grammar.

The visualization technique Dan talked about next was somewhat related to the Sequitur function. This technique is based upon Jonathan Foote's paper "Visualizing Music and Audio Using Self-Similarity," as well as "DotPlot Patterns: A Literal Look at Pattern Languages," by Jonathan Helmans. This visualization technique uses white, black, and grey pixels to determine whether various file bits are similar, dissimilar, or semi-similar. This technique then creates a patterned image with an equality line running diagonal from the top left to the bottom right of the image file it produces. What is special about this technique, Dan says, is that it is actually useful. When using the equality line as a reference, anyone analyzing the file can then pinpoint where exactly in the file a point in the image occurs. Color was also developed for this application based on suggestions from http://colorbrewer.org.

In a last-minute addition, Dan also discussed flaws in various SSL implementations. The first thing Dan suggested was not to put the same SSL key on different boxes. Also, he suggested that if you want to keep some DNS names secret, you need to be careful about what certificates users are allowed to scan. Finally, Dan suggested that banks need to rethink how they provide user authentication. Most banks do not have users enter login information on a secure SSL page. This can allow a hacker to hijack your session. Dan suggested that banks use iframes to cache the SSL secured page, and then switch to the protected page via JavaScript when a user goes to enter his or her password.

Unrelated to anything else, Dan also mentioned that SSH works very well as an extremely flexible VPN solution. However, he noticed that it had a tendency to leak DNS requests from remote users onto the local LAN. To resolve this problem he found a way to tunnel the DNS requests by basically going from DNS to SSH back to DNS.

All the slides from this presentation can be found at http://www.doxpara.com/ and at http://www.usenix.org/events/lisa06/tech/slides/kaminsky.pdf. Any released code can also be found at the Doxpara Web site. Dan has stated that he will respond to emails requesting the code talked about in this presentation.

Most questions revolved around whether the programs Dan discussed in his presentation were released or not. Others asked what he used his programs for. Dan stated that currently his hex program will not properly display files that have structures on bit boundaries and not byte boundaries. He also suggested that it may have potential uses for system administrator data sets, but he has yet to really test that.

INVITED TALK

Seriously, Tape-Only Backup Systems Are Dead

W. Curtis Preston, Glasshouse

Summarized by Nathaniel Husted (nhusted@iupui.edu)

W. Curtis Preston has been a data protection specialist for 13 years and is known to his friends as "Mr. Backup." According to Preston, if you're performing LAN-based backups directly to today's tape drives, you're doing nothing but shooting yourself in the foot. The problem with tape drives is that they are much faster then our networks and a streaming tape drive cannot write slower than its minimum speed. Matching the speed of networks with the speed of tape drives is not possible these days, because there has been a 27% increase in the speed of tape drives every year. Before 2001, tape drives and networks had the same speed, from 2001 to 2003 networks were faster than tape drives, but after 2003 tape drives outpaced networks, causing problems.

Variable-speed tape drives are available in the marketplace but all still have a minimum speed. Despite vendor claims, these drives have not eliminated shoe-shining. The good news is that a number of vendors have worked very hard on disk-based solutions that solve all these problems. Disk speeds are infinitely variable. Disks can take hundreds of slow or fast backups all at once without multiplexing. You can then dump them to tape at its maximum speed or replicate them to another location.

Disk can be used in various ways for backup both as a disk (e.g., SAN or NAS) and as a tape (e.g., a virtual tape drive or a virtual tape library). Now the main question is whether to opt for a filesystem or a virtual tape. The answer to this question depends on multiple factors, such as which backup software you are using (since not all are fully filesystem aware), the speed of backup, and whether the disk itself supports fragmentation (disk as tape doesn't support it). There is a provisioning/sharing issue when disk is used as a disk. VTL can be used as a standalone as well as integrated. A standalone VTL system sits next to your physical tape library, whereas an integrated VTL system sits in front of your physical tape library. A standalone system pretends to be another tape library, whereas an integrated tape library pretends to be your physical tape library.

VTL can be used as a single node or in clustered mode. "Clustered" VTLs allow you to expand capacity or throughput by adding additional data movers, but they manage as a single VTL.

All major VTL vendors are releasing de-dupe products right now. File de-duplication (sometimes called data reduction factoring) is space-saving technology intended to eliminate redundant (duplicate) files on a storage system. By saving only one instance of a file, disk space can be significantly reduced. De-duplication reduces effective cost of disk by 10:1 or more. To identify the redundant data, a hash comparison is used; calculating hash can be done with different hash calculation algorithms, such as SHA-1, MD5, or custom algorithms. Sometimes bit-level comparison is also used to double-check. Most products available in the market can use two methods. After the redundancy is identified, forward or reverse referencing needs to be used to identify the new data.

De-duplication can be in-band or out-of-band. In-band de-dupes in RAM and never writes redundant data to disk. Out-of-band de-dupes write original data to disk, read it later, and de-dupe it; hence it requires more I/O then in-band de-dupes.

There are various open source backup tools available in the market (e.g., BackupPC, Rdiff-backup, Rsnapshot). Preston also strongly recommends reading his book *Backup & Recovery*, which has 750 pages dedicated to free and open source backup.

POTPOURRI

Summarized by Gautam Singaraju (singara@uncc.edu)

The NMI Build & Test Laboratory: Continuous Integration Framework for Distributed Computing Software

Andrew Pavlo, Peter Couvares, Rebekah Gietzel, Anatoly Karp, Ian D. Alderman, and Miron Livny, University of Wisconsin, Madison; Charles Bacon, Argonne National Laboratory

Andrew Pavlo presented a framework for building and testing software in a heterogeneous, multi-

user, distributed computing environment. The automated tool, developed as a part of NSF Middleware Initiative, provides automated builds and tests access across administrative boundaries.

The users explicitly define the execution workflow of build-and-test procedures, which is stored at a central repository. The framework will dynamically deploy tests to appropriate computing resources. Any artifact that is created during the test process is transferred to the central repository. The NMI tool has been implemented on top of Condor, a high-throughput distributed batch-computing support tool. The NMI tool is both tool and platform independent, is lightweight,and provides well-controlled environments, centralized results, fault tolerance, and test separation.

Responding to questions, Pavlo pointed out that the software allows the users to control their test environment with a centralized repository that allows the users to replay their tests. Pavlo invited users to download the tool from http://nmi.cs.wisc.edu/.

- ***Unifying Unified Voice Messaging***

Jon Finke, Rensselaer Polytechnic Institute

Jon Finke presented his experiences merging a voice messaging system with the email domain. Rensselaer installed the unified voice messaging system after the voicemail system failed. By unifying the voice messaging into the Exchange servers, users were able to listen to their voicemail from their inbox. The files can be downloaded as .wav files, which can be used to forward the messages to other users. The Cisco Unity voicemail system was used to interact with Exchange servers. Standalone Exchange servers were used because the Active Directory schema change was not appreciated by the Exchange installation support personnel.

A tool was developed that managed voicemail by creating the mailboxes and populating the appropriate call handlers. With a call handler changing the extension was not necessary when a student changed rooms. Once the unified voicemail was configured on standalone Exchange servers, the system was then migrated to the production email domain with the help of a tool that copied each user's content into the production server.

- ***Fighting Institutional Memory Loss: The Trackle Integrated Issue and Solution Tracking System***

Daniel S. Crosta and Matthew J. Singleton, Swarthmore College Computer Society; Benjamin A. Kuperman, Swarthmore College

In the last presentation, a tool for part-time system administrators was presented. Trackle keeps a record of past actions, as the tool is used for documenting system services. Trackle provides an integrated trouble ticket and solution tracking system. Trackle has been developed as a tool that documents the process that is performed by an experienced system administrator. These actions are documented and can be used in educating untrained student system administrators.

Trackle has been developed to provide functionality for system administrators and users, an easy ticket-filing mechanism, and wiki-like referencing tools with minimal dependencies on existing software.

Referring to the additional feature requests, Crosta stated that the tool will provide ticket extensions, multiple machine support, file revision control, and further high-level abstractions. When asked about the tool, Crosta invited system administrators to download and try Trackle from http://www.sccs.swarthmore.edu/org/trackle/.

NETWORK SECURITY TRACK

- ***Zombies and Botnets: Attacks on Messaging Security by Organized Criminal Enterprises***

Dmitri Alperovitch, Research Scientist, Cipher Trust, Inc.
Summarized by Nathaniel Husted (nhusted@iupui.edu)

Dmitri Alperovitch is a former researcher for CipherTrust and now works for Secure Computing after the merger. He discussed recent trends in online attacks such as the increase in spam and phishing. He also discussed the prevalence of botnets and touched upon the organizations behind them. The talk ended with a short question-and-answer session.

Dmitri started by discussing various trends in online criminal activity from the past 25 years. Dmitri stated that criminals are getting dumber, but their populations are increasing. He also mentioned that the attacks are getting smarter and so are the security tools. Everyone is now a victim of these attacks. He also broke into three stages the types of people who have been behind these attacks for the past 25 years. The first stage was the hackers of the 1980s. The second stage was the crackers of the 1990s. The third and final stage is the organized crime enterprises we are now seeing in the 2000s. Dmitri said that these organized crime enterprises are now winning. Fully 90% of all email is spam, 400,000 bots appear daily, 5,990 vulnerabilities were reported in 2005, 3,744 phishing sites are found monthly, 200,000 viruses have been reported this year, and all these numbers have gone up from their previously recorded metrics.

Dmitri did state that things are improving, but there are also challenges. Law enforcement efforts are improving, arrests and prosecutions are going up around the world, international cooperation and trust are improving, and laws are slowly catching up to technology. Progress is still slow, however, because investigations can take months and even years to complete. There is also considerable corruption in some countries where these Internet criminals operate, thus allowing them to buy their way out of the judicial system. The enemy is also progressing in its tactics, becoming more secretive, less centralized, and operating with more secure communication channels.

Dmitri then discussed how spam and phishing attacks have evolved over the years. In the 1990s spam consisted of basic text-based messages. Then in 2003 they included random words to throw off the content-filtering spam blockers. Now, in 2006, the appearance of image-based spam has required a new style of OCR-based content filtering. The spammers are also committing large-scale but short-lived zombie attacks that make blacklisting useless. Dmitri also stated that in the future more spam will be image-based and possibly composed of random images from a zombie computer's hard drive. Phishing evolved from small-scale email-based phishes to special trojan-creating toolkits with a software support structure. Trojans were preferred over traditional phishing emails and Web sites because they are longer lived and easier to use. Dmitri ended the spam and phishing discussion by talking about various forms of online bank security and how much trojan-creating software costs.

The presentation finished with Dmitri discussing zombies and botnets. Zombies are the workhorse behind almost all online attacks. Now zombies adapt themselves to check for blacklists. Dmitri also stated that 20% of all bots are currently located in China, with the United States coming in second at 10.55%. Bots now have greater intelligence and also use peer-to-peer communication mechanisms instead of IRC. This modification in how bots talk to one another has made botnets harder to shut down. Dmitri stated that the best defense against botnets is to shut them down if possible or filter traffic from any known compromised network into a network with limited functionality. He also suggested that end users need to be made more responsible for the security of their machines. Other ways to lower the botnet population include increasing banking security and lowering the monetary benefits of spamming. Sadly, Dmitri stated that this problem will never go away, but we need to hope that it doesn't get any worse.

The questions asked in this talk revolved around what can be done to limit the propagation of zombies and why the problem is so bad. Dmitri suggested that one reason the bots are still spreading rapidly is that users still open executable email attachments. He also suggested that when IPv6 is implemented it might slow down the propagation because of the increased IP range that needs to be scanned. To make sure we are not the problem, Dmitri suggested that we make sure we know exactly where our network traffic is going.

INVITED TALK

Power-Managed Storage: Longer Data Life and Lower Energy Consumption

Aloke Guha, CTO, COPAN Systems

Summarized by Sumeet Jain (jain.sumeet@yahoo.com)

Aloke Guha began by saying that we have witnessed some of the most extraordinary growth in the recent history of technology in computing power, switching/routing capability, and data-carrying capacity. Data storage growth has outpaced all other growth rates, being explosive rather than exponential, touching 200 billion gigabytes in the year 2007, up from 12 billion in 2002. This focus has changed from system-centric in the 1970s to content-centric, with a complementary increase from ten million users to close to a billion users currently. We have witnessed islands of data move from Monolithic Direct Attached Subsystems to Dynamic Tiered Storage (DTS). DTS is capable of handling transactional as well as persistent data.

Managing persistent data is easier said than done when compared to managing transactional data. Persistent data, though matured (i.e., having a very low probability of any changes), has to be retained for a longer duration not only because of stringent regulatory compliance requirements but also because of the vital role it plays in business today. Coupled with the event-driven requirement, bandwidth constraints, and small recovery windows of few hours, the challenge of managing large volumes of data on tape drives is mission impossible. Tape backups are more beneficial for vaulting.

Storage on disks has its own challenges: power, cooling, reliability, longevity, maintaining life cycle from migration to salvage/regeneration, and, last but not least, floor space requirements. According to recent studies, power costs consume 40% of IT budgets and 33% of data centers expect to be out

of power and cooling capacity by the end of 2007, while 96% expect to reach the ceiling by 2011. More persistent data, almost 80% of which is being retained for a longer duration, still consumes power, has cooling requirements, and needs to be accessed online for regulatory compliance, while more data is being generated each minute for corporate governance and business continuity.

This is where MAID can help in managing data in a better and intelligent manner. MAID (Massive Array of Inexpensive Disks) is basically a large array of power-managed disks, with 50% of its drives powered off and only 25% of its drives spinning at any given time. It's a storage system comprising an array of disk drives that are powered down individually or in groups when not required, which helps to reduce power consumption. MAID has a three-tier architecture to scale performance with capacity and uses DISK AEROBICS software to check disk reliability and data integrity.

COPAN provides Enhanced MAID systems, which scale from 28 TB to 448 TB, with 5.2 TB per hour performance and the capability of handling 1 billion stored files (file and disk block) or 8,192 virtual tape cartridges. Since the system has been designed inside-outside with energy consumption and data integrity concerns, MAID performance outpaces FC storage systems by 2,140% and SATA-based storage systems by 424% on a TB/kW unit.

The MTBF of DISK AEROBICS software is 3.3 million hours, compared to 1.2 million for FC and 0.6 million for SATA. DISK AEROBICS proactively monitors and manages drives as well as RAID groups. Any suspect drive is backed up on spare drives and is "failed out" to avoid long RAID rebuilds. It performs policy-based turn-off of drives or RAID groups when not in use. It also assures drive health by exercising and testing idle drives at least once every 30 days to ensure data integrity.

INVITED TALK

The Future of System Administration: How to Stop Worrying and Learn to Love Self-Managing Systems

Alva L. Couch, Associate Professor of Computer Science, Tufts University

Summarized by Marc Chiarini (marc.chiarini@tufts.edu)

Alva Couch gave an informative talk about how the profession of system administration must change in the face of vendor-propagated autonomic computing initiatives. Vendors would have us believe that self-managing systems will be able to handle most current administration tasks on their own. On the assumption that they are correct, Couch proposes a clear path for evolving the job title of today's system administrators from "Plumber" to "Translator/Manager." To help us along, the talk focuses on the big picture before and after the coming "autonomic revolution." For example, the duties of managing configurations, twiddling bits on disk, troubleshooting configurations, and understanding file formats will be replaced by the duties of managing architecture, setting policy, analyzing system dynamics, and understanding performance factors, respectively. The dream of autonomics is to have present-day managers input business process and have everything work properly. The difficulty is that "manager speak" does not represent to an autonomic system what is *needed*. This is where the new sysadmin comes in. He or she will figure out what the real business process is, what aspects of it can be supported, and how to translate it for implementation by an autonomic infrastructure.

Pushing the profession into the next phase requires not just a new set of skills, but a new attitude as well. According to Couch, we have not yet started to retool our thinking in a way that will produce the new breed of sysadmin. This is dangerous because the modern world no longer promotes "survival of the fittest"; rather, we are in a world of "survival of those who fit." The old niche is full, and without a change in attitude there will be a slow death for the profession as we know it. We must create a new niche populated by "managers of human-computer communities." Old survival skills (communication, intrapersonal, time management, analysis, etc.) must be coupled with new survival attitudes: Value yourself and your professionalism, place management goals above self-interest (which requires understanding the attitudes and language of management), be able to "close the box" and delegate, and be able to leave good enough alone. The new sysadmin cannot base his or her job security upon being essential now; sysadmins must be perceived as essential to the future. The best way to become more important and indispensable is to get on management's radar by making your job the easiest (and most efficient) way to accomplish business objectives.

Couch went to review several case studies and lessons from his own experience. Good works are not always sufficient to keep your job in the face of changing business needs or structure: In a single year, everyone important at Tufts University

who owed Couch a favor for pulling them out of the fire was let go, forcing him to rebuild his reputation. As another example, he took a look at database administration. He made the convincing points that design is already outsourced, automation can tune performance to within 80% of human capability, and much programming is being replaced with reflection modeling. To maintain his value, the new DBA really needs to serve as the interface (and interpreter) between management and infrastructure. As a third angle, Couch urged us to consider autonomics as analogous to asbestos abatement: This stuff is dangerous; one slip and the business loses a lot of money; it's all driven by complex policies that untrained people (i.e., those in management) shouldn't try to understand. Finally, Couch provided tips for interfacing with management: Stop distinguishing between "us" and "them"; make our goals the same as those of management; learn to speak like the managers; learn to justify our decisions in management terms; listen intently instead of explaining; and make ourselves partners rather than servants.

During the Q&A session it was pointed out that this type of evolution is nothing new. Couch agreed but went a step further by claiming that if autonomics begins to replace the junior sysadmin, the training loop will be broken and it may spell disaster for the profession. It was asked whether we have enough time to evolve, given the reduction in LISA attendance in the past few years. The response was that the drop was due to social rather than technical factors and that the profession may actually be reaching a saturation point (all the more reason to evolve more quickly). When asked to what extent the new breed of sysadmin will need (or want) to understand the technology of autonomics itself, Couch answered by analogy to admins understanding the kernel.

WORK-IN-PROGRESS REPORTS

Summarized by Robert Marmorstein and Beth-Lynn Eicher (rmmarm@cs.wm.edu, bethlynn@lookandsee.net)

- ***NAGIOS and SEC: A Happy Re-Union for Advanced System Monitoring***

  John Rouillard, University of Massachusetts, Boston

  SEC is an event correlator that supports many UNIX platforms. SEC can act as an event mapping layer between NAGIOS plug-ins and the NAGIOS core to set thresholds for alarms. SEC can also monitor NAGIOS log files for errors. John is currently looking for beta testers interested in this combination of tools. Slides are available at http://www.usenix.org/events/lisa06/wips.html.

- ***PoDIM: Policy Driven Infrastructure Management***

  Thomas Delaet, Katholieke Universiteit, Leuven

  PoDIM is a generalized and scalable mechanism for component distribution. It interfaces with several backends, including bcfg2 and cfengine2. The software is not currently available. Thomas is working on the PoDIM project as part of his Ph.D. study at K.U. Leuven, Belgium.

- ***NIS to Kerberos in a Psynch Environment***

  David Pullman, National Institute of Standards & Technology

  David talked about his experiences migrating an NIS-based account system to a Kerberos-with-LDAP system. He explained the challenges involved in providing account locking and unlocking, as well as giving an outline of the architecture he used to implement the transition. The result was a Psync portal that entailed NIS, LDAP, and Kerberos.

- ***Using Redirection to Enhance Software Management***

  Marc Chiarini, Tufts University

  Marc discussed the problem of nondistribution standard packages overwriting files belonging to other packages. He described a solution in which packages are repackaged and wrapped in a special environment so that libraries and other dependencies match. This also protects the core distribution and allows multiple software environments to coexist peacefully.

- ***Symlinking for Fun and Profit***

  Wout Mertens

  Wout talked about downward compatibility problems he had with multiple versions of Solaris sharing an NFS file system. By using symbolic links and custom-built scripts he was able to design a "poor man's union-mount" that alleviated these problems. He also discussed some of the stumbling blocks to this approach (e.g., the sudoers file cannot be a symlink) and how he was able to address those issues.

- ***Portable Cluster Computers and Infiniband Clusters***

  Mitch Williams, Sandia National Laboratories

  Mitch gave a slideshow presentation featuring various clusters he has built. These systems vary in size from a foot high to several racks. Two notable examples are the toolbox-sized cluster of 16 ARM 200s, which was presented on the show floor of Supercomputing 2006, and the 64-TB Flustre storage cluster, which is in production at Sandia National Laboratories.

■ ***Miscellaneous Data Management II***

Jason Heiss, Yahoo!

Configuration management has been a very hot topic in the system administration world. Jason pointed out that data management is also a very challenging problem that deserves attention. Jason talked about various solutions for managing drives, data, and backups. He focused on managing small chunks of data from service configurations, such as a Kerberos database.

■ ***A Configuration Management Tool Used by the University of Wisconsin, Madison, Computer Science Department***

David Parter, University of Wisconsin, Madison, Computer Science Department

Using an existing inventory database, David created a configuration management tool that simplifies installation, configuration, and maintenance of his department's systems. The tool uses simple XML templates to describe the system policy and interfaces with both kickstart and jumpstart. The system also created templates for common types of systems (desktops and research servers) at his university.

■ ***What Is a Computer?***

Beth Lynn Eicher, Look and See Networks

Is your cell phone a computer? What about your dishwasher or your toaster? The evolution of technology has blurred the lines between what is a computer and what is not. System administrators were challenged to think of computers and users in a larger than traditional scope. Beth Lynn broke the issue down into several yes-or-no questions and emphasized the importance of privacy in computing.

After each talk was presented the WiPs chair, Esther "Moose" Filderman, called for an audience vote by round of applause. Traditionally the crowd has one or two favorites, but this year everyone enjoyed all the presentations equally. The dilemma of who gets the coveted WiPs award was quickly resolved when someone had suggested that Anthony from MSI ("the A/V guy") deserved the prize for quickly resolving projector issues.

NETWORK SECURITY TRACK

■ ***Corporate Security: A Hacker Perspective***

Mark "Simple Nomad" Loveless, Security Architect, Vernier Networks, Inc.

Summarized by Kevin L. James (kevljame@cs.iupui.edu)

Mark Loveless enlightened us about the nature of hacking today. The founder of the NMRC hacker collective, now a security researcher, he still maintains ties with hackers on both sides of the fence: black hats, who crack systems for pride, politics, and profit, and white hats, who attempt to find flaws in systems before they can be exploited.

He began with a list of hacker "goals and dreams." The first goal is to find 0days (number zero) and to be the first to exploit a flaw. Next is remote root access, a dream of many hackers as it allows unfettered access to a system without being logged into the console. The holy grail of hackerdom is the remote root 0day, wherein not only is complete control gained, but in a way that has never before been seen, and therefore is more difficult to detect and stop. According to Loveless, 0days are worth more than ever.

Mark next expanded on the concept of 0day. Originally, 0day meant the number of days a commercial piece of software had been on the market before it was hacked. Cracking a copy of a new game before it is even released was considered "wicked cool." Attempts at this were common because of the copy protection measures used in the day: All software had to be cracked to back it up. When it came to exploiting security flaws, an exploit ceased to be 0day when vendors or system administrators discovered it. Today, 0day refers to an unpatched flaw that vendors and sys admins have discovered. This applies to both nonpublic working exploits for a patched flaw and those reported to vendors or industry groups by researchers.

Interestingly enough, both white hats and black hats have a disclosure cycle when it comes to discovered flaws. Researchers report flaws to software vendors, who in turn develop a fix for the flaw and release a patch for it. Afterwards, the researcher releases an advisory to a third-party group such as Carnegie Mellon's CERT (Computer Emergency Response Team, www.cert.org). Excluded from these advisories are the technical details of the flaw. In contrast, when attackers find flaws, they share their finds with very few close friends in an effort to minimize usage of the exploit and therefore vendor discovery. Before the flaw is discovered, the attacker also attempts to find other flaws. Mark says that hackers often sell their used or discovered flaws. When these two disclosure cycles clash, both hats work vigorously to reverse-engineer patched and unpatched versions of fixes using tools such as bindiff and clues from advisories to narrow the focus of reverse engineering. Similarly, they also develop exploit code based on the discovered flaw: white hats to develop scanning signatures and black hats to de-

velop code that will be used to attack unsuspecting systems. Another commonality to their jobs is that they often look evidence of silent patches done by vendors to determine whether they fix a possible exploit. According to Loveless, all vendors patch silently at times.

He next described trends in targeted penetration and attacking techniques. Although attempts at breaking into systems still abound, statistically, the successes have not grown proportionately. Referring to the popular book series, he called much of the current crop the "Hacking Exposed" generation, as many sys administrators protect their systems using these types of book. Conversely, these books are also used by would-be hackers. To their detriment, he says, these books talk simply about the act of penetration of a system, focusing on perimeter security. They also fail to give fledgling hackers tips, such as not hacking from your own system.

There are also some new "wrinkles" in old reconnaissance techniques. Many hackers use known exploits to make sure their attacks are discovered. This is done to judge the responsiveness of system administrators, to see whether they are simply immediately fixing exploits or actually watching the attacks to see their patterns. Another interesting technique is to use "dark IP space." Many admins check to see whether their unused IP space is being probed, because if someone is attempting to use unallocated IP space, they are probably an attacker. Conversely, clever attackers sometime attack this space to see whether an admin is really watching. There are even ideas to determine whether an automated system is in use and the type. A final technique is to use a 0day attack masked by many well-known exploits. The hope is that administrators will be too occupied dealing with the known problems to determine how the attacker actually got in.

Next Mark gave facts about the professional black hat world. Traditionally, they work for a single very organized group that specializes in spamming, spyware, and id theft. Many of these groups are run by organized crime organizations and are very much like regular software businesses, with tight release cycles and product testers. They are very well paid (around $200,000 per year), often for substandard work. Some of these activities are even funded by nation states, organized cybercrime, and even legitimate computer defense companies that are willing to pay $40,000 to $120,000 for a remote root 0day. There are even sites acting like EBay, complete with rating systems, where hackers buy, sell, and trade exploits and stolen identities.

Freelance black hats also work for spammers and information brokers. They are more concerned with keeping 0days hidden from vendors and administrators. Often they are very proficient at reverse engineering, making money from exploits alone. Loveless quips, "Anytime you couple questionable morals with some type of mad-coding-fu going on, you're going to make the mad-coding-fu money."

Next he talked about what's hot in hacker circles. He said that anything WiFi or Bluetooth is very popular, as they offer disconnected points of attack. There has also been a trend toward targeted malware as more money has become involved in the process. The more victims reported, the more likely people are to patch the flaw. To minimize this, IP ranges or targets that have been prelaundered by previous spamming efforts are more likely to be attacked. Another new area will be Blackberry-like devices, as they are able to connect with many diverse systems. Lastly, vast botnets (large clusters of machines that are used for attacks) are occurring in six-figure sizes and being leased to other attackers or groups to launch attacks. Current hotspots include the perennial Microsoft, but also Apple, which he describes as one of the worst reporters of security flaws.

Concluding, Mark Loveless offered a few suggestions. Continue to patch regularly, as the average time from patch to exploit is shrinking. Limit access to what is needed, because "locking down ACLs can save your bacon with regard to 0day." Finally, consider new types of technologies to reinforce your security, such as newer, more intelligent intrusion detection and protection systems.

INVITED TALK

System Administration: Drowning in Management Complexity

Chad Verbowski, Software Architect, Microsoft Research

Summarized by Raj Kumar Gurung (RK-Gurung2@wiu.edu)

This invited talk dealt with the growing complexities in systems and provided various approaches for systems management to aid system administrators in increasing the number of systems a single administrator can effectively manage. Complexity is constantly growing with the growing number of devices, applications, and users. Key pints were that we simply cannot rely on software advances

to address this complexity and that advances in the management space post-software development time are required. Verbowski proposed a data-driven management approach to reduce the complexity by using automated monitoring and analysis tools. The main advantage of this approach is that it traces all the interactions between the applications and configurations for analysis, thus providing simple troubleshooting space, reducing the problem space for other techniques, and leveraging existing machine-learning work. However, designers should always keep in mind scalability and cross-machine equivalence.

ADVANCED TOPICS WORKSHOP

Summarized by Josh Simon (jss@clock.org)

Tuesday's sessions began with the Advanced Topics Workshop; once again, Adam Moskowitz was our host, moderator, and referee. We started with an overview of the revised moderation software and general housekeeping announcements. (Well, we really started by picking on Adam, on Andrew Hume, who earned 2005's Most Talkative Participant award, and on Trey Harris, who was 2004's Most Talkative by a factor of 2.)

We followed that with introductions around the room. For a variety of reasons, several of the Usual Suspects weren't at this year's workshop; in representation, businesses (including consultants) outnumbered universities by about 3 to 1; over the course of the day, the room included three LISA program chairs (one each past, present, and future; down from five last year) and five past or present members of the USENIX, SAGE, or LOPSA Boards (down from seven last year).

We went around the room to say how we believed system administration has changed in the past year. The general consensus seemed to be autonomic systems; challenges of new jobs; education of teachers to bring them up to what students know; improvements in automation; life-event changes, with marriages, deaths, and births; lower budgets; metrics; more fallout from legislation (such as SOX); more reliance on external infrastructure, such as external mail, calendar/scheduling systems, and wikis; organizational restructuring and staff turnover; targeted offshore security attacks; telecommunications integration; and rising virtualization.

Our first subject of discussion was storage. Several organizations have larger and larger storage needs; one example is a site that's growing at 10 TB a month or 2.5 PB a year, and smaller (such as 16-GB) drives no longer scale. Other places are more than doubling their data storage every year. We discussed some options, such as the promise of iSCSI/ZFS (with which current users are pleased, for the most part), and the forthcoming open source GFS-like. The comment about determining your real needs and taking metrics is important: Some 1/10-GB switches can't really switch among many users, and if you're not measuring end to end you won't know where your bottlenecks are.

In addition to primary storage needs (how much disk is needed? how is it attached? what bandwidth do you need?), there are ancillary issues, such as backups (do you store snapshots on disk? do you back up to tape or to another spinning disk? how much is really enough, given that nobody ever deletes data?). One point was to use software compression before the data hits the tape drive; for example, hardware compression can require 90 MB/s for 3:1 compression.

Another point was that if we do the math on ECC corrections, we find we're now having enough disks that at one site in particular we are seeing bit-rot rates in untouched files on spinning disks of about 1 error per several terabyte-years (1 TB spinning for 1 year, or 2 TB for 6 months). Yes, the rate is very very low, but it's definitely nonzero, so if you don't always checksum everything you have the risk of bit-rot and thus lost or corrupted data on disk (which leads to the issue of where you store your checksums and what happens if they themselves get bit-rot).

We digressed into a brief discussion of backups: Do you back up files just at the OS level or at the application level as well? Do you back up laptops? Rates of backup can differ among data types: The OS tends to change less frequently than home directories, for example. Finally, consider not backing up what you don't need (for legal, compliance, regulatory, and similar reasons). It's recommended that if you don't have a policy, you should write one yourself, then get approval or rewrites from your legal or compliance folks afterwards.

Our next large topic area for discussion was monitoring. We went around the room: 29% are using some kind of home-grown monitoring software, 83% are using open source tools, and only 8% are using commercial software. (You'll notice that these numbers won't add up to 100%, as several places use combinations.) The software packages explicitly mentioned include Big Brother, Cacti, cron that sends email on unexpected errors, home-grown syslog watcher, logcheck, MRTG, Nagios, NetCool, Net Vigil, OpenNMS, RRD,

smokeping, and Spyglass. Most people tolerate their monitoring, but very few are "very happy" with it. Nagios had the largest representation. The consensus seemed to be, "It's the best of the bad choices"; although most of us use it, nobody was an evangelist for it. In general, the suggestions are:

- Monitor what does happen that shouldn't.
- Monitor what didn't happen that should've.
- Monitor what you care about; don't monitor what you don't care about.
- Gather history: We may not care about one ping down, but we do care about multiple successive failures, and then we won't care until it comes back and stays up (the success table).

One problem is that we need more detail than just "up/down." Nagios as written doesn't differentiate among several states: Is it there (ping)? Does it have a heartbeat? Did it give a response? Did it give a valid response, and did it give a valid and timely response? The phrase "service is up" isn't necessarily meaningful. We discussed what we'd want in an ideal monitoring system, including cooperative signaling, so "If I take 10 minutes it's okay, if it's longer there's a problem" is a valid case.

Another issue we have with Nagios is that it often doesn't monitor the right things, or it performs the wrong tests. Who writes your tests? Is it the person responsible for the application, or a monitoring group, or someone else? The actions taken also need to be aware of criticality: How urgent is the problem? How often should it be tested for? and so on.

This led to a discussion about machine learning (monitoring tools that build or configure themselves) and self-aware applications that can determine on their own whether they have a problem and can send alerts themselves. Better application design can lead to better monitoring.

After our lunch break, we went through and mentioned tools new to us as individuals since last year's conference; the tools included Adobe Lightroom, Asterisk, Aware I Am, decoy MX server to block spammers, DocBook SGML, Dragon Naturally Speaking, drupal, Google Spreadsheets, hardware security monitors and crypto ("key roach motels"), IP KVM, IP power, IPMI cards, isolation booths at work for private phone calls, LYX, Mind Manager for mind mapping, Mori, OpenID, Password Safe, photography management (for births and weddings), Rails for admin interfaces, relationships with intellectual property lawyers, RSS feed-reading software, SQL Gray, Solaris 10, Solaris Zones and ZFS, Sparrow, USB-attached RFID readers, VOIP, wikis (because "they work now"), and x2vnc and x2x.

Next we talked in more detail about ZFS. Someone asked if it was as wonderful as the hype said it would be, and the answer boiled down to "Yes and no." For the most part, it's very very well designed. It does what you want, and even though it sounds too good to be true it's pretty close to that. However, if you use it long enough you'll see the warts. It works well with zones, but not everyone at Sun support knows enough to troubleshoot problems; so far, there's only one commercial product to back it up (Legato); there aren't any best practices; and there's no way to say, "Evacuate this disk and give it back to me."

Next we discussed calendaring. As a group we use a lot of software and at best we tolerate it. The big ones are Exchange's calendaring on the PC side and iCal on the Mac. We came up with a feature list of a good system, which included multi-OS, specifically Mac, Windows, Linux, Solaris, HP-UX, and *BSD; integrating both home and work calendars, keeping them separate so other "home" users (such as spouse and kids) can only see the "work" entries as "busy" without details; being able to see free/busy on others' calendars and to schedule events with negotiation, which requires ACLs of some kind. There's no good solution yet.

We next discussed cheap scientific clusters. Now that there are quad-CPU dual-core processors, someone built an inexpensive yet effective four-node (soon growing to ten-node) cluster with Infiniband Infinipath for internode communication and gigabit TCP/IP for networking. The cluster uses RAID 5 on the head node, and each node has 32 GB RAM. This cluster can almost make the decade-old Cray obsolete (since just about any job can use up to 32 GB of memory and the Cray has only 40 GB). It's doing better than expected, but it's very noisy.

This led us to a discussion about power consumption and heat generation. One site recently got a supercomputer grant for hardware that needs 300 tons of coolant, but its entire data center only has 45 tons; the entire campus doesn't use as much power as this one supercomputer will (once it's fully loaded). Going to virtual machines reduces power and heat by using several smaller virtual machines on one larger machine. Some articles say that DC power helps some, since you can avoid converting between DC and AC. There's not a huge market for better power consumption yet,

mainly because few people in the purchasing processes are discussing it, but if you require low-power, low-voltage, slower-but-cooler hardware in the hardware selection process, possibly by specifying "total wattage" instead of a number of systems, the market will auto-correct and give you more options. Other suggestions for reducing power consumption and heat generation included replacing your CRTs with LCD flat panels, using thin clients in conference rooms and secretarial desks where you don't need an entire PC (which has positive side effects on security), replacing desktops with permanently docked laptops, and replacing incandescent lights with compact fluorescent lights. Any and all of these can reduce your power costs, cooling costs, and fan noise.

After the afternoon break, we talked about support changes. As has been the case in recent years, more places are trying to do more—more services, more products, more projects, more hours of support—with fewer resources—fewer or the same number of people, fewer machines, and so on. In general, folks are accomplishing this by remote access (ssh into corporate environments, remote VNC to client, or customer machines supported from the technician's desk). There is also the issue of who supports home machines: Because they're used by the home and the corporation, they don't fit neatly into most support categories. It should be noted that supportability implies backups.

We next went around the room to discuss our most important or most difficult problems. This year, the big one was resource allocation: insufficient staff in both quantity and training, and insufficient time. Finding people is hard, keeping people can be hard (they quit or are reorganized away from your team), and cross-team communications is often hard. There are often too many fires to put out, so prioritizing which fire gets fought first is necessary. The other most common problem is the learning curve; several of us are in new environments and it's challenging first to learn what was done and why, and how things got into their current state, and then to improve things to use best practices; many resist change management, even at the level of "Tell someone when you change something." The third most common problem is career management: What can we do when we're getting bored with our current role, or if there's no growth path to "senior engineer"? Finally, compliance (for legal issues, such as HIPAA and SOX) is taking up more of our time; about 25% of us are doing more with it now than last year.

Finally, we discussed what's on our horizon, or what we expect the next year will be like for us. We predict that our challenges for the next 11 months will include application and OS upgrades back to the bleeding edge; clustering; compliance; exponential scaling; leading understaffed teams and dealing with staff retirement; making the infrastructure more reliable, more robust, and more reproducible; virtualization; and working with 10GigE.

CONFIGURATION MANAGEMENT TOOLS AND PRACTICE WORKSHOP

Summarized by Chris Cooke and Sanjai Narain (cc@inf.ed.ac.uk, narain@research.telcordia.com)

This year's workshop focused on configuration validation. Sanjai Narain presented the motivation. A central infrastructure management problem is testing whether infrastructure complies with end-to-end requirements. Requirements can be on functionality, security, performance or reliability, or those derived from government regulatory policies. Often, these span multiple components and layers of abstraction. Typical approaches to compliance testing, such as invasive testing and simulation, have significant limitations. A new approach that overcomes these limitations is noninvasive analysis of component configurations. These configurations represent the "source code" of infrastructure, in that deep prediction of infrastructure behavior can be made from their analysis. A new class of algorithms, analogous to that for static analysis of software, needs to be developed. This workshop brought together many researchers investigating this idea.

Dinesh Verma presented his work expressing configuration constraints as policies, and then ensuring conformance with those policies. This work has been applied to configuration validation of storage area networks.

Rajesh Talpade discussed a software system called VCAS for vulnerability and compliance assessment for IP networks. Over the past two years VCAS has been successfully undergoing trials at the infrastructure of six major enterprises. It is based on patent-pending algorithms for diagnosing vulnerabilities such as single points of failure and those arising out of interactions between protocols. It contains a proprietary, vendor-neutral knowledge-base of rules covering most IP network protocols.

Srinivas Bangarbale discussed challenges of managing change in mid-sized enterprises. Configuration management is a much needed discipline but many factors stand in the way of a successful configuration management practice: organizational culture, the need for flexibility, and operating constraints. Whereas large organizations can afford the overheads of a full-fledged configuration management practice, and small ones may not need to be as rigorous as the large ones, mid-sized enterprises are frequently caught between the two extremes and find the good solution to be a tough balancing act.

Geoffrey Xie argued that in order to turn network configuration into a principled engineering process, it is critical to develop a class of high-level abstractions, each equipped with a unifying analytical framework for reasoning about the joint effect of related low-level mechanisms and protocols. He introduced such a high-level abstraction, called the reachability matrix, for modeling network security policy.

John Orthoefer discussed configuration management from the systems administrator perspective, including what works in real-life situations, from small sites with fewer than 10 people and a few hundred machines, to large sites with more than 20 people and thousands of machines over a geographically dispersed area. The concept of what is a "valid configuration" and how one arrives at that configuration differs for these two cases.

Sanjay Rao discussed two key challenges to establishing configuration management as a rigorous scientific discipline in academia. First, access to configuration data is limited. Second, evaluating solutions and demonstrating "success" is also difficult. To address these challenges, his team is setting up a "white-box" approach involving extensive collaboration with network operators, including Purdue campus operators and AT&T. His goal is to empirically study operational networks with a view to systematically understanding operational practice and scientifically quantifying trade-offs managers are making while designing networks.

Yiyi Huang presented a technique for improving fault-isolation by analyzing network-wide routing configuration information. For the Abilene network, this technique detected every reported disruption to internal nodes and links and correctly explained the scenarios that caused the disruptions.

Paul Anderson presented an overview of the LCFG configuration management tool, along with thoughts on how it could support configuration validation and synthesis.

Panel Discussion on Configuration Management: The Big Picture

Alva Couch started the discussion by saying that we need a formal way to express constraints and specifications: He pointed out that during the workshop all the participants had been doing this in English. But what sort of thing would be acceptable to, and used by, system administrators? Sanjai Narain suggested that first-order logic would be suitable. He enquired whether it would be useful to embed this into languages such as Perl that system administrators already know and understand. Mark Burgess said that we need to model behavior, not specifications. The language must encompass uncertainties: unreliability, voluntary co-operation, access restrictions. He disagreed about the suitability of first-order logic because it has no typing. A meta-model is needed to relate things to each other.

Panel Discussion on Are Tools Enough?

Alva Couch also led this discussion at the end of the day. For the discussion he invited Æleen Frisch, Tom Limoncelli, and David Parter. The invited guests made a collective plea for simplicity and ease of use in configuration management tools, reiterating and emphasizing a point that had been made by John Orthoefer earlier in the day.

Æleen said that part-time, busy system administrators will not use configuration management tools if they're too complicated for the matter at hand. Needed are proper, real, more comprehensive examples of how to do realistic, valuable, real-world tasks with configuration management tools.

Tom Limoncelli made a similar point. Configuration management tools have a huge barrier to entry. He suggested that the main tool developers spend the next year removing that big barrier. He suggested stopping all-or-nothing solutions and starting with just one little hack. He made a memorable plea: "Stop making tools for smart people!" It really would be better for vendors to adopt mediocre configuration management standards than have wonderful configuration management that nobody uses. Get the vendors on board. Get standards that marketing people can boast about. He also suggested an approach to achieving this. To help a configuration management tool grow in popularity, get the authors of the ten most popular

open source software packages to support and provide hooks for your configuration management tool.

Mark Burgess objected to the implication that there are currently no configuration management standards, pointing out that international telecommunications companies do already have standards in this area, to which they are required to work.

Kent Skaar elaborated on the examples: We need not just examples, but explanations of the thought processes behind them; we need design patterns.

Luke Kanies pointed out a bootstrapping problem: A selection of real-world, useful, usable examples to accompany a configuration management tool can only come from the tool's user community. How can a community be built around the tool in order to get the examples, without the examples already existing? He would like Puppet to have such examples, but "there is no community."

Tom Limoncelli also emphasized the quick-changing nature of business process: A company's business process tends to be driven by what magazine the CEO just read! Such things are ephemeral; a new one will be along next week. Nevertheless, system administrators seem to have to spend a great deal of effort dealing with arbitrary, ad-hoc, unsuitable, or unworkable technical diktat coming from nontechnical management. Also, a lot of system administration involves systems that have been running for years and were set up by staff long since departed; such work can be termed "system archaeology." Configuration management tools have to be able to deal with such suboptimal real-world circumstances.

Main Ideas from the Workshop

Two ideas seem to have predominated. First, there was Alva Couch's observation that we should move from managing components to managing architecture. This was reiterated in his talk and paper presented at the conference. Configurations can be more helpfully represented as an interlocking mesh of interrelated "aspects" than as a lot of individual configuration parameters.

The second idea to be presented again and again was a plea for simplicity and ease of use of configuration management tools. System administrators often find them too complex and frightening to adopt. Easy routes to adoption have to be provided before a large-scale take-up of configuration management tools can take place.

About three dozen people attended the workshop. Of these, 12 were from academia, 1 was a consultant, 15 were configuration management tool developers, 15 did configuration management–related research, and 15 were new to the workshop.

For additional information, see http://homepages.inf.ed.ac.uk/group/lssconf/config2006/index.html.