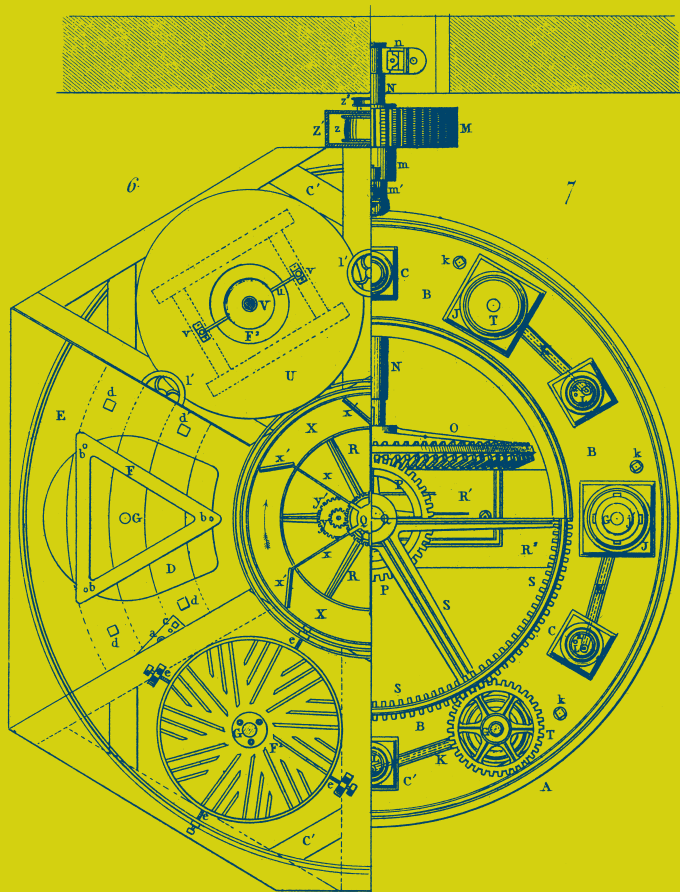




THE USENIX MAGAZINE



OPINION

Musings 2
RIK FARROW

SYSADMIN

IPv6: It's Time to Make the Move 7
MARK KOSTERS AND MEGAN KRUSE

Are Commercial Firewalls Ready for IP Version 6? 14
DAVID PISCITELLO

An Introduction to Logical Domains, Part 4 24
OCTAVE ORGERON

Linux Kernel Resource Allocation in Virtualized Environments 34
MATTHEW SACKS

Hacking 802.11 Protocol Insecurities 38
ADITYA K SOOD

PROGRAMMING

Achieving High Performance by Targeting Multiple Parallelism Mechanisms 45
MICHAEL D. MCCOOL

COLUMNS

Practical Perl Tools: Back in Timeline 55
DAVID N. BLANK-EDELMAN

Pete's All Things Sun (PATS): The Security Sheriff 62
PETER BAER GALVIN

iVoyeur: Comply 68
DAVID JOSEPHSEN

VoIP and IPv6 73
HEISON CHAK

/dev/random 76
ROBERT G. FERRELL

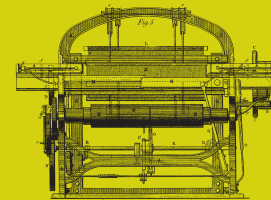
BOOK REVIEWS

Book Reviews 78
ELIZABETH ZWICKY ET AL.

USENIX NOTES

Open Public Access to All USENIX Conference Proceedings 82

USENIX Upcoming Events



USABILITY, PSYCHOLOGY, AND SECURITY 2008

Co-located with NSDI '08

APRIL 14, 2008, SAN FRANCISCO, CA, USA
<http://www.usenix.org/upsec08>

FIRST USENIX WORKSHOP ON LARGE-SCALE EXPLOITS AND EMERGENT THREATS (LEET '08)

Botnets, Spyware, Worms, and More

Co-located with NSDI '08

APRIL 15, 2008, SAN FRANCISCO, CA, USA
<http://www.usenix.org/leet08>

5TH USENIX SYMPOSIUM ON NETWORKED SYSTEMS DESIGN AND IMPLEMENTATION (NSDI '08)

Sponsored by USENIX in cooperation with ACM SIGCOMM and ACM SIGOPS

APRIL 16–18, 2008, SAN FRANCISCO, CA, USA
<http://www.usenix.org/nsdi08>

THE SIXTH INTERNATIONAL CONFERENCE ON MOBILE SYSTEMS, APPLICATIONS, AND SERVICES (MOBISYS 2008)

Jointly sponsored by ACM SIGMOBILE and USENIX

JUNE 17–20, 2008, BRECKENRIDGE, CO, USA
<http://www.sigmobile.org/mobisys/2008/>

2008 USENIX ANNUAL TECHNICAL CONFERENCE

JUNE 22–27, 2008, BOSTON, MA, USA
<http://www.usenix.org/usenix08>

2ND INTERNATIONAL CONFERENCE ON DISTRIBUTED EVENT-BASED SYSTEMS (DEBS 2008)

Organized in cooperation with USENIX, the IEEE and IEEE Computer Society, ACM SIGSOFT, and ACM SIGMOD

JULY 2–4, 2008, ROME, ITALY
<http://debs08.dis.uniroma1.it/>

2008 USENIX/ACCURATE ELECTRONIC VOTING TECHNOLOGY WORKSHOP (EVT '08)

Co-located with USENIX Security '08

JULY 28–29, 2008, SAN JOSE, CA, USA
<http://www.usenix.org/evt08>

2ND USENIX WORKSHOP ON OFFENSIVE TECHNOLOGIES (WOOT '08)

Co-located with USENIX Security '08

JULY 28, 2008, SAN JOSE, CA, USA
<http://www.usenix.org/woot08>
Submissions due: June 1, 2008

WORKSHOP ON CYBER SECURITY EXPERIMENTATION AND TEST (CSET '08)

Co-located with USENIX Security '08

JULY 28, 2008, SAN JOSE, CA, USA
<http://www.usenix.org/cset08>
Paper submissions due: May 15, 2008

17TH USENIX SECURITY SYMPOSIUM

JULY 28–AUGUST 1, 2008, SAN JOSE, CA, USA
<http://www.usenix.org/sec08>

3RD USENIX WORKSHOP ON HOT TOPICS IN SECURITY (HOTSEC '08)

Co-located with USENIX Security '08

JULY 29, 2008, SAN JOSE, CA, USA
<http://www.usenix.org/hotsec08>
Position paper submissions due: May 28, 2008

22ND LARGE INSTALLATION SYSTEM ADMINISTRATION CONFERENCE (LISA '08)

Sponsored by USENIX and SAGE

NOVEMBER 9–14, 2008, SAN DIEGO, CA, USA
<http://www.usenix.org/lisa08>
Extended abstract and paper submissions due: May 8, 2008

SYMPOSIUM ON COMPUTER HUMAN INTERACTION FOR MANAGEMENT OF INFORMATION TECHNOLOGY (CHIMIT '08)

Sponsored by ACM in association with USENIX

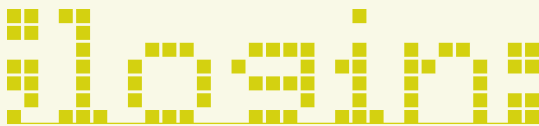
NOVEMBER 14–15, 2008, SAN DIEGO, CA, USA
<http://www.chimit08.org>

ACM/IFIP/USENIX 9TH INTERNATIONAL MIDDLEWARE CONFERENCE (MIDDLEWARE 2008)

DECEMBER 1–5, 2008, LEUVEN, BELGIUM
<http://middleware2008.cs.kuleuven.be>
Abstracts due: April 23, 2008

For a complete list of all USENIX & USENIX co-sponsored events, see <http://www.usenix.org/events>.

contents



VOL. 33, #2, APRIL 2008

EDITOR
Rik Farrow
rik@usenix.org

MANAGING EDITOR
Jane-Ellen Long
jel@usenix.org

COPY EDITOR
David Couzens
proofshop@usenix.org

PRODUCTION
Casey Henderson
Jane-Ellen Long
Michele Nelson

TYPESETTER
Star Type
startype@comcast.net

USENIX ASSOCIATION
2560 Ninth Street,
Suite 215, Berkeley,
California 94710
Phone: (510) 528-8649
FAX: (510) 548-5738

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

login: is the official magazine of the USENIX Association.

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

\$90 of each member's annual dues is for an annual subscription to *login*. Subscriptions for nonmembers are \$120 per year.

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

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

©2008 USENIX Association

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

OPINION	Musings RIK FARROW	2
SYSADMIN	IPv6: It's Time to Make the Move MARK KOSTERS AND MEGAN KRUSE	7
	Are Commercial Firewalls Ready for IP Version 6? DAVID PISCITELLO	14
	An Introduction to Logical Domains, Part 4 OCTAVE ORGERON	24
	Linux Kernel Resource Allocation in Virtualized Environments MATTHEW SACKS	34
	Hacking 802.11 Protocol Insecurities ADITYA K SOOD	38
PROGRAMMING	Achieving High Performance by Targeting Multiple Parallelism Mechanisms MICHAEL D. MCCOOL	45
COLUMNS	Practical Perl Tools: Back in Timeline DAVID N. BLANK-EDELMAN	55
	Pete's All Things Sun (PATS): The Security Sheriff PETER BAER GALVIN	62
	iVoyeur: Comply DAVID JOSEPHSEN	68
	VoIP and IPv6 HEISON CHAK	73
	/dev/random ROBERT G. FERRELL	76
BOOK REVIEWS	Book Reviews ELIZABETH ZWICKY ET AL.	78
USENIX NOTES	Open Public Access to All USENIX Conference Proceedings	82

RIK FARROW

musings



rik@usenix.org

I HAVE BEEN TO THE BRAIN GYM RE-
cently and really gotten a workout. Perhaps
you've heard all the buzz about how you
need to "exercise" your brain if you want to
stay sharp as you get older. Although many
products out there purport to do this, all I
need to do is leap into learning something
new. Stretching my brain is tiring, but
stimulating.

While I was attending LISA '07 in Dallas, I wan-
dered into the BoF run by two gents from ARIN.
Mark Kusters, CTO of ARIN, was talking to an
almost empty room about the need to start the
migration to IPv6. Granted, this was late in the
evening after a great reception (mechanical bull
riding, armadillo racing, and free drinks), but I
found myself feeling sorry for these earnest folks
who were largely being ignored. I decided to invite
Mark to write an article about the depletion of IPv4
address blocks and to dig deeper into the issue
myself.

Immediately I found other things to do. Some were
brain gym-like forays into weird, alien landscapes,
such as setting up iptables within Xen 3.0, with
bridges and unreal and virtual network interfaces.
Others didn't stretch my brain at all, because they
were familiar tasks.

I procrastinated until I came up against the wall of
a firm deadline: this issue of *;login:* was going to be
published without my column. Faced with a final
deadline, I finally cracked the books and the Web,
and got serious about IPv6.

The Next Generation

IPv6 goes way back, almost to the dawn of the
Internet. Well, not the real dawn, but to 1994,
the beginning of public awareness of the Internet.
There had already been rumblings about the fast
depletion of IPv4 addresses, and the number of In-
ternet hosts was growing at double digit rates *every*
month. IPv6 was designed not simply to create a
humongous address space, which it does, but also
to provide a more flexible framework which will
support new services such as mobility, autocon-
figuration, and improved security.

IPsec has already been integrated into IPv4, and
thus security is a less interesting reason for migrat-
ing to IPv6. Autoconfiguration is much more in-
teresting, as is the possibility of getting huge amounts
of routable address space in IPv6. No more fighting

with RIRs (Regional Internet Registries) to get a scrawny /24; register now, and get more host addresses than even Google currently needs.

I won't attempt to repeat the arguments that Kusters makes in his article. IPv4 address blocks are vanishing rapidly, and that will make it more difficult for you to get the IPv4 address space you or your organization needs. I also believe it will lead to, at the least, a gray market in IPv4 addresses, as hoarded address space gets auctioned off. Seems silly to me to get involved with another sordid affair, with domain squatters replaced with v4 address brokers, when an almost unlimited number of IPv6 addresses are available.

Instead, I'd like to point you in the direction of some resources, as well as to amuse you with my own attempts to use IPv6.

Transition

If you travel back in time far enough or are really an Internet pioneer, you will know of the original flag day. On that day (January 1, 1983), the Network Control Protocol (NCP) could no longer be used within ARPANET and TCP/IP was the only acceptable protocol. Now, imagine for a moment making a similar transition from IPv4 to IPv6.

Okay, that's long enough. We really don't want to go there, and neither did the designers of IPv6. They provided a number of transition mechanisms, including dual-stack hosts and routers and various forms of tunneling. I found a couple of books [1, 2], chose the smaller one, started reading, and quickly learned how I could start using IPv6.

Of the three methods that looked relatively easy, 6to4 tunneling interested me the most. Teredo, a method of tunneling IPv6 packets within UDP packets, has the disadvantage that its main purpose is to make IPv6 accessible to people using NAT or behind stateful firewalls. Teredo uses relay servers, one type for encapsulation and another for both registration and getting clients to set up state to the relay servers. Teredo sets up globally routable IPv6 addresses for systems behind NAT or firewalls. Microsoft has added this capability to Vista, and it can be added to XP. If you are a control freak, like firewalls, or are merely paranoid, you may wish to block this behavior [3].

Then there are tunnel brokers, organizations such as www.sixxs.net, which will match you up with tunnel providers if you are an ISP, and even set up your own tunnel right from your PC. I found myself a bit wary of this approach as well, but would have gone this route if my ISP still was set up to use this.

6to4 tunneling, on the other hand, is something you can do yourself as long as you have a public IPv4 address that you can use, and a Linux or BSD system handy. I plugged a laptop loaded with Ubuntu into the hub outside my firewall, gave it a static IP address, started hacking away . . .

And ran into problems immediately. There really isn't a lot of info about configuring Linux systems for 6to4 on the Net, and even less about debugging your setup. 6to4 tunnels IPv6 packets within IPv4 packets using protocol 41. Like Teredo, this system relies on public relays, but they work quite differently. One set of relays advertises a route to 2002::/16, and these routers convert IPv6 packets destined for your 6to4 tunnel to IPv4 packets destined for the IPv4 address of your tunnel. The other set of relays consists of systems advertising 192.88.99.1/32, an anycast route (see the February 2008 *login*: article about anycasting). These systems convert the IPv4 packets you send into IPv6 packets and forward them onto the IPv6-capable Internet.

You do need to learn something about IPv6 addresses to work with 6to4, but not a lot. Your 6to4 IPv6 address consists of the 2002::/16 prefix and your IPv4 address converted into base 16, something you can easily do with a few lines of Perl (the `Net::IP` module does the work) or even bash shell scripting:

```
IPV4=your.address.here PARTS=`echo $IPV4 | tr . ' ' `
printf "%02x%02x:%02x%02x\n" $PARTS
```

Then you follow the instructions for setting up the 6to4 tunnel for any recent Linux or BSD variant [4]. So I followed the instructions, tried `ping6 www.kame.net` (KAME is the group responsible for the BSD implementation of IPv6), and waited for the results—and waited, and waited.

Perhaps the anycast route to 192.88.99.1 doesn't work, I thought. I tried `traceroute 192.88.99.1`. This stopped before reaching the relay server—blocked by an ACL, I suppose. I asked people on other networks to try this as well. I found a couple that worked (Qwest and Sprint networks) and several that didn't (including AT&T, my upstream provider). I also noticed that some routes terminated in Europe.

But perhaps these ISPs are just blocking `traceroute`. Maybe I had other problems. I did a lookup on `www.kame.net`, and it turned out that my own ISP doesn't return the AAAA records used for IPv6 addresses. My internal DNS server does, so I just typed in one of the addresses for KAME: `2001:200:0:8000::42`. Still not working, and watching `tcpdump` output showed me that even though `ping6` claimed to be sending out packets, I sure couldn't see them.

I was convinced that I had done something wrong with the tunnel or the interface it was using. Linux kernels, like most IPv6 implementations, will automatically assign link-local addresses, beginning with `fe80::`, to interfaces, and I thought this might be the problem. But IPv6 allows you to assign multiple addresses to each interface, so `eth0` with more than one address is not the problem.

Finally, I noticed that I had misentered the command that creates the tunnel. I had carefully converted my IPv4 address into hex, then mistyped that hex when using the `ip` command, sigh. Once I got that working, `ping6` to KAME worked, and I could `ping6` a 6to4 tunnel router, `2002:c058:6301::1`, as well. Success!

The Future

Obviously, my exercise would have been a lot simpler if my own ISP offered IPv6, but it doesn't. It doesn't even support AAAA records in its DNS server.

I asked Vint Cerf, a big supporter of IPv6, when Google would start advertising AAAA records for its servers. Cerf said that hosts trying to reach Google using IPv6 might not get access because they live in a disconnected IPv6 island, but that Google is working with people on this issue.

There are loads of other issues as well. Dave Piscitello wrote a report for ICANN about support for IPv6 in commercial firewalls, as well as writing an article about it for this issue of *login*:. The answer at this point is that open source software currently has better support for IPv6 firewalling. Your Linux systems have `ip6tables`, Mac OS X has `ip6fw`, and so on. But if your organization relies on a commercial firewall product, support is sketchy.

Besides, if tunnels are available, will you ever have to move to IPv6? I believe that you will, and the sooner you start learning about IPv6, the better it will be for you. Not just avoiding the panic of a personal flag day, when you

hear that management has decided you will transition next week, but also the advantage you can personally gain by becoming familiar with technology that is going to be getting a lot more important in the near future.

Just imagine a future where most people carry around computers that are constantly in contact with the Internet. Oh, yeah, that's right, a good percentage of cell phone users already are carrying around Internet connected computers. In the US, most of these cell-phone users essentially use provider-controlled tunnels. But in China and other parts of the world, cell phones get fully routable IPv6 addresses (there is nothing like RFC1918 private address space in IPv6). There are already more cell phones in the world than IPv4 addresses. Will cell phone users want to tunnel IPv6 over IPv4 to reach your Web site?

Other than the growth of new IPv6 users, there has yet to be an IPv6 killer app. But given the issues with tunneling, as IPv6 users increase a new Internet divide, between the old and the new Internet, might arise.

I suggest taking advantage of the access you already have to computers and network devices that are IPv6-enabled, and learn now, while you are still ahead of the game.

The Lineup

I've already mentioned two articles, the first by Mark Kusters and Megan Kruse of ARIN. NAT (private network addresses: RFC 1918 [5]) and CIDR (Class InterDomain Routing, RFC 1519) have allowed us to cruise along using IPv4 without tremendous pain. And while early projections of IPv4 address depletion had us running out of addresses in 1996, today's projections are a lot more convincing. Kusters and Kruse not only discuss the dire danger, but also tell us more about getting IPv6 addresses.

Dave Piscitello, himself a networking pioneer, had mentioned to me that he had done some research on support for IPv6 in commercial firewalls. I tried some polite armtwisting, with the result that Dave has written a complete description of his research project. The news could be better, but it will get better only when customers start asking for more IPv6 support from firewall vendors.

Next up, Octave Orgeron finishes his series on working with Solaris 10 LDoms. LDoms are interesting even if you don't run Solaris and have the right hardware, because they point the way to future systems with hardware hypervisor support.

Matthew Sacks shares his experiences with working with Linux and VMware. Sacks had encountered problems with VMs crashing because they ran out of memory. He and his co-workers report on their solution here.

Aditya Sood next explains WiFi security. Sood explains its weaknesses and offers suggestions for better WiFi security.

Michael McCool then writes about the issues in achieving high performance on hardware that supports parallel execution. McCool begins by describing the various CPU features that support parallelism, starting with the obvious ones such as multicore processors. But he goes much deeper than that, in the first of several articles we hope to publish about parallel programming.

Filling out the magazine, we have our regular columnists, but no summaries. Strangely enough, no one seems interested in attending conferences or workshops over Christmas vacation, and even shortly thereafter, so this issue of *;login:* is a bit shorter than usual. Have no fear: the next issue will

include conference summaries from FAST '08 and the 2008 Linux Storage & Filesystem Workshop.

REFERENCES

- [1] Niall Murphy and David Malone, *IPv6 Network Administration* (O'Reilly, 2005).
- [2] Marc Blanchet, *Migrating to IPv6: A Practical Guide for Mobile and Fixed Networks* (Wiley and Sons, 2005).
- [3] Teredo security considerations: http://en.wikipedia.org/wiki/Teredo_tunneling#Security_considerations; http://www.microsoft.com/technet/network/ipv6/ipv6_teredo.mspix.
- [4] Setup of 6to4: http://www.getipv6.info/index.php/Linux_or_BSD_6to4_Relays; <http://tldp.org/HOWTO/Linux+IPv6-HOWTO/configuring-ipv6to4-tunnels.html>.
- [5] RFC 1918, address allocations for private networks: <http://www.faqs.org/rfcs/rfc1918.html>.

MARK KOSTERS AND MEGAN KRUSE

IPv6: it's time to make the move



Mark Kusters is currently serving as the Chief Technology Officer for ARIN. Before coming to ARIN, Mark worked for VeriSign/Network Solutions for over 16 years in various positions, starting as a software engineer and exiting as a vice president.

markk@arin.net

Megan Kruse is the Public Affairs Officer for ARIN, focusing primarily on community outreach and public relations. Before ARIN, she had similar roles with other nonprofit organizations.

megank@arin.net

WHEN ENGINEERS DEPLOYED IPV4 IN 1981, four billion IP addresses seemed like plenty. As the world caught on to the commercial possibilities of the Internet, though, engineers realized that the number of IP addresses simply wasn't enough for all the laptops, mobile devices, Web servers, routers, and other devices coming online. IPv6, the new numbering system, enlarges the address pool drastically, but it is unfamiliar and relatively unused so far. In this article, we show that, with only about 16% of the IPv4 address space remaining, the world will run out of IPv4 address blocks within a few years, and we suggest what you can start doing now to prepare for the transition to IPv6.

IPv4 History

In the late 1960s, various U.S. universities and research centers needed a way to connect their computers together to access each other's resources. At that point, all interconnection technologies were proprietary, required homogeneous equipment, and were very expensive to deploy.

The Advanced Research Project Agency, a U.S. government organization, developed a network called ARPANET, incorporating interconnection and the ideas of the design, implementation, and use of network techniques in general and packet-switching in particular. The company Bolt, Beranek and Newman (BBN) had ARPANET operational by 1971, but two years later the existing lower-layer protocols had become functionally inadequate. The improvement goals were to be independent from underlying network techniques and from the architecture of the host; to have universal connectivity throughout the network; to provide end-to-end acknowledgments; and to standardize application protocols.

TCP/IP (IP version 4) was fully implemented in 1983. The success of TCP/IP has been based largely on three factors: (1) the 1983 University of California at Berkeley implementation of TCP/IP placed into the public domain, leading to free implementations others could use; (2) the National Science Foundation's interconnections of various U.S. educational institutions and international players; and (3) decreased interconnection costs.

By 1992, the world was beginning to realize the advantages of the Internet. More and more companies and users wanted to connect to the Internet, leading to increased demand for IP addresses. Until 1992, sites that connected to the Internet received allocations based on class: class A (16,777,214 maximum possible addresses), Class B (65,534 maximum possible addresses), and Class C (254 maximum possible addresses). For many, Class A was too big and Class C was too small, leading to a large demand for Class B addresses. This particular scheme promoted a lot of waste, and engineers consequently created the Classless Inter-Domain Routing (CIDR) scheme, allowing allocations to be based on bit boundaries. For example, a Class C is now considered a /24, a Class B is now a /16, and a Class A is now a /8. This allowed IP allocation agencies to hand out “right-sized blocks” to ISPs who requested space. If an ISP required a /19, they could get it—not a /16, as was happening in the classful days [1].

IPv4 Depletion

Although the introduction of CIDR slowed the consumption of IPv4 address space, continued global demand still makes IPv4 depletion inevitable. Figure 1 shows the global depletion of IPv4 address space over the past three years. The Internet Assigned Numbers Authority (IANA) allocates address space blocks in /8 increments to the five Regional Internet Registries (RIRs) [2] that manage the distribution of address space to Internet Service Providers (ISPs), large organizations, universities, and other entities. There are 256 /8s in the entire IPv4 pool. As of December 2007, there are 42 /8s remaining, or 16.4%.

The Regional Internet Registries have collectively allocated about 10 /8s of IPv4 address space each year, on average. If that trend continues unchanged, IPv4 address space will be fully depleted by 2011. This scenario assumes that demand does not increase, which is unlikely, given the ever-increasing number of Internet-enabled devices. This scenario also assumes no industry panic (hoarding, withholding, etc.), no IANA or RIR policy changes, and no other external factors influencing address space allocations, any of which could push the IPv4 depletion date earlier.

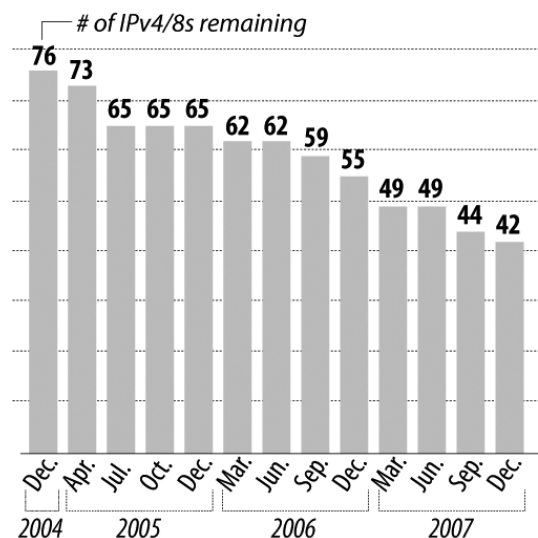


FIGURE 1: IPV4 ADDRESS BLOCK DEPLETION FROM DEC. 2004 TO DEC. 2007

Figure 2 shows allocations from RIRs to ISPs and other large entities by year. APNIC (Asia Pacific), ARIN (North America), and RIPE NCC (Europe) have allocated the most space to date. LACNIC (Latin America) and AfriNIC (Africa) follow, as newer RIRs are growing the infrastructure in their regions. The 2007 data in Figure 2 is as of 30 September, putting the numbers on track to exceed allocations from 1999 through 2006.

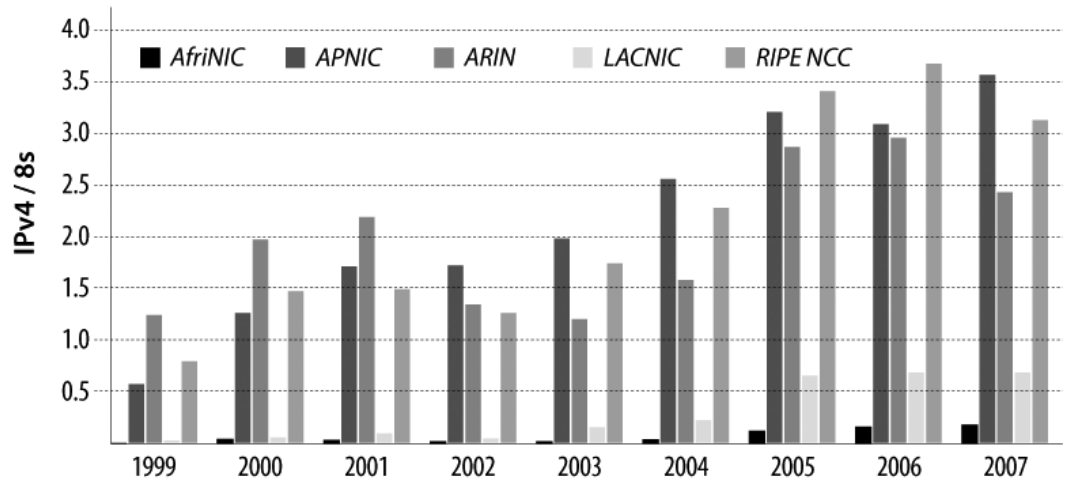


FIGURE 2: IPV4 ALLOCATIONS FROM REGIONAL INTERNET REGISTRIES

IPv6 Arrives

Once the Internet became a commercial success, demand for IP address space increased dramatically. Based on increased demand, various technologists in the Internet Engineering Task Force (IETF) studied demand and possible depletion of IPv4 addresses. From those studies, the IETF started looking at a replacement for IPv4 called IPng (IP next generation).

Debates over various technologies within IPng resulted in IPv6 as the replacement technology to supersede IPv4. IPv6 promised many things, many of which were back-ported to IPv4. The biggest issue IPv6 solves is the growth of the address space—from 32 bits, or four billion addresses, to 128 bits, an astronomical 16 billion-billion addresses. Although still a finite space, IPv6 should provide enough addresses for a very long time.

With available /8 address blocks diminishing and annual allocations increasing, the American Registry for Internet Numbers (ARIN) is now actively advising the Internet community that transitioning to IPv6 is necessary for any applications that require ongoing availability of contiguous IP address space.

ARIN, the Regional Internet Registry that manages the distribution of IP addresses in Canada, many Caribbean and North Atlantic islands, and the United States, cannot and will not force anyone to transition to IPv6. However, soon organizations that require larger contiguous blocks of IP address space will only be able to receive them in IPv6. In the meantime, ARIN will continue to issue IPv4 address space as available; its distribution practices will change only when its community creates or revises policies.

Recognizing the inevitability of IPv4 depletion, on 7 May 2007 the ARIN Board of Trustees passed a “Resolution on Internet Protocol Number Resource Availability” [3]. In addition to advising the community on IPv6 transitioning, the resolution directs ARIN staff to heighten its efforts to verify the authenticity of IPv4 resource requests and asks that ARIN’s elected pol-

icy body, the Advisory Council, consider working with the community on policy changes to encourage transition to IPv6.

To implement this resolution, ARIN has reviewed its internal resource request procedures, sent progress announcements to the community, produced new educational documentation, and focused on IPv6 in many of its general outreach activities, such as speaking engagements, trade shows, and technical community meetings.

Current Allocation Policies

ARIN and the other RIRs have community-defined policies that dictate how they distribute IPv4 and IPv6 address space within their regions. In the ARIN region, these community decisions are recorded as policies in the Number Resource Policy Manual [4].

Table 1 shows ARIN's current allocation policies for IPv4 and IPv6, for both ISPs and end users. This table is current as of December 2007 but is subject to change. See the Policy section on the ARIN Web site for the most up-to-date policy set.

<p>Initial allocation varies from a /22 (1,024 addresses) to a /20 (4,096 addresses); larger amounts are possible in some cases.</p> <p>Eligibility is as follows:</p> <p>For a /22: efficient utilization of a /23 from upstream; intent to multihome; agree to renumber</p> <p>For a /21: efficient utilization of /22 from upstream; intent to multihome; agree to renumber</p> <p>For a /20: efficient utilization of /21 from upstream; intent to multihome; agree to renumber</p> <p>Efficient utilization of /20 from upstream (no renumbering required)</p>	<p>Minimum initial allocation is a /32 (296 addresses); larger amounts are possible in some cases.</p> <p>Eligibility requires:</p> <p>being an ISP</p> <p>routing the aggregate</p> <p>having a plan to make at least 200 /48 assignments to other organizations within five years</p>
<p>Minimum assignment is a /22 (1,024 addresses); larger amounts are possible in some cases.</p> <p>Eligibility is based on:</p> <p>Current and planned utilization (25% immediate utilization and 50% utilization within one year)</p> <p>Multihoming state</p>	<p>Minimum assignment is a /48 (280 addresses); larger amounts are possible in some cases.</p> <p>These assignments come from a distinctly identified prefix and are made with a reservation for growth of at least a /44.</p> <p>Eligibility is based on:</p> <p>Being an end user</p> <p>Qualifying for an IPv4 assignment or allocation from ARIN under current IPv4 policy</p> <p>In other words, if you could get IPv4 space, then you can get IPv6 space.</p>

TABLE 1: IPV4 AND IPV6 ALLOCATION CRITERIA

Getting an initial allocation of IPv6 address space from ARIN is a relatively simple and straightforward process.

- Step 1: Review the requirements for IPv6 in ARIN's Number Resource Policy Manual [5].
- Step 2: Complete and submit the appropriate forms, including contact and organization identifiers (if new to ARIN) and the IPv6 Network Request Template [6].
- Step 3: Correspond with ARIN's Registration Services Department to obtain answers to any questions or required documentation.
- Step 4: Receive approval from ARIN.
- Step 5: Pay any required fees and sign the Registration Services Agreement.
- Step 6: Receive allocation from ARIN.

IPv6 Implementation

ARIN and the other Regional Internet Registries have distributed IPv6 address space since 1999. As shown in Figure 3, as of 30 September 2007 RIPE NCC in Europe is responsible for nearly half of the IPv6 prefixes that have been allocated; APNIC in the Asia-Pacific region and ARIN in North America are responsible for just under one-quarter each; and AfriNIC in Africa and LACNIC in South America handle the remainder.

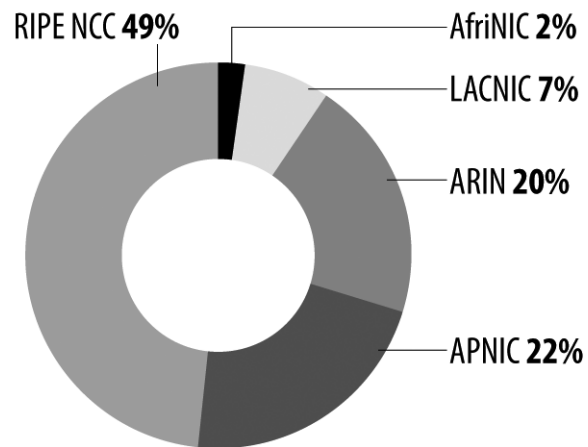


FIGURE 3: IPV6 ALLOCATIONS FROM REGIONAL INTERNET REGISTRIES

IPv6 has not yet really taken hold, with only a few hundred prefixes allocated globally. Over the past three years, the IPv6 routing table has grown from 400 independent address blocks with announced routes to around 1,000. In comparison, there are nearly 250,000 IPv4 routes.

Lack of education, ubiquitous use of NATs, and uncertainty about the costs involved in transitioning have all made IPv6 slow to deploy. Specific transition needs and costs vary based on many factors, but may involve:

- Obtaining IPv6 addresses and connectivity
- Upgrading operating systems, software, and network management tools
- Updating routers, firewalls, and other hardware that is “middleware”
- Training IT staff and customer service representatives

Why Should You Begin the Transition Today?

IPv6 IS READY

IPv6 is stable and well tested and already deployed in locations across the globe. Because both IPv4 and IPv6 will coexist and work simultaneously for years to come, applications that currently work with IPv4 will continue to work in a dual-stacked network. Transition costs vary by situation, but planning ahead by getting IPv6-ready equipment in regular purchasing cycles will help minimize costs.

IMPENDING IPv4 DEPLETION

Only about 16% of the IPv4 address pool remains, and that percentage gets smaller every month. As IPv4 demand increases and available address space decreases, organizations will soon have no choice but to continue network operations in IPv6. The sooner you learn and deploy IPv6, the farther ahead of your competition you will be.

CUSTOMER REACH

Right now, people are trying to reach you through IPv6. At this point, the user is choosing to use IPv6 and will use IPv4 by default when the IPv6 query fails. Once the IPv4 address pool is depleted, people will try to reach you through IPv6 only. In a few years, if a user's IPv6 query fails, that user will not be able to communicate with you and you will lose business.

Your Next Steps

There are several practical steps you can take toward the transition to IPv6.

Replace any outdated equipment and software with IPv6-ready devices and applications. Encourage vendors to support IPv6, and specifically include IPv6 support in RFPs and contracts.

Send your IT staff to IPv6 training seminars and encourage them to read forums such as the ARIN IPv6 Wiki or to get involved in organizations such as the IETF or the North American Network Operators' Group (NANOG) to learn from other engineers already deploying IPv6 in their networks.

Talk to your ISP about getting IPv6 service. If it cannot provide such service, experiment with tunneling IPv6 over IPv4 with tools such as Teredo or TSP. Start by looking at listings of freely available tunnel brokers: There are multiple brokers on the Internet that can cater to specific needs. For example, one user-friendly broker that has helper applications and configurations packaged for various platforms is go6 [7]. The service is free and has client applications that completely automate tunnel configurations on Windows, Linux, FreeBSD, and OS X.

Additionally, you may have had to undergo the excruciating exercise of renumbering in IPv4—moving from one set of IP addresses to another. These renumbering exercises have been and continue to be painful, especially in networks that are poorly documented and have hidden address dependencies. Although in some ways IPv6 has made renumbering easier, ARIN recommends that organizations design their networks to allow for easy renumbering, as you will have a clean slate to build upon. ARIN also recom-

mends that upstream providers who receive a /32 prefix directly from the RIR enter into contractual arrangements with their customers stipulating that the address space may have to be returned, requiring all end-sites to be renumbered.

Summary

With the IPv4 address space decreasing and demand for IP addresses increasing, now is the time for you to begin the transition to IPv6. Getting IPv6 address space from ARIN is a simple process, and the faster you learn about IPv6 and prepare your network, the further ahead of your competition you will be and the better prepared you will be to handle requests from *all* of your customers well into the future.

More Information

ARIN hosts an IPv6 Wiki site [8] to facilitate discussion and information-sharing on IPv6 topics and issues. Its purpose is to provide interested individuals with an opportunity to collaborate on IPv6, with specific focus on implementation and migration to IPv6 in the ARIN region.

More information about IPv6, including general educational materials, specific registration services information, and contact information, is available from the IPv6 Information Center [9]. You can also visit the main ARIN Web site at www.arin.net or email us at info@arin.net.

ABOUT ARIN

The American Registry for Internet Numbers is a nonprofit corporation that distributes Internet number resources, including both IPv4 and IPv6 address space, to Canada, many Caribbean and North Atlantic islands, and the United States.

REFERENCES

- [1] A chart showing the number of unique IP addresses possible in classful addressing, IPv4, and IPv6 is available at http://www.arin.net/education/IP_Address_Block_Size_Equivalents.pdf.
- [2] More information on the Regional Internet Registry system is available at <http://www.arin.net/community/>.
- [3] The ARIN Board Resolution is available at <http://www.arin.net/v6/v6-resolution.html>.
- [4] The Number Resource Policy Manual is available at <http://www.arin.net/policy/nrpm.html>.
- [5] IPv6 policies are available at <http://www.arin.net/policy/nrpm.html#six>.
- [6] All templates are available at <http://www.arin.net/registration/templates/index.html>.
- [7] <http://www.go6.net>.
- [8] <http://www.getipv6.info>.
- [9] <http://www.arin.net/v6/v6-info.html>.

DAVID PISCITELLO

are commercial firewalls ready for IP version 6?



Dave Piscitello is a Senior Security Technologist for ICANN. A 30-year Internet veteran, Dave currently serves on ICANN's Security and Stability Advisory Committee.

dave.piscitello@icann.org

THE DEPLETION RATE OF THE IP VERSION 4 (IPv4) address space has been the subject of considerable analysis and even greater speculation for nearly 15 years. However, Network Address Translation [1, 2] and classless inter-domain routing (CIDR [3]) have extended the lifespan of the IPv4 address space beyond many projected exhaustion dates. Today, many organizations still choose to dismiss experts who voice IPv4 addressing concerns as modern-day “boys who cry wolf.” Whether we are perilously close to the day when ignoring the cries will prove fatal to the flock remains an open question. Assuming that exhaustion of the IPv4 address space is imminent, we consider whether the community will be able to secure networks when we are left with little choice but to deploy IPv4’s successor, Internet Protocol version 6.

IPv4 Lifetime Projections

In 2005, Tony Hain of Cisco Systems applied several mathematical models to project IPv4 address lifetime [4] (see Figures 1 and 2) and concludes, “Depending on the model chosen, the nonlinear historical trends . . . covering the last 5- and 10-year data show that the remaining 64 /8s will be allocated somewhere between 2009 and 2016, with no change in policy.”

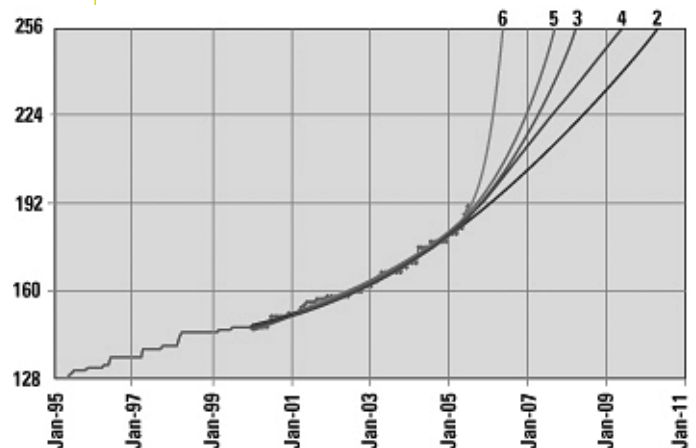


FIGURE 1: IPV4 LIFETIME PROJECTIONS FOR ORDER-N POLYNOMIALS, POST-2000 HISTORY BASIS

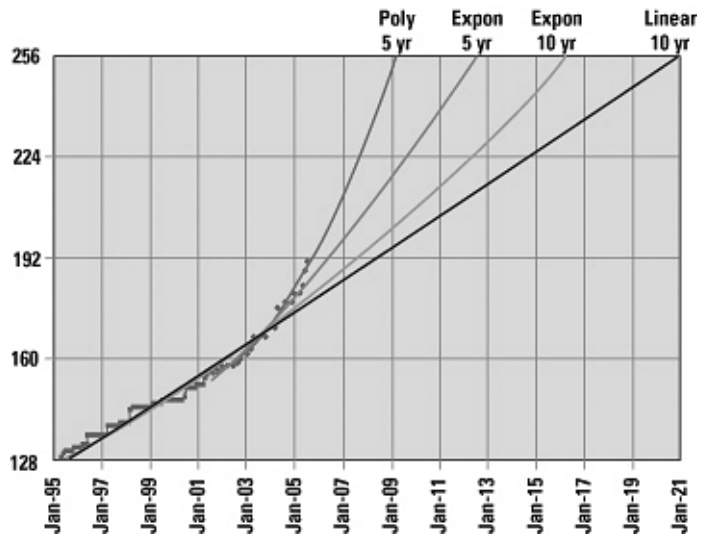


FIGURE 2: IPV4 LIFETIME PROJECTIONS FOR POLYNOMIALS AND EXPONENTIALS

These projections appear to be spot on; in particular, Geoff Huston, a respected authority on IPv4 routing and addressing, offered that “these different predictive approaches yield slightly different outcomes, but not beyond any reasonable error margin for predictions of this nature. Sometime in the forthcoming 5 to 10 years the current address distribution policy framework for IPv4 will no longer be sustainable for the current industry address consumption model because of effective exhaustion of the unallocated address pool.” (Bear in mind that his comments were offered in 2005.) The Cooperative Association for Internet Data Analysis (CAIDA) has an equally sobering projection: “If current consumption rates continue unchanged (a wholly unwarranted assumption) and little of the already allocated space is ever reclaimed (a realistic assumption), then Internet Assigned Numbers Authority’s (IANA) unallocated IPv4 pool and currently reserved spaces would run dry in March 2009”[5].

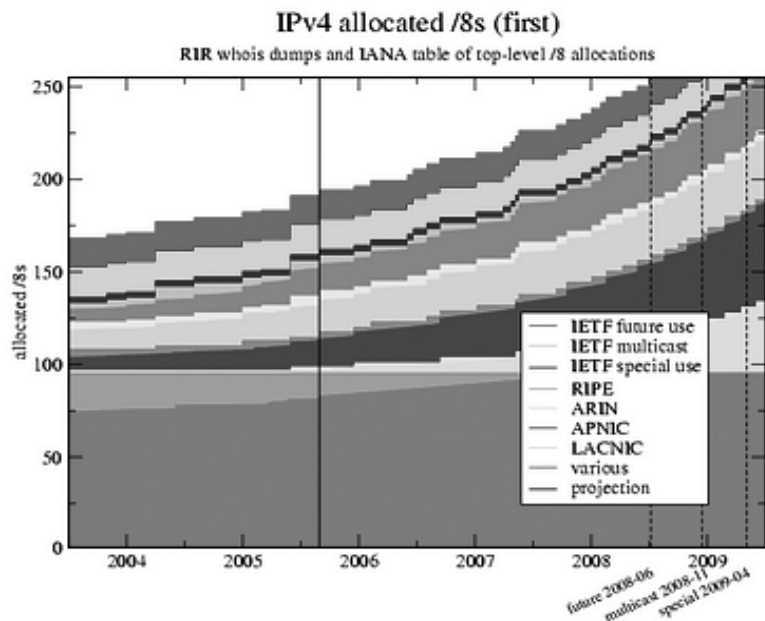


FIGURE 3: ALLOCATED IPV4 ADDRESS SPACE

If you doubt the accuracy of these claims, look at the allocation of IPv4 address space as of 28 October 2007 [6] (see Figure 3). Regional Internet registries (RIRs) are struggling to allocate contiguous address blocks of sufficient size to service providers. Proposals to reclaim unused (or “hoarded,” as some claim) IPv4 address space remain nonstarters for operational and legal reasons; for example, attempts to use the RFC 1700 experimental space (known as Class E addresses) will prove problematic for some IPv4 implementations, and there is no legal basis for recovering previously allocated address space. More important, if the projections are accurate, reclamation will not happen fast enough to have an impact.

The only practical way forward is to deploy IP version 6. Claims that IPv6 adds nothing that has not been added to IPv4 notwithstanding, the one indisputable fact about the next-generation Internet Protocol is that it does provide more address space. But at what cost? IPv6 standards and implementations are available, but they are little used, and little is known about the availability of security products and services. Will relieving the addressing problem put organizations in a position where they will not be able to provide the same security baseline for IPv6 networks that they currently are able to do for IPv4 networks?

A security baseline encompasses many policies, practices, operations, and technologies. Any thorough analysis would undoubtedly span multiple studies, involve detailed product testing, and require considerable resources. However, a survey that limits the scope of the question to “Can a commonly deployed security product provide the same breadth of security policy enforcement for IPv6 networks as it does for IPv4 networks?” may provide a useful reference point for the Internet community.

ICANN’s Security and Stability Advisory Committee (SSAC) considered candidate security systems for such a study and concluded that Internet firewalls would serve the purpose well. Firewalls are among the oldest and most commonly employed security technologies and are still considered critical components of security deployment. Thus, we should be able to gain meaningful insight into the state of IPv6 readiness of the Internet security industry by studying firewalls.

Methodology

We compiled a list of commercial firewall vendors to survey using search engines, portals that list security products and vendors (e.g., network intrusion [7] and Rik Farrow’s firewall product selector [8]), and contact lists compiled by ICSA Labs [9]. This survey only includes commercial firewall products and in particular does not include personal firewall software or open source firewall libraries that could be installed and configured on PC and server platforms. The survey also excludes broadband access routers that only provide rudimentary firewall features. We collected information to identify the features we would survey using vendor publications (Web sites, white papers, product specifications, and administrative and user manuals). To further shape the survey, we consulted with firewall administrators and security experts for additional input. Ultimately, we chose to include both networking and security features that we believe to be commonly used at firewalls to enforce security policy in IPv4 networks, and we agreed that it would be useful to study security feature availability according to three market segments: small office/home office (SOHO), small and medium business (SMB), and large enterprise/service provider (LE/SP). Finally, we chose to

keep the number of survey questions small and the degree of technical specificity low, with the expectation that this would increase our response rate.

We contacted firewall vendors using general contact email addresses and telephone numbers. We also solicited direct technical contact information from firewall vendors by posting a general inquiry to popular firewall and security mailing lists (e.g., bugtraq@securityfocus.com, pen-test@securityfocus.com, firewall-wizards@listserv.icsalabs.com). We corroborated vendor responses by contacting multiple parties within each company, experts at large, colleagues at reputable testing laboratories, or firewall administrators. Whenever available, we consulted vendor documentation (e.g., configuration and administration guides that were accessible via a vendor's technical support Web portal).

It is important to note that we did not conduct formal testing of any product included in this survey. Our objective was to gauge feature availability, not to qualify or certify any product as being IPv6 "security capable." We relied on the accuracy of available documentation, the expertise of administrators we consulted, and, ultimately, on vendor contacts acting in good faith. We have no reason to believe that any party contacted misrepresented IPv6 feature availability to us; in fact, the majority of correspondence was earnest and involved numerous dialogues beyond the initial survey query and response: Overall, vendors were eager for input that helps prioritize product development or shapes an opportunity for expanding market share and were eager to cooperate.

Survey Results

We obtained survey responses and compiled complementary information for 42 of 60 products from commercial firewall vendors. Several vendors identified a single product as satisfying multiple market segments, resulting in 81 product placements across the three defined market segments. Specifically, 19 results were collected for SOHO products, 35 for SMB products, and 27 for the LE/SP market. In this article, we present a subset of the results. Complete details are available in SAC 021, "Survey of IPv6 Support in Commercial Firewalls" [10].

[Note: In the charts, we label the bars representing these respondents with ALL, SOHO, SMB, and LE/SP based on the unique totals for each segment (i.e., percentages are based on 42, 19, 35, and 27, respectively).]

How broadly are IPv6 transport and routing supported by commercial firewalls?

Many organizations will be able to obtain ample IPv6 address space [11] and will want to take advantage of autoconfiguration and other IPv6 addressing features. Firewalls in such deployments must be able to forward IPv6 traffic between internal and external interfaces. (Note that the ability to encapsulate IPv4 datagrams arriving from internal networks as payloads in IPv6 datagrams and forward these to IPv6 destinations is considered separately in the full report; see [10].) All firewalls surveyed support IPv4 transport. Figure 4 illustrates that IPv6 transport is supported in fewer than one in three of the firewalls surveyed.

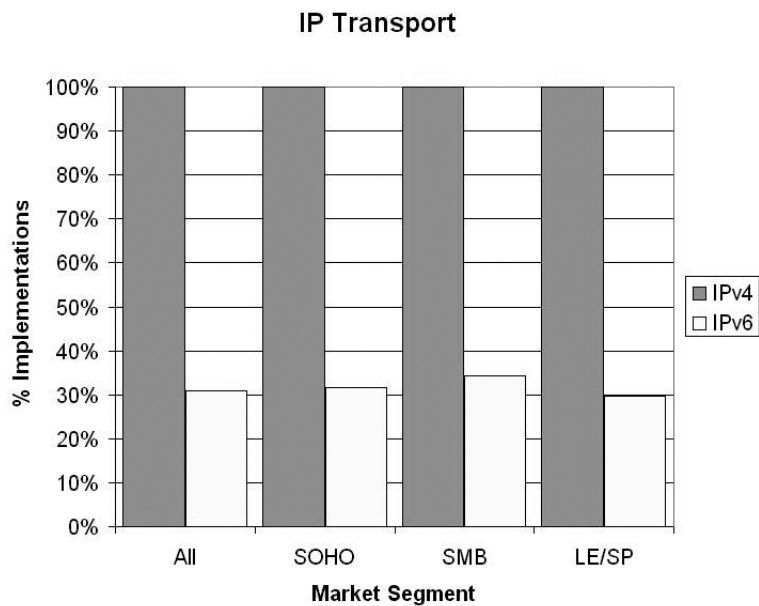


FIGURE 4. FIREWALL SUPPORT FOR IPV4 AND IPV6 TRANSPORT

Firewall systems (as opposed to routers that support certain firewall features) are often used in complex topologies that are designed to satisfy an organization’s redundancy, failover, and high-availability needs. Such organizations may run firewalls in transparent or bridging mode, or they may choose to have the firewall participate as a peer in an adaptive routing or neighbor discovery protocol. Figure 5 illustrates support for neighbor discovery and peer routing protocols.

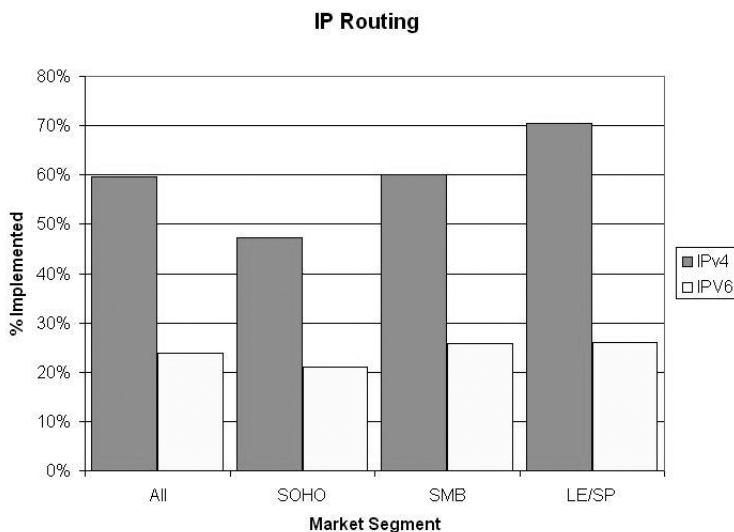


FIGURE 5. FIREWALL SUPPORT FOR IPV4 AND IPV6 ROUTING

Sixty percent of the 42 firewall products surveyed can peer in IPv4 routing exchanges or perform neighbor discovery, but only 24% can peer when IPv6 is used. The results suggest that an organization would have limited choices if it intended to include a firewall in a topology where adaptive recovery from link failure is required. As one might expect, little support exists among SOHO products that are typically deployed in single and “stub” networking topologies.

What types of IPv6 traffic inspection and policy enforcement are available on commercial firewalls?

Commercial firewalls are commonly used to enforce a security policy on traffic that passes between an organization's internal networks and external networks. Three forms of traffic inspection are available when IPv4 transport is used: static packet filtering, stateful packet inspection, and application-layer inspection. We surveyed these individually.

Static packet filtering is the most basic form of security policy enforcement firewalls provide; it is used even when more advanced inspection methods are available (e.g., to enforce a policy on a new protocol or application). This method inspects each arriving IP packet individually. If the packet complies with the security policy, it is allowed to pass through the firewall; if not, it is typically blocked and (silently) discarded.

Ninety-five percent of the commercial firewalls surveyed provide static packet filtering in all market segments when IPv4 transport is used. Twenty-nine percent provide static filtering when IPv6 transport is used (see Figure 6).

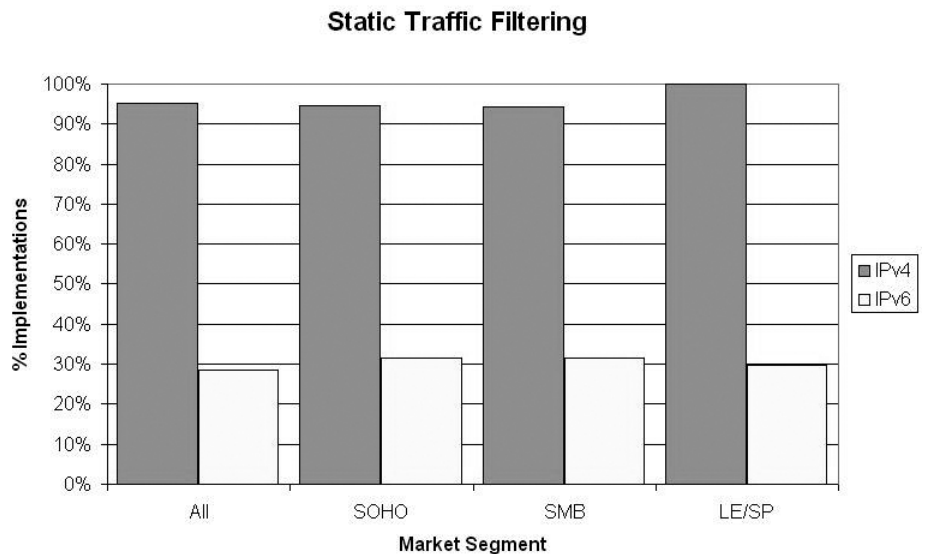


FIGURE 6. FIREWALL SUPPORT FOR IPV4 AND IPV6 STATIC PACKET FILTERING

Stateful inspection of IP layer packets is a more advanced form of security policy enforcement. Stateful inspection considers all IP datagram payloads associated with a given TCP connection, UDP stream, or application datum and enforces a policy on multipacket and complete traffic flows. Ninety percent of commercial firewall products surveyed provide stateful inspection when IPv4 transport is used, whereas only 23% do so when IPv6 transport is used (see Figure 7). (Note that firewalls capable of supporting stateful packet inspection typically support static packet filtering, and this appears to be true for both IPv4 and IPv6. We also observed from the results that if a product supports IP transport and one or more forms of traffic inspection, that product supports IPsec for IPv4 and IPv6 transport. These observations are discussed in some detail in [10].)

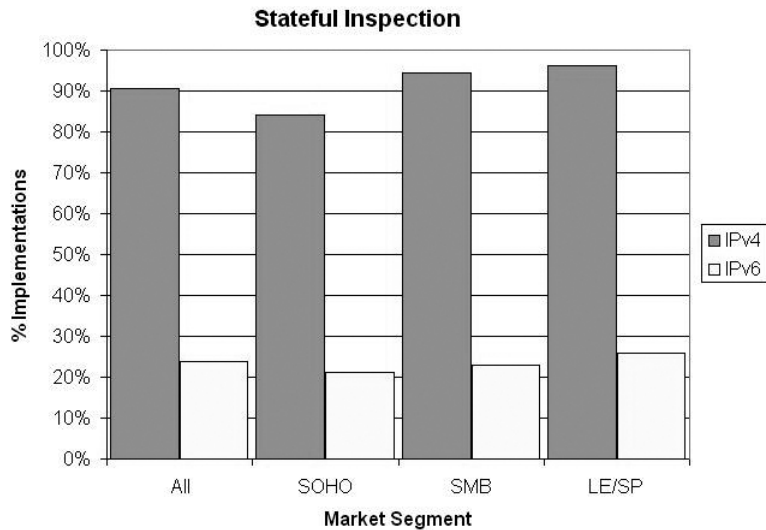


FIGURE 7. FIREWALL SUPPORT FOR IPV4 AND IPV6 STATEFUL INSPECTION

Increasingly, organizations expect firewalls to protect Web, email, DNS, and other Internet servers and clients from exploitation and privilege escalation attacks. Firewall vendors use application-layer gateways (proxies) or stateful traffic inspection techniques to detect and block malicious traffic that can cause an application or system to fail, respond incorrectly, disclose sensitive data, or allow unauthorized parties to gain administrative control over a system. In the survey, we were agnostic about the method used and simply asked whether vendors provide application-level inspection.

Eighty-one percent of commercial firewalls surveyed can apply stateful inspection or proxy techniques to application-level traffic when IPv4 transport is used, but only 17% are able to do so when IPv6 transport is used (see Figure 8).

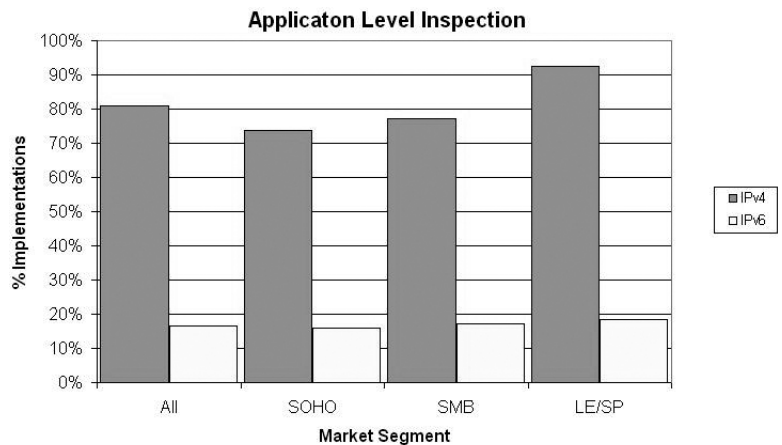
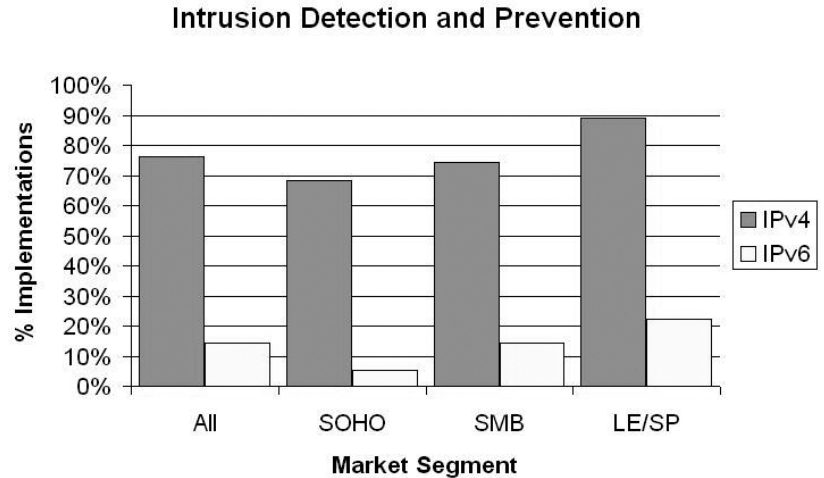


FIGURE 8. FIREWALL SUPPORT FOR IPV4 AND IPV6 APPLICATION-LEVEL INSPECTION

Do commercial firewalls provide intrusion detection or intrusion prevention when IPv6 transport is used?

Firewalls are in-line devices and are designed to detect and prevent attacks by blocking traffic or stripping objectionable content prior to forwarding traffic to a destination. Certain commercial firewalls incorporate detection and mitigation techniques to protect an organization from sophisticated



network, transport, and application attacks (“intrusions”). These firewalls may provide one or combinations of signature- and anomaly-based detection methods as adjunct services to the three forms of traffic inspection described earlier.

FIGURE 9. INTRUSION DETECTION AND PREVENTION SERVICES

Figure 9 shows that 76% of commercial firewall products surveyed provide some form of intrusion detection or prevention when IPv4 transport is used. Only 14% offer IDS/IPS when IPv6 transport is used. We note that some vendors commented that the signature sets for IDS/IPS inspection engines for IPv6 were not as extensive as the signature sets for IPv4. (The very low availability of IDS/IPS among SOHO products biases this result. The survey result for LE/SP products is perhaps a more accurate representation of IDS/IPS availability when IPv6 transport is used for organizations that require such features.)

Do commercial firewalls provide (distributed) denial-of-service protection when IPv6 transport is used?

Flooding forms of denial-of-service (DoS) attacks attempt to exhaust the resources of a targeted application, system, or network and thus deny service to users. Whereas exploitation attacks can deny service to users, flooding attacks are familiar to most Internet users and thus represent a marketing opportunity. For this reason, we chose to survey protection against flooding separately from IDS/IPS. A higher percentage of commercial firewalls offer some form of rate-limiting when DoS and DDoS attacks are detected than offer IDS/IPS protection when IPv6 transport is used, but generally support is still relatively weak (see Figure 10). We note that some vendors indicated that DoS protection is not as comprehensive when IPv6 transport was used (i.e., fewer kinds of DoS attacks are mitigated or reduced).

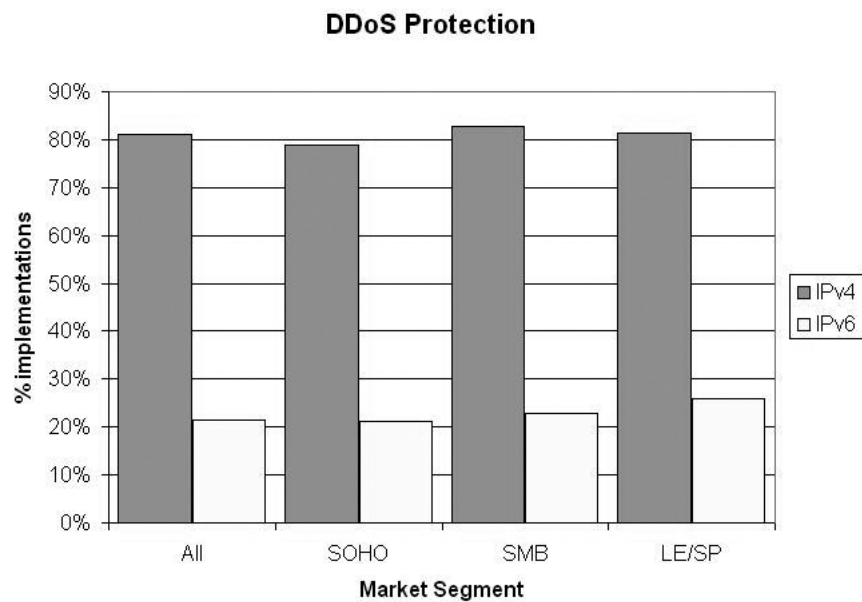


FIGURE 10: DDOS PROTECTION

Conclusions

IP version 6 transport is not broadly supported by commercial firewalls. If organizations attempt to “go native IPv6” today, they will be limited to choosing among the 31% of the firewall products surveyed that support IPv6 transport. We do note, however, that although fewer than one in three products support IPv6 transport and a desirable set of security features, support among the firewall market share leaders improves this figure somewhat. This observation is consistent with recent *Network World* product testing conducted by Dr. Joel Snyder [12].

We find the limited support for IPv6 stateful packet inspection across the commercial firewall product sector quite worrisome. Many vendors extend stateful packet inspection techniques to provide additional application-level protection measures. We also find another cause for concern in the limited availability of IPv6 support at the “periphery” of the Internet. Support for advanced security features is weakest in SOHO and SMB segments, although we did not include broadband access devices that claim firewall capabilities in our survey. Such devices have very little, if any, firewall capability beyond static packet filtering. We speculate that support is no stronger in the broadband market than in SOHO, and we speculate further that if we had included such devices, the overall results of IPv6 support among commercial firewall and “router/firewall” products would have been even more discouraging.

We conclude by quoting from our report:

Internet firewalls are the most widely employed infrastructure security technology today. With nearly two decades of deployment and evolution, firewalls are also the most mature security technology used in the Internet. They are, however, one of many security technologies commonly used by Internet-enabled and security-aware organizations to mitigate Internet attacks and threats. This survey cannot definitively answer the question, “Can an organization that uses IPv6 transport enforce a security policy at a firewall that is commensurate to a policy currently supported when IPv4 transport is used?” The survey results do suggest that an organization that

adopts IPv6 today may not be able to duplicate IPv4 security feature and policy support.

A comment we heard all too frequently and from altogether too many commercial firewall vendors during our study was, “No one’s asking for IPv6.” Markets can turn quickly, but not overnight. If we begin asking commercial firewall vendors *soon* we might expect the availability of IPv6 support to improve within the next 9–18 months. If the available IPv4 address pool evaporates faster, some organizations may experience difficulties satisfying security policies with the commercial firewalls they currently employ.

REFERENCES

- [1] RFC 1631, The IP Network Address Translator (NAT), K. Egevang and P. Francis: <http://www.faqs.org/rfcs/rfc1631.html>.
- [2] RFC 2663, NAT Terminology and Considerations, P. Srisuresh and M. Holdrege: <http://www.faqs.org/rfcs/rfc2663.html>.
- [3] RFC 1519, Classless Inter-Domain Routing (CIDR), V. Fuller, T. Li, J. Yu, and K. Varadhan: <http://www.faqs.org/rfcs/rfc1519.html>.
- [4] Reprinted from T. Hain, “A Pragmatic Report on IPv4 Address Space Consumption,” *The Internet Protocol Journal*, 8, 3. *IPJ* is a quarterly technical journal published by Cisco Systems. See http://www.cisco.com/web/about/ac123/ac147/archived_issues/ipj_8-3/ipv4.html.
- [5] IPv4 Consumption: <http://www.caida.org/research/id-consumption/ipv4/>.
- [6] Internet Protocol v4 Address Space: <http://www.iana.org/assignments/ipv4-address-space>.
- [7] Computer Network Defense, Ltd., Talisker firewalls overview page: <http://networkintrusion.co.uk/firewall.htm>.
- [8] Rik Farrow’s firewall product tester page: <http://www.spirit.com/cgi-new/report.pl?dbase=fw&function=view>.
- [9] ICSA Labs certified firewalls list: <http://www.icsalabs.com/icsa/product.php?tid=fghhf456fgh>.
- [10] SAC 021, Survey of IPv6 Support in Commercial Firewalls: <http://www.icann.org/committees/security/sac021.pdf>.
- [11] Guidelines—Initial IPv6 Allocation from ARIN: http://www.arin.net/registration/guidelines/ipv6_initial_alloc.html.
- [12] UTM and IPv6: Do they mix? J. Snyder: <http://www.networkworld.com/reviews/2007/111207-utm-firewall-test-ipv6.html?nwwpkg=utm>.

OCTAVE ORGERON

an introduction to logical domains



PART 4: ADVANCED TOPICS

Octave Orgeron is a Solaris Systems Engineer and an OpenSolaris Community Leader. Currently working in the financial services industry, he also has experience in the e-commerce, Web hosting, marketing services, and IT technology markets. He specializes in virtualization, provisioning, grid computing, and high availability.

unixconsole@yahoo.com

IN THE FEBRUARY 2008 ISSUE OF *login*, I discussed advanced topics concerning networking and storage for logical domains (LDoms). In this final article of the series, I will discuss other advanced topics that are key to the successful management of logical domains. These advanced topics will aid your design and implementation decisions concerning LDoms.

Hardware and LDoms

Since I started writing this series of articles, Sun has continued to release new hardware that supports LDoms. This includes UltraSPARC-T2 rack-mount servers such as the T5120 and blade servers such as the T6320 [1]. LDom-capable equipment will continue to increase with the release of new sun4v platforms, such as the “Victoria Falls” and “ROCK” platforms [2]. This will help flesh out the medium- to high-end LDom-capable platforms and provide a full range of equipment to choose from.

Currently, the UltraSPARC-T1 and UltraSPARC-T2 platforms have significant differences that can impact your platform decisions with LDoms (see Table 1). The two differentiating factors are CPU features and I/O capabilities.

Feature	UltraSPARC-T1	UltraSPARC-T2
UltraSPARC-T2	8 maximum	8 maximum
Threads per core	4	8
Floating-point units	1 shared with each core	1 per core
Cryptographic units	1 (MAU) per core	1 (MAU/CWQ) per core
Memory controller	4 @ 23 GB/s	4 @ 50 GB/s
PCI-E controller	External ASIC connected to processor on the JBUS	Embedded into chip; 8 lanes @ 2.5 GHz with 2 GB/s bandwidth in each direction
PCI-E slots	Depends on model: ranges from 1 to 3	Depends on model: ranges from 3 to 6
PCI-X slots	T2000 has 2 PCI-X slots	None
Networking	Two external dual port 1 Gb Ethernet ASICs or through option cards	Embedded dual 10 Gb Ethernet on chip; two external dual-port 1 Gb Ethernet ASICs or through option cards
Storage	External SAS controller or through option cards	External SAS controller or through option cards

TABLE 1: COMPARISON OF ULTRASPARC-T1 AND ULTRASPARC-T2 PLATFORMS

These features play an integral role for designing solutions around LDomS. Table 2 outlines the key factors you should consider.

Feature	Consideration
Core	It is recommended that the primary domain have at least one core. So the number of cores available for guest domains is $N - 1$.
Threads per core	This determines the maximum number of VCPUs available for LDomS.
Floating-point units	The UltraSPARC-T1 platform floating-point performance suffered because only one FPU was available. This can negatively impact performance of heavy FP applications. Luckily, the UltraSPARC-T2 does not suffer from this limitation, as each core has a dedicated FPU.
Cryptographic (MAU) units	Only one LDom can own an MAU in any given core.
Memory controller	The memory bandwidth can impact the performance of applications that make heavy use of memory resources.
PCI-E controller	The UltraSPARC-T2 platform has a PCI-E controller embedded into the chip. This controller communicates with different PCI-E switches to which the PCI-E slots are connected. This enables higher I/O bandwidth to the PCI-E components.
PCI-E slots	The number of PCI-E slots determines the number of option cards that can be installed and utilized by LDomS.
Networking	Only the UltraSPARC-T2 has a dual 10 Gb Ethernet controller embedded into the chip. This increases throughput for high-performance network requirements, NAS, iSCSI, and streaming data.
Storage	The number of disks available internally can directly affect the number of guest domains that can be created when JBOD, SAN, NAS, or iSCSI storage is not available.

TABLE 2: FACTORS IN DESIGNING SOLUTIONS AROUND LDOMS

CPU Affinity

The UltraSPARC-T1 and UltraSPARC-T2 processors have a shared L2 cache that is utilized by all of the cores. Although you could technically divide a T5120 into 64 LDomS, that may not be practical for real workloads. For example, if you were to take a single core and allocate each VCPU out to separate guest domains with vastly different workloads, the probability of cache misses would increase. This causes the cache to work harder to feed each VCPU the required data. This can negatively impact the performance of your guest domains. To avoid this scenario, the following recommendations should be considered:

- Use whole cores for guest domains where possible for performance.
- Only allocate partial cores to guest domains that will host low-impact applications and services.
- Bind and start your larger guest domains first. This helps to ensure that your larger guest domains utilize a full core where possible. For

example, if you have a guest domain with eight VCPUs and another guest domain with two VCPUs, it makes sense to bind and start the larger guest domain first.

Split PCI-E Configurations

The PCI-E configuration on a platform can affect the ability to create a second I/O domain. As you recall, for an LDom to be an I/O domain, it must own part of the PCI-E bus. This is accomplished by assigning a PCI-E leaf to a guest domain as described in the vendor documentation [3]. You can then turn your I/O domain into a service domain by virtualizing networking and storage devices for your guest domains. However, not all platforms have multiple PCI-E buses available to be split among multiple service domains. The UltraSPARC-T1–based T2000 platform is one such machine that does have this support, but there are some limitations when using it:

- The T2000 only has a single SAS controller for the internal disk. By splitting the PCI-E configuration, only one of your domains can use the internal storage. As such, you'll need JBOD or SAN storage for your second I/O or service domain. Care must be taken to prevent the PCI-E leaf with the primary domain disks from being removed from the primary domain itself.
- One of your domains will have a single PCI-E slot and two 1 Gb Ethernet ports. The other domain will have two PCI-E slots, two PCI-X slots, and two 1 Gb Ethernet ports.

With the UltraSPARC-T2 platform, this capability is removed, as the PCI-E controller is embedded into the processor. All of the PCI-E components are connected via PCI-E switches. The only I/O component that can be split off to another domain is the NIU or 10 Gb Ethernet controller, which is also embedded into the processor. By doing so, you can have a guest domain capable of high-performance network throughput and communications.

In the future, the networking and storage controllers will be virtualized, with greater control for guest domains. The OpenSolaris NPIV project will enable a guest domain to have its own virtualized Fibre Channel HBA [4]. Also, the Hybrid I/O feature will enable a PCI-E leaf device to be assigned directly to a guest domain [5]. These features will become available in the future and provide greater flexibility for LDomS.

Dynamic and Delayed Reconfiguration

LDoms are capable of having virtual devices added and removed. Some of these virtual devices can be added or removed dynamically, which means that the LDom does not require a reboot for the change to take effect, whereas other virtual devices can only be added or removed when an LDom reboots. These differences are known as dynamic and delayed reconfiguration.

Currently, only VCPUs can be dynamically reconfigured with LDomS; all other virtual devices are relegated to delayed reconfiguration. As the technology evolves, more virtual devices will be capable of dynamic reconfiguration.

In this example, four VCPUs will be dynamically added to a guest domain:

```
ldom1:~ # psrinfo -vp
The physical processor has 4 virtual processors (0-3)
UltraSPARC-T2 (clock 1417 MHz)
```

```
primary:~ # ldm list
NAME      STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary   active -n-cv  SP    8     8G     0.3%  8h 46m
ldom1     active -n---  5000  4     2G     48%   5h 52m
```

```
primary:~ # ldm add-vcpu 4 ldom1
```

```
primary:~ # ldm list
NAME      STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
primary   active -n-cv  SP    8     8G     0.3%  8h 46m
ldom1     active -n---  5000  8     2G     48%   5h 52m
```

```
ldom1:~ # psrinfo -vp
```

```
The physical processor has 8 virtual processors (0-7)
UltraSPARC-T2 (clock 1417 MHz)
```

The VCPUs were added dynamically to the guest domain ldom1 without it having to be rebooted. This means that VCPU resources can be dynamically moved around depending on resource requirements. This can be useful for moving VCPU resources to where they are needed for application workloads.

Delayed reconfiguration requires the LDom to be rebooted. Multiple reconfiguration changes can be requested for the same LDom before it reboots, as they will be queued. Once a delayed reconfiguration operation for an LDom has been queued, reconfiguration requests for other LDomS are disabled until the queued requests are handled.

In this example, our guest domain will have memory and storage added:

```
ldom1:~ # prtdiag -v | grep Mem
Memory size: 2048 Megabytes
```

```
ldom1:~ # format
Searching for disks...done
```

```
AVAILABLE DISK SELECTIONS:
  0. c0d0 <SUN-DiskImage-10GB cyl 34950 alt 2 hd 1 sec 600>
     /virtual-devices@100/channel-devices@200/disk@0
  1. c0d1 <SUN-DiskImage-10GB cyl 34950 alt 2 hd 1 sec 600>
     /virtual-devices@100/channel-devices@200/disk@1
Specify disk (enter its number): ^D
```

```
primary:~ # ldm add-mem 2g ldom1
```

```
Initiating delayed reconfigure operation on LDom ldom1. All configuration
changes for other LDomS are disabled until the LDom reboots, at which time
the new configuration for LDom ldom1 will also take effect.
```

```
primary:~ # mkfile 5g /ldoms/local/ldom1/ldom1-vdisk2.img
```

```
primary:~ # ldm add-vdsdev /ldoms/local/ldom1/ldom1-vdisk2.img ldom1-vdisk2@primary-vds0
```

```
primary:~ # ldm add-vdisk ldom1-vdisk2 ldom1-vdisk2@primary-vds0 ldom1
```

```
-----
Notice: LDom ldom1 is in the process of a delayed reconfiguration.
Any changes made to this LDom will only take effect after it reboots.
-----
```

```
ldom1:~ # reboot
```

```
...
```

```
ldom1:~ # prtdiag -v | grep Mem
Memory size: 4096 Megabytes
```

```
ldom1:~ # format
Searching for disks...done
```

AVAILABLE DISK SELECTIONS:

0. c0d0 <SUN-DiskImage-10GB cyl 34950 alt 2 hd 1 sec 600>
/virtual-devices@100/channel-devices@200/disk@0
1. c0d1 <SUN-DiskImage-10GB cyl 34950 alt 2 hd 1 sec 600>
/virtual-devices@100/channel-devices@200/disk@1
2. c0d2 <SUN-DiskImage-5GB cyl 17474 alt 2 hd 1 sec 600>
/virtual-devices@100/channel-devices@200/disk@2

Specify disk (enter its number): ^D

Delayed reconfiguration requests can be canceled for an LDom. However, doing so will remove any queued items as well.

```
primary:~ # ldm rm-mem 2g ldom1
```

Initiating delayed reconfigure operation on LDom ldom1. All configuration changes for other LDomS are disabled until the LDom reboots, at which time the new configuration for LDom ldom1 will also take effect.

Notice: this remove operation will prevent any future VIO device removal operation from being accepted for the duration of this delayed reconfiguration, i.e. until the domain reboots or the delayed reconfig is cancelled.

```
primary:~ # ldm remove-reconf ldom1
```

Notice that this operation of removing memory prevented any further removal operations from queueing. The LDM software will alert you of such conditions.

There are a few caveats about dynamic and delayed reconfiguration that should be kept in mind:

- Be mindful of removing VCPUs from an LDom that has an MAU bound in the same core. The MAU may have to be removed first through delayed reconfiguration if the VCPUs being removed are the only ones assigned to the LDom from that core.
- For better cache coherency, an LDom should scale within a single core until additional VCPUs are required from another core.
- The ldm command will warn you if requests can be handled or if they must be held off until a reconfiguration operation has completed.

Configuration Management

The configuration for your LDomS should be backed up regularly. Each LDom configuration can be dumped into an XML configuration file. The configuration dump only contains the mapping of the resources and virtual devices that are configured for the LDom. However, this does not include the configuration of the underlying device services such as the VDSDEVs or the VSWs. This configuration file can be used to recreate or duplicate LDom configurations:

```
primary:~ # ldm list-constraints -x ldom1 > ldom1.xml
```

This configuration file can be used to recreate the LDom in a recovery scenario or when migrating a guest domain from one server to another. For example, if the above LDom were removed accidentally, the configuration could be restored:

```
primary:~ # ldm list ldom1
```

LDom "ldom1" was not found

```
primary:~ # ldm add-domain -i ldom1.xml ldom1
```

```
primary:~ # ldm list ldom1
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
ldom1	inactive	-----		4	4G		

```
primary:~ # ldm bind ldom1
primary:~ # ldm start ldom1
LDom ldom1 started
```

```
primary:~ # ldm list ldom1
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
ldom1	active	-t---	5000	4	4G	30%	0s

```
ldom1:~ # uname -a
SunOS ldom1 5.11 snv_77 sun4v sparc SUNW,SPARC-Enterprise-T5120
```

This process cannot be used to restore the primary domain configuration. However, the XML dump can provide valuable information in the event that it must be recreated manually.

High Availability

Clustering today with LDoms is in its infancy. Many of the clustering products, such as Solaris Cluster and Veritas Cluster Server, are just beginning to support installation into control and I/O domains. However, they lack agents to properly support guest domains and the applications contained within them. This will change as the products mature to support LDoms. However, in the meantime you can create a standby environment for your guest domains in the event of a failure. For this you will need the following:

- Two or more servers that are configured similarly
- SAN or NAS storage for your guest domains

Here is an example of creating such a standby environment utilizing NAS storage:

```
primary:~ # mkfile 10g /ldoms/nas/ldom4-vdisk0.img
primary:~ # df -h /ldoms/nas
```

Filesystem	size	used	avail	capacity	Mounted on
192.168.2.70:/export/ldoms	107G	10G	97G	1%	/ldoms/nas

```
primary2:~ # df -h /ldoms/nas
```

Filesystem	size	used	avail	capacity	Mounted on
192.168.2.70:/export/ldoms	107G	10G	97G	1%	/ldoms/nas

At this point, we can create ldom4 on our first server:

```
primary:~ # ldm add-domain ldom4
primary:~ # ldm add-vcpu 4 ldom4
primary:~ # ldm add-mem 4G ldom4
primary:~ # ldm add-vnet vnet0 primary-vsw0 ldom4
primary:~ # ldm set-variable auto-boot\?=false
primary:~ # ldm add-vdsdev /ldoms/nas/ldom4-vdisk0.img ldom4-vdisk0@primary-vds0
primary:~ # ldm add-vdisk ldom4-vdisk0 ldom4-vdisk0@primary-vds0 ldom4
primary:~ # ldm bind ldom4
primary:~ # ldm start ldom4
LDom ldom4 started
```

```
primary:~ # ldm list ldom4
```

NAME	STATE	FLAGS	CONS	VCPU	MEMORY	UTIL	UPTIME
ldom4	active	-n---	5004	4	4G	0.0%	34

Now we can dump the XML configuration of our guest domain ldom4:

```
primary:~ # ldm list-constraints -x ldom4 > /ldoms/nas/ldom4.xml
```

The configuration can be imported on our second server, once the VDS device has been configured:

```
primary2:~ # ldm add-vdsdev /ldoms/nas/ldom4-vdsk0.img ldom4-vdsk0@primary2-vds0
primary2:~ # ldm add-domain -i /ldoms/nas/ldom4.xml
```

```
primary2:~ # ldm list ldom4
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldom4         inactive  ----          4     4G
```

Once we have installed the OS into the guest domain on our first server, we can test our configuration:

```
ldom4:~ # uname -a
SunOS ldom4 5.11 snv_77 sun4v sparcsunw,SPARC-Enterprise-T5120
ldom4:~ # shutdown -y -g0 -i 5
...
```

```
primary:~ # ldm stop ldom4
LDom ldom4 stopped
primary:~ # ldm unbind ldom4
```

```
primary:~ # ldm list ldom4
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldom4         inactive  ----          4     4G
```

```
primary2:~ # ldm bind ldom4
primary2:~ # ldm start ldom4
LDom ldom4 started
```

```
primary2:~ # ldm list ldom4
NAME          STATE  FLAGS  CONS  VCPU  MEMORY  UTIL  UPTIME
ldom4         active  -n-cv  5004   4     4G     0.3%  5m
```

```
ldom4:~ # uname -a
SunOS ldom4 5.11 snv_77 sun4v sparcsunw,SPARC-Enterprise-T5120
```

One could script this process to migrate guest domains between servers once the configuration is in place on each server. In the future, LDoms will also support the ability to migrate guest domains between servers without any downtime. This feature is called “Live Migration” and will be similar to the VMWare Vmotion feature [5].

Running Multiple Operating Systems

One of the strengths of LDoms is the ability to run multiple guest domains with different operating systems at the same time. You can install the following operating systems into a guest domain:

- Solaris 10 Update 3 (11/06) and above
- Solaris Express Community Edition, build 70 and above
- Solaris Express Developer Edition 09/07 and above
- OpenSolaris, build 70 and above
- Ubuntu Linux 7.10 and above

There are other operating systems that already have sun4v platform support or are developing support. The key to working with LDoms is to have support for the virtualized devices, such as the VNETs and VDISKS. Once the proper support is added to an OS, it can be used in a guest domain. Here

is a demonstration of different OSes running on a single physical server by using LDomS:

```
ldom1:~ $ uname -a
SunOS ldom1 5.11 snv_77 sun4v sparc SUNW,SPARC-Enterprise-T5120

ldom2:~ $ uname -a
SunOS ldom2 5.11 snv_75 sun4v sparc SUNW,SPARC-Enterprise-T5120

root@ldom3:~ $ uname -a
Linux ldom3 2.6.22-14-sparc64-smp #1 SMP Tue Dec 18 05:40:10 UTC 2007 sparc64 GNU/Linux
root@ldom3:~# cat /etc/lsb-release
DISTRIB_ID=Ubuntu
DISTRIB_RELEASE=7.10
DISTRIB_CODENAME=gutsy
DISTRIB_DESCRIPTION="Ubuntu 7.10"

ldom4:~ # uname -a
SunOS ldom4 5.11 snv_77 sun4v sparc SUNW,SPARC-Enterprise-T5120

ldom5:~ $ uname -a
SunOS ldom5 5.10 Generic_120011-14 sun4v sparc SUNW,SPARC-Enterprise-T5120
```

As you can see, there are three guest domains running Solaris Express at different releases, one guest domain running Ubuntu Linux 7.10, and a final guest domain running Solaris 10 Update 4 (08/07).

This can be very beneficial for applications testing or development projects that require different OS versions, patch levels, or configurations. It can also be an efficient method for testing new products before migrating to them on the same hardware. The cost savings can be significant in both time and equipment.

Comparisons

There are many virtualization technologies today that can be utilized across a wide range of platforms and operating systems. As the demand for server utilization efficiencies increases, the requirement to leverage virtualization will become common practice in data centers. All of these technologies can be broken into three major categories, as outlined in Table 3.

Technology	Description
Hardware partitions	Hardware partitions are created from specialized ASICs and firmware that enable components in a platform to be grouped into smaller systems and electrically isolate them from failure. This provides the highest level of separation between multiple OS instances. This is seen on Sun equipment such as the E25k or the new SPARC Enterprise M9000.
Virtual machines	Virtual machines are created through software that is either in firmware or in a management OS instance. This software is able to virtualize or emulate the hardware into groupings capable of running isolated instances of an operating system. This provides many pros and cons depending on the requirements. Virtual machines are commonly seen in technologies such as VMware, Xen, Sun xVM, IBM LPARs, Parallels, and QEMU.

TABLE 3: VIRTUALIZATION CATEGORIES (continued on p. 32)

OS virtualization	OS virtualization occurs when a single OS instance is able to create an isolated run-time environment that closely emulates a standalone OS installation. When combined with resource management, this can effectively utilize hardware resources, because the overhead is very low. This is seen in technologies such as Solaris Containers (Zones), BSD Jails, IBM WPARs, OpenVZ, and Linux-VServer.
-------------------	--

TABLE 3: VIRTUALIZATION CATEGORIES (continued from p. 31)

Logical domains are a hybrid of hardware partitioning and virtual machines. The control domain uses the hypervisor to partition CPU and memory resources into groupings for guest domains, whereas the service domain virtualizes the I/O components, such as networking and storage for guest domains. This interesting combination provides many benefits:

- High level of integration with the hardware via the sun4v hypervisor.
- The ability to leverage built-in hardware features of both the UltraSPARC-T1 and the UltraSPARC-T2 processors, such as CMT, cryptographic engines, and 10 Gb Ethernet
- Reduced overhead for CPU and memory resources
- Flexibility in virtualizing I/O components
- The ability to leverage Solaris features such as ZFS and iSCSI for guest domains
- The ability to create Solaris Containers within LDOMs, increasing the level of virtualization

Summary

This article has introduced you to advanced topics concerning the configuration and management of logical domains. With this knowledge, you should be able to explore this technology in greater detail and discover interesting ways in which it can be applied. This technology will continue to evolve and mature. As it becomes open sourced, you will be able to help with the development and advancement of this technology.

WHERE TO FIND MORE INFO

OpenSolaris LDOMs Community: <http://www.opensolaris.org/os/community/ldoms>.

OpenSolaris LDOMs Community discussion: <http://www.opensolaris.org/jive/forum.jspa?forumID=203>.

Sun LDOMs home page: <http://www.sun.com/servers/coolthreads/ldoms/index.xml>.

Installing Ubuntu Linux on SPARC: <https://help.ubuntu.com/community/Installation/Sparc>.

My blog: <http://unixconsole.blogspot.com/>.

REFERENCES

- [1] Sun LDoms home page: <http://www.sun.com/servers/coolthreads/ldoms/index.xml>.
- [2] OpenSolaris LDoms discussion on future features: <http://www.opensolaris.org/jive/thread.jspa?messageID=148688𤓐>.
- [3] *LDM 1.0.1 Administrative Guide*: <http://docs-pdf.sun.com/820-3268-10/820-3268-10.pdf>.
- [4] OpenSolaris NPIV Project: <http://opensolaris.org/os/project/npiv>.
- [5] LDoms presentation by Liam Merwick: <http://opensolaris.org/os/community/ldoms/files/LDoms-LOSUG-Oct-2007.pdf>.

MATTHEW SACKS

Linux kernel resource allocation in virtualized environments



Matthew Sacks works as a systems administrator at Edmunds.com. His focus is network, systems, and application performance tuning.

matthew@matthewsacks.com

NOTE: VMware's ESX platform is certainly not the be-all and end-all of virtualization. However, it is widely used and accepted. The same principles used in this example may apply to other virtual platforms. The tuning methods provided in this article are not intended as a replacement for good capacity planning.

THE BEHAVIOR OF THE LINUX KERNEL

and its resource allocation methods are an art and science, more the former than the latter. When working with Linux in a virtualized environment, the complexities of the Linux kernel's resource allocation algorithms increase. New performance issues may arise and proper functionality can come to a halt—especially on overutilized systems. The Linux kernel behaves differently on a virtualized platform in comparison to bare-metal. Why Linux behaves differently on a virtual platform and how to address performance and stability issues when it starts to malfunction or degrade in performance relate, but do not depend on, the environment. The solution presented here is not that of changing the environment, but, rather, that of making adjustments to the Linux kernel to coalesce with the hypervisor.

The Environment

The virtual environment comprised 6 VMware ESX 3.01 servers running approximately 11 virtual machines per server. Each ESX server had Red Hat Enterprise Linux 4 Update 4 machines running a wide array of application and Web servers. The environment was intended to simulate a high-volume, high-traffic production Web site by simulating load tests on the virtual servers. The phenomenon experienced was the Linux Kernel's OOM-Kill function, which would trigger and kill processes that were consuming the most resources. How the ESX server interacts with this Linux kernel in allocating resources is the starting point.

VMWARE ESX SERVER RESOURCE ALLOCATION

The VMware ESX server adds another layer of abstraction between the Linux server's physical and virtual memory and the real memory of the ESX server. The ESX server creates additional memory overhead in managing the virtual devices, CPUs, and memory of the virtual machine. It can be thought of as virtual memory that manages virtual memory: a new set of resources that must be managed on top of the guest operating system's own virtual memory management algorithms. Resources

can run thin quickly and resource allocation issues tend to increase faster on a virtual server than on a bare-metal server.

For example, consider an ESX server with 1 GB of memory, running two virtual machines with 256 MB of “physical” memory allocated for each virtual server. The amount of free resources available is approximately 170 MB. The service console uses approximately 272 MB, the VMkernel uses somewhat less than that, and, depending on how many virtual CPUs and devices are added to each virtual server, the memory overhead increases.

ESX uses a proprietary memory ballooning algorithm to adjust and allocate memory to virtual servers. ESX loads a driver into the virtual server which modifies the virtual server’s page-claiming features. It increases or decreases memory pressure depending on the available physical resources of the ESX server, causing the guest to invoke its own memory management algorithms. When memory is low the virtual server’s OS decides which pages to reclaim and may swap them to its virtual swap.

However, sometimes pages cannot be reclaimed fast enough or memory usage grows faster than can be committed to swap; then the Linux OS kills processes and “Out of Memory” errors appear in the syslog.

The Linux Out of Memory Killer

The `out_of_memory()` function is invoked by `alloc_pages()` when free memory is very low and the Page Frame Resource Allocation Algorithm has not succeeded in reclaiming any page frames. The function invokes `select_bad_process()` to select a victim and then invokes the `oom_kill_process()` to kill the process that is utilizing the most resources. Typically, the `select_bad_process()` function chooses the process that is not a critical system process and is consuming the largest number of page frames. This is why, when running a resource-intensive application or Web server on a virtual environment, the application or Web server may begin crashing frequently. Check the logs for the “Out of Memory” errors to see if the `oom_kill_process()` function is being called by the Linux kernel. The `oom_kill_process()` function comes into play because of how Linux is allocating memory into lower memory zones.

Memory Zones and the Dirty Ratio

By default, the Linux kernel allows addressing of memory in the lower zone called `ZONE_DMA`. This zone contains page frames of memory below 16 MB. In high workloads, once the `ZONE_NORMAL` (normal memory zone) and `ZONE_HIGHMEM` have been exhausted by an application, it will begin to allocate memory from `ZONE_DMA` and the requestor of the application will pin them, thereby denying access to these zones by other critical system processes. The `lower_zone_protection` kernel parameter determines how aggressive the kernel is in defending the lower memory allocation zone.

The dirty ratio is a value expressed in percentage of system memory at which limit processes generating dirty buffers will write data to disk rather than relying on the `pdflush` daemons to perform this function. The `pdflush` kernel thread scans the page cache looking for dirty pages (pages that the kernel has set to be swapped to disk) and then ensures that no page remains dirty for too long. `ZONE_DMA` can be protected from being utilized by applications and the dirty ratio can be adjusted by tuning the Linux kernel.

Tuning the Linux Kernel for Virtualization

The `/etc/sysctl.conf` file allows modification of select kernel settings without recompilation of the kernel. The `/etc/sysctl.conf` file is used to adjust behaviors of the Linux kernel to address issues with resource allocation. A set of virtual memory tunable parameters is available for tuning from within this file. Two tunable virtual memory parameters in particular will solve the “Out of Memory” problems and most other problems with memory allocation in a virtual Linux server.

To protect the lower zones of memory from being utilized by the applications on a virtual Linux server, edit the `/etc/sysctl.conf` file to include the following parameter:

```
lower_zone_protection = 100
```

To increase the ratio by which the `pdflush` kernel thread scans the page cache to look for dirty pages to 5 percent of the system memory, edit `/etc/sysctl.conf` to include the following parameter:

```
dirty_ratio = 5
```

Reboot the system or type the command `sysctl -p` so that the new kernel settings will take effect. Now most memory resource allocation issues should be resolved in a virtualized Linux environment. Tuning these few settings provides a small insight into how tuning the Linux kernel can solve performance-related problems in a virtualized environment. As a result of the tuning changes, “Out of Memory” errors are reduced dramatically in scope and frequency, and virtual memory is utilized more effectively.

Conclusion

There are numerous algorithms at work with VMware’s ESX server and within the Linux kernel itself. In a low-volume environment the standard configurations and settings may be sufficient. In a high-volume, high-performance environment where load tests are constantly making requests against application and Web servers, the defaults are typically insufficient. To squeeze the maximum amount of performance out of a system, an understanding of the underlying algorithms and behaviors of the ESX server is essential before tuning the guest operating system’s kernel. The end result is maximal performance on an otherwise overutilized or poorly performing virtual environment. The key is to understand which algorithms need to be changed and to set them to the right values. This is where kernel tuning becomes more of an art than a science.

ACKNOWLEDGMENTS

I want to acknowledge the Systems Administration Team at Edmunds.com, Safdar Husain, David Wolfe, David Morgan, Nikki Sonpar, and Eric Theis, who all contributed in some way to this article.

REFERENCES

- [1] D. Bovet and M. Cesati, *Understanding the Linux Kernel* (Sebastopol, CA: O'Reilly & Associates, 2005).
- [2] B. Matthews and N. Murray, "Virtual Memory Behavior in Red Hat Linux A.S. 2.1," Red Hat white paper, Raleigh, NC, 2001.
- [3] N. Horman, "Understanding Virtual Memory in Red Hat Enterprise Linux 4," Red Hat white paper, Raleigh, NC, 2005.

ADITYA K SOOD

hacking 802.11 protocol insecurities



Aditya K Sood, a.k.a. oknock, is an independent security researcher and founder of SecNiche Security, a security research arena. He is a regular speaker at conferences such as XCON, OWASP, and CERT-IN. His other projects include Mlabs, CERA, and TrioSec.

adi.zerok@gmail.com

SECURITY AND PRIVACY ARE TWO CRITICAL entities in any communication protocol. Security itself is a prerequisite for robust implementation of networks. In this article, I dissect the 802.11 [1] protocol attacks possible because of persistent problems in wireless networks. Before going into the attack patterns against the protocol, I will briefly describe how 802.11 works by splitting frames into functional objects.

The protocol is constructed to work between access points and stations. Every second (unless disabled), the access point transmits a signal in the form of wireless messages called beacons. The station listens for beacons on different frequencies called channels. Stations can also use probe request messages to scan a certain network for finding an access point. This probing and beaconing initiates the association between a station and an access point. An association message is used for initial connection by using a request/response mechanism. Similarly, a dissociation method is applied for connection termination. The frames-based IEEE 802.11 Frame Format is used for sending data (Figure 1). Three types of addresses are used for sending data. The Service Set Identifier (SSID) [1] is defined for networks to uniquely identify various access points. The identification process is completed by sending a Preamble as a first element of the frame. The PLCP header holds information regarding receiver logic (data rate, etc.). The MAC header is used for address specification. The user data is checksummed (CRC) for transmission and reception errors.

Preamble	PLCP	MAC	User Data	CRC - Cyclic Redundancy Check
----------	------	-----	-----------	-------------------------------------

FIGURE 1: IEEE 802.11 STANDARD FRAME FORMAT

The access points can communicate wirelessly with other access points by using a process called wireless bridging. The Media Access Control uses four different types of addresses to complete the protocol communication. Transmitter Address (TA), Receiver Address (RA), Source Address (SA), and Destination Address (DA) comprise the 802.11 communication address pattern. The MAC frames are dissected into three main categories: control,

data, and management. The working functionality of the protocol revolves around this. Insecurities define the domain over which an attack occurs. The size of the attack surface increases with the number of insecurities in the 802.11 protocol. Attacks can be split into a logical hierarchy, shown in Figure 2.

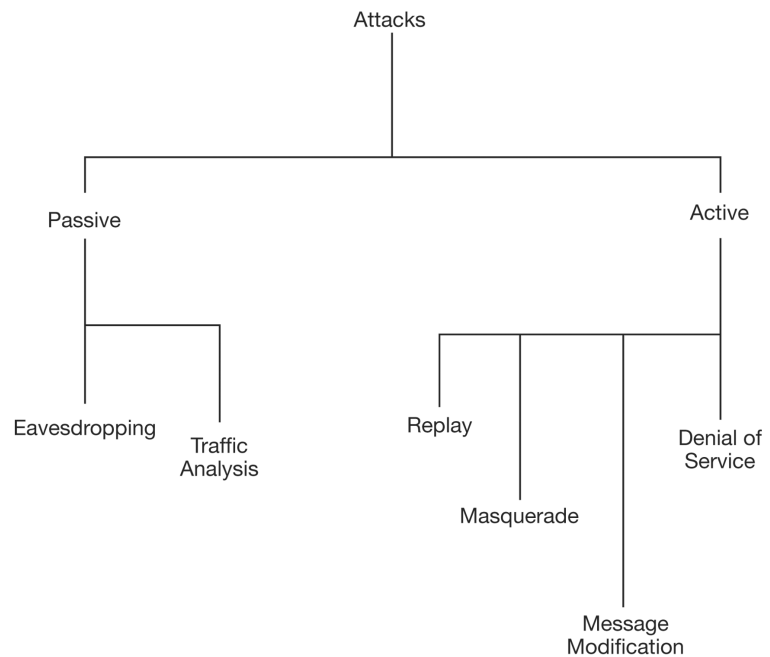


FIGURE 2: HIERARCHY OF ATTACK TYPES

Protocol Insecurities

MAC DISCLOSURE

One of the most insecure vectors in 802.11 is the public display of the MAC [2] address, which is a prime cause of spoofing attacks and traffic manipulation. 802.11 defines MAC operations in contention-free and contention-based modes. The term *contention* here means the procedure the station uses to communicate with an access point or media. Hijacking attacks take over a connection by masquerading the MAC address of a station. MAC relates to security context directly. ARP poisoning attacks are possible through man-in-the-middle techniques. These attacks are based on sniffing the network traffic. An attacker can easily change the MAC address of the devices under control. In this way an attacker performs the man-in-the-middle attack. On a shared network, it is possible to coexist with different hosts while having the same IP and MAC address, a state called piggybacking. The attacker must be very cautious in sending the packets in the network, because too many reset packets or ICMP unreachable messages can cause problems in the wireless network, resulting in network instability. A WIDS (Wireless Intrusion Detection System) catches the culprit host in the network when an attacker tries to kick the victim host out of the network. To overcome this

problem attackers try to find a host that is active in the network but does not generate traffic. This results in virtual control of a host, because the attackers change their identity by transforming the identity addresses, thereby sending deassociate frames to the victim host. This process is considered to be silent control of the host. The network is flooded with deassociate frames that are continuously sent to the victim host by the attacker by spoofing the MAC address of the access point. In Linux the MAC address can be changed easily during boot time or with an efficient utility called sea [3]. It directly configures the adapter with the type of MAC address specified by the attacker. In a Windows environment the MAC address can be altered easily by changing the registry settings.

WEP INSECURE VECTORS

The Wired Equivalence Privacy (WEP) [4] is a security-driven mechanism used for wireless network security. The authentication is based on a challenge–response mechanism (Figure 3). The basic problem is that using the same keys for encryption and authentication breaks the rule of independent keys. Authentication covers the simple encryption and decryption check of a random number string. Another problem is preserving identity, as no tokens are used for transactions. The double XOR operation on a pseudo-random string with plain text enables an attacker to bypass the authentication mechanism easily without even knowing the secret key. This ambiguity marginalizes the security of networks substantially. Specifically, no standard method is defined for access control—it is entirely based on MAC list generation in which allowed targets are specified. Failure of identification by MAC or WEP key causes direct access failure and no connection. Another problem associated with WEP is that no particular method is provided to combat against replay attacks. The MAC address of the victim can be used to resend messages to an access point, which automatically decodes it since no subtle protection is provided to scrutinize replay requests.

Let’s look at the mechanism of shared key message authentication flow for a better understanding.

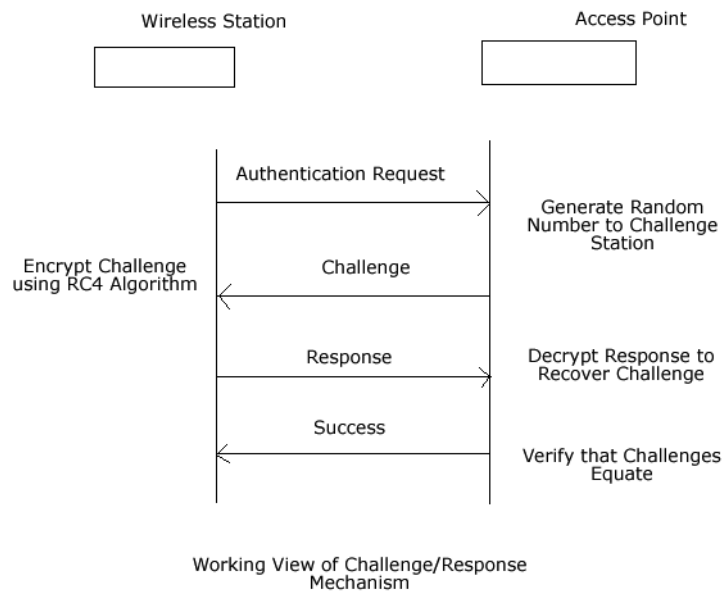


FIGURE 3: WEP CHALLENGE-RESPONSE AUTHENTICATION SEQUENCE

WEP uses a linear method to compute a cyclic redundancy check. An assumption has been made that if a message is computed with a CRC value and is encrypted, then data modification attacks can be circumvented. But this is totally false. Flipping a bit in the original message always shows the same flipping effect in the encrypted message. WEP is unable to prevent cipher text modification attacks. Message privacy can be bypassed easily through brute-forcing attacks on WEP keys or generating techniques to decode a message. As per standards it has been noticed that the 40-bit WEP key generation algorithm is vulnerable to a number of flaws, as a result of which brute-force attacks on 40-bit keys are easy to perform.

The attacks on WEP are classified as either passive or active. Passive attacks comprise attacks that are performed on the static log files, debug responses, etc. The FMS [5] technique is one of the finest key-recovery procedures. Attackers use this procedure effectively to crack keys in a static manner. Active attacks comprise injecting extra traffic in the network to crack keys within specific time limits. The injection of traffic accelerates the WEP-cracking process. Active attacks are possible despite less traffic. The injected traffic by the attacker not only enhances the cracking process but is also helpful in understanding protocol structure, which further results in host discovery and enumeration. It works on low-level protocol structure and analyzes the flags sent in TCP and ICMP protocols. So once an attacker understands the required pattern of traffic, the attacks become easy to perform.

IMPLICIT DENIAL OF SERVICE

Wireless networks are prone to different types of denial-of-service attacks.

Clear to Send (CTS) and Request to Send (RTS) are control type frames. The RTS operation comes into play whenever a big packet is to be sent with continuous transmission. To avoid collision the station sends an RTS packet to an access point for reserving a channel for some time. If the access point agrees, a CTS packet is sent to the station in return. The client is unable to use the CTS packet because of the hidden-node problem. Attackers exploit CTS packets by continuously injecting them in the network to produce a denial-of-service attack. This reduces the robustness of the network, thereby resulting in service degradation.

The second factor involves communication failure between two hosts that are communicating on a connection-oriented basis; if a link fails in the connection-oriented protocol there should be retransmission of packets. This process is continuous until the whole datagram is passed to the destination. As a result of this the number of optimum packets is increased and so are the frames used to capture it. On the other side, if the frame size is decreased to reduce the incoming packet data to be sent, the problem persists because this enhances the fragmentation process in the network. Either way, a small mismatch in the network can cause large problems in the network.

The third factor that can lead to a denial-of-service attack is link disruption, which generates an excessive amount of traffic, which in turn generates routing updates. This type of problem persists in wireless networks when routers go down. If a router stops working, then a flood occurs that generates new data for the link state protocol. This means that the algorithm used in routing updates triggers with new data routes. As a result, load rises and network time is spent in overcoming this problem. If this process becomes periodic, then the routes are affected continuously, marking those specific routes as flapped. Distance vector protocols such as RIP/IGRP generate traffic regularly, but because of link failure produce a flood of regular updates.

An attacker can easily exploit any of these three factors to disrupt the functioning of a wireless network. None of the solutions to combat these factors is very reliable, because the root cause of these problems is protocol malfunctioning, which in itself entails technology manipulation.

The 802.11 insecurities are enumerated as follows:

- Tempering VPN tunnels: Virtual private networks are implemented with PPTP [6] and IPSEC. Attackers can easily attack PPTP to leverage a lot of information directly from the traffic flow. The technique is based on the concept of a falsified parameter. The attacker sniffs the traffic and tries to understand the packet layout used in communication. Basically, the attacker wants to control the authentication mechanism between the VPN server and the client. As we already know, PPTP implements MSCHAP [7] and MSCHAP-2 [6], the Microsoft Challenge Handshake Protocol for password authentication and password change protocol. Software has been designed by attackers to control the authentication credentials by a fake process. Attack software actively monitors the traffic and detects when a client tries to log onto a server using PPTP. The software activates a false dialog and tricks the user into providing credentials (a man-in-the-middle attack). An amateur user simply provides the credentials, which in turn are replayed by the attacker on the server.
- Once the MSCHAP hashes are sniffed, they can be cracked to produce a clear text password. Tools such as Ettercap with plug-ins can perform this task in an efficient manner.
- IPSEC attacks: Another possible attack target is IPsec. The attacker scans the whole wireless network against the IPsec implementation. With the help of denial-of-service attacks, the culprit can force the network administrator to shut down the IPsec implementation for some time. Actually, the IPsec concept is based on Internet Key Exchange (IKE), in which IKE scanners find the vulnerable host and compromise it by successfully running exploit code.
- Rogue access points: Rogue access points are used to attack wireless networks that use the EAP-MD5 authentication mechanism. For this an attacker requires a fake RADIUS [8]. RADIUS will provide fake authentication credentials to the client host. This is also considered a man-in-the-middle attack. A single machine can easily provide a base for both access point and RADIUS. Because of this stringent problem most administrators have started using the EAP-MD5 solution as a fallback only. The attack becomes more subtle when the attack starts jamming the real access point signals and injecting its own access point signals to a network a number of channels away. This gives the attacker hidden control over the network. Such jamming is possible by junk traffic being sent to the network with the help of tools that manipulate layer 1 functionality of the OSI model. Parameters used for the rogue access point should be similar to real ones, to avoid conflicts in the network. The layer 2 attacks are performed by sending deassociation and deauthentication packets to the victim to kick it out of the network. An attacker performs layer 1 and layer 2 attacks frequently and in a defined manner to exploit the functionality of rogue access points. This problem is inherited in wireless networks because of its open access point methodology.
- WPA insecurity context-cracking: Wi-Fi Protected Access (WPA) is a subset of the Robust Security Network (RSN) [9]. It defines the protected access mechanism in the form of the encryption protocol that is deployed in 802.11 wireless networks. Its running structure is differ-

entiated between home mode and enterprise mode. Home mode uses a Pre-Shared Key (PSK) and enterprise mode uses a RADIUS server for authenticating clients. A Pairwise Master Key (PMK) is computed from PSK and SSID. A hashing function is used for generating PMK. Precomputed hash attacks can easily be applied to crack the hashes. It works very effectively on WPA1 and WPA2 because both versions use four-way handshake mechanisms for association. The packets can be easily decrypted by hardware-based tools that accelerate the cracking process. The Extensible Authentication Protocol (EAP) [10] and Protected EAP (PEAP) [11] are very hard to exploit, because the working algorithm used is RSA. EAP is based on certificate exchange between server and client. The only method to compromise it is to steal the keys to control EAP on the network.

Overall Countermeasures

- Understand the organizational requirements. Normally, several layers of network protection are added (e.g., multiple authentication) to prevent attacks. How many layers depends a lot on the need of the organization and the physical structure of the network. If an organization plans on communicating financial transactions then it must be assured that a hacker will not be able to intercept the traffic and steal the credentials. If remote working is required then VPN solutions are advised. The network should be constructed in a simple manner, enabling the administrator to control and maintain the wireless network efficiently.
- Apply encryption in multiple layers. The main stress should be laid on the generation of WEP keys per user per session. This means users will encounter different WEP keys for every session they establish, thereby lowering the possible theft and reuse of the WEP keys because an attacker can benefit only when a user is active. Once the user closes the session the keys become useless. This technique is implemented with LEAP [4]. The number of packets encrypted with a LEAP-generated key is much lower than the number of packets required to break the algorithm. This type of encryption not only provides a secure mechanism but also an interoperable environment.
- Design VLANs as a backbone to wireless networks. In such a design, the access points are connected to the wired network physically or logically. This can be accomplished by setting a separate switched network, which is possible with VLANs. The administrator sets a VLAN device behind the firewalls. It enables the firewalls to filter the wireless traffic that is coming inside and leaving the network. Multiple layers of security can be added with extra features by enabling security devices.
- Alter the default setting of various network parameters and protocols to unique values. First, the default passwords should be changed. The SSID value must be changed to something different from the factory value. Second, change the cryptographic keys provided by the manufacturer for shared key authentication. Most wireless networks use SNMP agents. The default SNMP parameters should be changed. The default channels of access points should be set differentially to reduce conflict between two networks. The overall change in default parameters is advised to reflect specific organizational policies.
- Apply patches as soon as a vulnerability is released. This process should encompass every single item of hardware and software used in the network design.

- Apply security at the perimeter level. This includes the implementation of firewalls, WIDS, and other devices in switched networks. These devices provide physical layer security and work on defined policies. Actually, signatures and rules filter the traffic on the inherited benchmarks, thereby reducing the attack vector from the security point of view. These devices are considered as the default layer of security.
- Design and implement MAC access control lists to circumvent MAC attacks. These lists have predefined MAC addresses that are to be given access in the network by the administrators. The access lists use the grant and permit operation to perform in the wireless network. But the MAC address is distributed in a clear text so that it can be captured easily. For normal networks the MAC access control list can be implemented to reduce the intensity of attacks based on MAC.

These countermeasures can control wireless network attacks to some extent but cannot be considered as direct solutions for wireless security.

Conclusion

These issues with the IEEE 802.11 protocol lead to the hacking of networks. The various insecurities generate a large attack surface and defenses can be breached very easily. You can prevent attacks to some extent but you cannot eliminate them. The many countermeasures listed strengthen the security aspect up to a point but cannot make your network bulletproof. The basic problem resides in the presence of the complexity endemic to protocol requirements in wireless networks. Security is a process, not a one-shot activity. Implementing heavy security entails looking at the hidden artifacts in the network to dethrone concurrent attacks.

REFERENCES

- [1] E. Danielyan, "802.11," *The Internet Protocol Journal*, 5, 1 (March 2002).
- [2] "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," IEEE Standard 802.11, 1999 Edition.
- [3] <http://www.openbsd.org>.
- [4] N. Borisov, I. Goldberg, and D. Wagner, "Security of the WEP Algorithm," <http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html>.
- [5] http://www.cs.umd.edu/~waa/class-pubs/rc4_ksaproc.ps.
- [6] <http://www.counterpane.com/pptpv2-paper.html>.
- [7] <http://packetstormsecurity.org/groups/teso/chap.pdf>.
- [8] C. Rigney, S. Willens, A. Rubens, and W. Simpson, "Remote Authentication Dial-In User Service," IETF RFC 2865, June 2000.
- [9] http://en.wikipedia.org/wiki/Robust_Security_Network.
- [10] http://en.wikipedia.org/wiki/Extensible_Authentication_Protocol.
- [11] http://en.wikipedia.org/wiki/Protected_Extensible_Authentication_Protocol.

MICHAEL D. MCCOOL

achieving high performance by targeting multiple parallelism mechanisms



Michael McCool is an Associate Professor at the University of Waterloo and is also a co-founder and Chief Scientist of RapidMind, Inc., which produces a unified development platform for multicore processors and many-core accelerators. His background includes experience with and research publications in real-time computer graphics, medical imaging, hardware design, parallel computing, compiler design and implementation, and mathematics.

Michael.McCool@rapidmind.com

RECENTLY, PROCESSOR VENDORS HAVE begun increasing performance by adding additional cores, rather than increasing the performance of a single core. The addition of multiple cores augments several other parallel hardware mechanisms already in place on each core. These features create the potential for increased performance, but only if they are properly utilized. Programmers who disregard the underlying design of the hardware in newer processors can actually produce code that runs slower on a multicore processor. In this article, I explain what these design features are. I also discuss the underlying memory models and the impact they have on processing in a multicore context. Finally, I present an example of a method of writing abstract parallel code that allows a development platform to do the heavy lifting of implementing code for different processor architectures.

The trend toward increased hardware parallelism results from several factors, but basically it is not possible to scale clock rate owing to the excessive power required, because power requirements grow nonlinearly with clock rate. It is also simply not cost-effective to use the very large number of transistors that can fit on a modern chip for only one core. There just isn't enough to do, and it takes too long to get signals from one side of the chip to the other.

Several recent multicore processor designs have also used heterogeneous cores, in which some cores are tuned for specific tasks or workloads. In particular, not all processors have to be able to run the operating system and the user interface; they can instead be specialized for high-performance computation. At their targeted workloads, specialized cores can be orders of magnitude more efficient in terms of space and power than general cores. This is the case with the Cell BE processor and with GPUs, the processors found in video accelerator cards, for example. These processors also use a relatively large number of cores (since each core is simpler) and often provide more direct control over on-board memory, which is also a major factor in performance.

There are two major factors to consider when targeting high performance: parallelism and memory access. First, hardware is naturally parallel, and multicore processors make this painfully obvious. However, as we will discuss, there are in fact *many* hardware mechanisms already available besides multiple cores that exploit parallelism, and the performance advantages of these are multiplicative. To get the most out of modern processors and have any hope of running well on future processors a massively parallel approach needs to be considered from the start. Second, memory access can easily become a bottleneck even on single-core processors and the problem is even worse on multicore processors. To achieve even adequate performance on modern processors, programming practices and data structures need to be compatible with the structure of the memory system. Certain naive programming practices that conflict with the memory system can easily drop performance by one or even two orders of magnitude and can make it impossible to scale to a large number of cores.

Parallelism

Modern computer systems and processors actually use several forms of parallelism internally, in addition to multiple cores. To achieve maximum performance, it is often necessary to target several of these forms of parallelism simultaneously. This can be accomplished by designing parallel algorithms in an abstract form first. Once a good parallel algorithm has been designed, the abstract or “latent” parallelism in the algorithm needs to be decomposed and mapped onto the concrete parallelism mechanisms available in the target hardware.

To understand this better, let’s review the various forms of parallelism supported in modern processors and summarize how to take advantage of them.

MULTIPLE CORES

The most obvious form of parallelism available in modern processors consists of the multiple cores, of course. Every core is capable of executing independent instruction streams. These cores may or may not share a common memory subsystem. To make use of multiple cores, a workload needs to be decomposed into multiple components and each component run on the various cores. It is desirable to break the workload into equal-sized pieces so that the cores are evenly loaded; otherwise some cores will finish early and have to wait for the slowest core to complete. It may also be necessary to coordinate work among the cores, so that access to shared data is done in the right order. Because the number of cores can vary and is also increasing over time, an approach to decomposing the work that is adaptable to different numbers of cores is desirable. Data parallelism, which drives the decomposition by the structure of the data, is one approach that can accomplish this. An alternative way to achieve parallelism is to use decomposition by task, for instance, mapping different software modules onto different cores, although usually there are a limited number of different tasks available. These two approaches can be combined.

MULTIPLE PROCESSORS

When multiple processors are placed in a system, the number of available cores is the sum of the cores in all the processors. It is necessary to auto-

matically distribute the work over all the available cores, even if they are in separate processors. In addition, specific banks of the memory may also be associated with specific processors, and in this case accessing the memory associated with a specific processor will be more efficient from that processor. This property is called Non-Uniform Memory Access (or NUMA for short). For the best performance, it is useful to preferentially assign work units to processors closest to the memory banks where the needed data is located. This can be controlled by using processor affinity, which allows particular threads to be preferentially run on particular cores (although one has to be careful about the core numbering, since the mapping to physical cores and processors varies among vendors).

VECTOR INSTRUCTIONS

Many processors have special vector instructions that can operate on multiple elements of data at once. These are also called Single-Instruction Multiple Data (SIMD) operations. For instance, a processor may be able to apply a single arithmetic operation to 4-tuples of numbers, and that operation will take place in parallel on each element of the 4-tuple. A vector length of four is typical for single-precision floating point but it may be longer or shorter on different processors or for different data types. Examples of such instructions include AltiVec instructions on the PowerPC and the SSE instructions on x86 processors. If these special instructions are not used the benefit of this form of parallelism will not be realized. Also, different processors, even within the same “family,” may support different instruction set extensions. In particular, there are several generations of SSE instruction set extensions on x86 processors.

Instruction Pipelining

Many operations, in particular floating-point operations, may take multiple clock cycles to complete. The hardware breaks such operations into several stages, like an assembly line. For example, consider a floating-point addition. This is a surprisingly complex operation. Floating-point numbers are represented as in binary scientific notation, with both an exponent and a mantissa. To add two floating-point numbers, it is necessary to (1) compute the difference of the two exponents, (2) shift the mantissa of the smallest value down by this difference to align the “binary” point, (3) add the aligned mantissas, (4) shift the result mantissa so that it is in normalized form (with a leading 1), (5) round the result, and (6) renormalize the result (shifting down by one bit) if the highest bit was rounded up. This process can be implemented with separate hardware units for each step, with one unit feeding its result to the next on every clock pulse. As in an assembly line, several “jobs” (instructions, in this case) can be in the pipeline at the same time, as all the stages can operate in parallel. However, if an instruction depends on a previous result, then that instruction cannot begin until the result of the previous instruction is available. To keep the pipeline operating at maximum efficiency, there must be a large number of independent instructions available. If independent parallel tasks are available, they can be interleaved to avoid dependencies among instructions.

SUPERSCALAR INSTRUCTION ISSUE

Many processors can also start (“issue”) multiple instructions in the same clock cycle, as long as they do not depend on each other or use the same

hardware resources. For example, it might be possible to issue an instruction for an integer multiply and a floating-point addition in the same cycle, since they use separate hardware resources (with one using the integer multiplier and the other the floating-point adder).

Some processors will automatically issue multiple instructions simultaneously whenever possible. This is typical of mainstream desktop and server CPUs, which often have two-way or four-way superscalar instruction issue. Long instruction words may also be used to explicitly specify multiple operations at once. The latter approach is called a Very Long Instruction Word (VLIW) architecture. Current ATI/AMD GPUs are examples of the VLIW architecture, in which every core can issue five floating-point operations and one branch operation in every instruction. The Cell BE SPE cores can also be considered to have a VLIW architecture: Each instruction “pair” can issue one four-way SIMD floating-point operation and one integer, branch, or load/store operation in parallel.

As with pipelining, latent parallelism in an algorithm specification can be used to create independent instruction streams to make best use of this hardware feature.

ASYNCHRONOUS MEMORY TRANSFERS

Data can typically be transferred in and out of on-chip memory in parallel with computation, as long as the computation does not depend on the result of the transfer. This can be used to hide the latency of memory transfer. Different processors have different mechanisms for this; on CPUs, cache prefetching instructions are used. Prefetch instructions indicate that the contents of a given memory address in DRAM should be copied into cache in advance of when it will be used. On GPUs and the Cell, DMA transfers must be specified explicitly to move data between on-chip memory and external DRAM. In either case, to exploit this form of parallelism, the need for the data stored in a given memory location must be anticipated.

SIMULTANEOUS MULTITHREADING (HYPERTHREADING)

Some processors are able to run multiple threads on a single core. These additional threads look as though they are running on two or more “virtual” cores per real, physical core. In many ways, this can be considered an alternative interface to some of the other mechanisms for hardware parallelism already noted. Sometimes these threads are used to generate additional instructions for superscalar issue; sometimes the processor time-slices between the threads or switches between the threads on a memory stall in order to hide latency when data needed by a particular thread needs to be fetched from main memory. It is important to understand that simultaneous multithreading has very different performance characteristics from true multicore threading: It is usually a mechanism for sharing virtualized resources, not for accessing additional resources. It is important, therefore, to understand how processor affinities map threads to both real cores and “virtual” cores. Many times, if the code is carefully scheduled to use pipelining and superscalar issue, and to use prefetching, then multithreading on one core may not add any additional benefit. However, if each thread has a lot of control flow, it can be harder to schedule pipelined and superscalar code explicitly, and in this case multithreading on one core can be beneficial.

ACCELERATORS

Accelerators, which are additional non-CPU co-processors often with their own dedicated memory, such as GPUs, can execute a computation in parallel with the host CPU. If an operation is invoked that targets an accelerator, it is possible to start that operation asynchronously. Control can then be returned to the host program immediately even if the computation on the accelerator is not yet complete. The host process may then continue with additional operations that can execute in parallel with the computation running on the accelerator. However, if the host tries to read the result generated by an accelerator operation still in progress, the host process must wait until the accelerated operation is complete.

Memory

Multicore processors put a high demand on the memory system, and if care is not taken to use the memory system carefully, it can quickly become a bottleneck. The memory system consists of multiple types and forms of memory with different performance characteristics. The most important distinction is between on-chip and off-chip memory. On-chip memory is small but very fast, whereas off-chip memory (typically implemented using DRAM) is high capacity but slow. The number of clock cycles needed to read a data element from memory is called its *latency*. On-chip memory typically have single-digit latencies. Off-chip memory can have hundreds of cycles of latency. Bandwidth is often much higher to on-chip memory as well.

Typically a core can only operate at full speed when operating out of on-chip memory, which has a severely restricted capacity. Therefore on-chip memory is a critical resource and needs to be carefully managed.

Different processors take different approaches to managing on-chip memory. Caches are an automatic approach that makes management of the on-chip memory functionally invisible to the programmer. This is the approach taken by most general-purpose processors. However, the programmer still should take certain steps to make sure the cache performs well. In many cases, more efficiency can be gained if the programmer has direct access to and control of the on-chip memory, since then the use of this critical resource can be adapted to a specific application. This is the approach taken by the Cell BE processor in its specialized high-performance SPU cores: Each SPU core (out of eight total) has 256 kB of dedicated on-chip memory, and data must be explicitly transferred to and from external DRAM.

CACHE

Cache is a small, fast, usually on-chip memory in which copies of frequently used data are stored temporarily. In fact, there is typically a cache hierarchy, with very small, very fast cache memories right next to the processor that are actually caching data from another, slower and bigger cache lower in the hierarchy. Modern multicore processors can have up to three levels of cache, and data is moved between them automatically in response to the memory access patterns of the running program.

The purpose of cache is to reduce memory access latency *on average*. Reading a data item from off-chip DRAM takes, from the processor's point of view, hundreds of cycles. It will take only a few cycles to read that same data from cache. On every memory access, the processor checks whether a copy of the needed data is in the cache. If it is not, then it must wait until a copy of the

appropriate memory item can be read from a lower, slower level of memory, ultimately from off-chip DRAM. If such *cache misses* happen very infrequently, then on average, the memory access latency is closer to the time to read from the cache than to read from DRAM. Data is also transferred in relatively large blocks (on the order of hundreds of bytes) from DRAM, to amortize the overhead of setting up a memory transaction. A cache miss is only taken on the first access to a block. Later accesses to the same block will find the data already in the cache.

Eventually the cache fills up and blocks have to be replaced when space is needed to handle a new cache miss. If the block to be overwritten has been modified, it needs to be written back to main memory. Also, the hardware needs to select which block to discard. This is done by some simple rule; for instance, the block that has not been accessed for the longest time might be the one replaced.

Unfortunately, certain programming practices can defeat the cache, and cache may also not benefit some applications.

First, if only one element is ever read from every cache block loaded, then the cache is useless. In this case prefetching should be used to hide the memory access latency. Prefetching allows the processor to request a cache block sometime in advance of actually using it.

Second, as noted, data is actually transferred in blocks from main memory. If one element in a block is touched, the whole block is brought into the cache. If other nearby items in the same block are used by the program—a property called *spatial coherence*—then additional cache misses can be avoided. If they are not, then the bandwidth for transferring the rest of the block has been wasted. Therefore, programmers should select algorithms with good spatial coherence. Unfortunately, typical data structures based on pointers between many small memory records are not very good for cache performance. Pointer chasing leads to a lot of jumping around in memory and often results in poor spatial coherence.

Third, the processor has to be able to quickly check if data is in the cache. The hardware structure for this only allows a few locations in the cache to be used to hold copies of a large set of elements in main memory. Typically the locations of the elements in this set are offset by powers of two. If repeated accesses are made to the elements in the same set, they will fight over a very limited set of slots in the cache, a situation called *cache conflict*. The resulting *cache thrashing*, where items repeatedly replace one another, essentially disables the cache and can severely degrade performance.

Finally, if writes are made to data stored in cache, this data needs to be written back to DRAM eventually. Complications can arise if two cores with separate caches write to the same block of memory, or if one tries to write to a block another core is reading from. To maintain the illusion of a single unified memory space, these cores then have to keep track of which processor has the most up-to-date copy of the block. This involves a lot of hidden interprocessor communication, which can degrade performance. Some cache coherency protocols give one core ownership of a block, and only the owner may write to a block. However, if two cores simultaneously try to work on the same block, they can end up fighting over who owns it, with disastrous results for performance. This may occur even if the cores (or processors) are actually trying to modify different locations in the same block, a situation called *false sharing*.

These issues with cache are made more severe by multicore processors. There are additional levels of cache to worry about, and the aforementioned

effects can occur at one or all levels. Issues such as cache coherency and false sharing only arise in systems with multiple cores or processors. With the advent of multicore processors, off-chip memory bandwidth is not likely to grow as rapidly as on-chip computational performance, so off-chip bandwidth is even more likely to be a bottleneck. Finally, if a thread is suspended and restarted on a different core or processor, it will have to reload all its data into the cache on that core, possibly displacing data used by another thread. Yet another form of thrashing can take place between threads if together they need more data than will fit in the cache.

To avoid these issues, several steps need to be taken by the programmer. First, data should be allocated aligned to cache boundaries, and nodes of data structures should be padded if necessary to align to cache boundaries. This may waste some memory space but will avoid false sharing. Also, data structures that have good spatial coherence should be chosen over those with an excessive number of pointers. For example, a B-tree is often better than a simple binary tree, since a B-tree uses large, fixed-size blocks internally (which can be aligned to cache boundaries) and has a shorter number of pointer jumps from the root to its leaves. Finally, offsets between data elements that are a power of two should be avoided if possible. In image processing and matrix operations, for example, power-of-two tile sizes should be avoided by padding row lengths as necessary, because access to elements in adjacent rows may accidentally cause a cache conflict. Unfortunately, exactly what powers of two cause trouble and what alignments are needed vary by processor and the cache structure it uses. Also, avoiding large power-of-two offsets to avoid cache conflicts can be at odds with the desire to align to small powers of two for cache blocking. Some odd multiple of the cache block alignment should be selected.

EXPLICITLY MANAGED MEMORY

Cache is automatic, which is useful for naive code. However, to avoid the many issues that caches raise in multicore systems, some processors have opted for explicitly managed local memory. This is the case with the Cell BE processor, and also to some extent with GPUs (although current NVIDIA GPUs actually have both cache hardware *and* explicitly managed local memory).

In the Cell BE processor, each core gets a dedicated local memory. A separate Memory Flow Controller (MFC) can be programmed to transfer data to and from DRAM to this local memory, and also to and from other local memories on the same chip. These transfers can take place in parallel with computation.

A cache can still be simulated in software on such an architecture. Although slightly slower than a hardware cache, a software cache can be sized and tuned to the properties of the data structure it is caching. In particular, a block size and replacement policy can be chosen that are most suitable for the access patterns and data structures used.

Programming

We have now summarized the main hardware mechanisms available for exploiting parallelism in modern processors and also the properties of the memory system. It should be clear at this point that there is a lot “under the hood.”

Unfortunately, programming at this level of detail is very challenging, and consequently it is rarely done. Also, portable software may not be able to exploit a particular hardware feature, such as SIMD instructions, that is not consistently implemented on all hardware targets. As a result, most portable software is relatively inefficient.

The other point worth noting is that threading only targets a few of the levels of parallelism noted, and if not properly managed it can lead to inefficiencies in the memory system. Throwing a large number of threads at a multicore system and letting them fight over resources is unlikely to produce optimal results. Instead, a thread should just be seen as a mechanism for getting access to a single core, and then on that core appropriate steps should be taken to manage the memory and exploit the other forms of parallelism available. Steps should also be taken to avoid moving threads between cores (to avoid cache thrashing) and to keep computations close to the memory banks they are accessing in NUMA systems.

There are now several software development platforms that seek to reduce the complexity of programming multicore systems. The fundamental observation of these systems is that there are actually only two key abstract design principles that need to be targeted: parallelism and data locality. In particular, many *mechanisms* for implementing parallelism in hardware are available, but if a large amount of latent parallelism is available at an abstract level, it is not necessary for a programmer to target each mechanism individually. Instead, it is possible for a semiautomated system to map an abstract, portable programming model to whatever is available. Likewise, if an interface is provided in which the programmer can express an abstract version of data locality, then it can be mapped onto what the physical memory hardware requires.

To make this more concrete, we can look at an example from the RapidMind platform, which does just this. RapidMind is based on three types that can be used within standard C++, using existing compilers: values, arrays, and programs. A value represents a scalar type (e.g., a number or Boolean), arrays manage collections of data, and programs manage code. A sequence of operations on values can be stored in a program, then applied to a collection of data stored in an array.

First, we will declare some one-dimensional arrays to hold the data:

```
Array<1,Value1f> A, B;
```

We won't bother sizing or filling these arrays with data here, although in a real application this would have to be done.

Now we will construct a really simple example program to increment a value:

```
Program p = BEGIN {  
    In<Value1f> a;  
    Out<Value1f> b;  
    b = a + 1.0f;  
} END;
```

In a real application, such programs might contain thousands of operations and might include control flow, declarations of temporary variables (including local arrays), random accesses into other arrays, any number of inputs and outputs, and calls to C++ functions and other RapidMind programs. RapidMind programs can be thought of as dynamically constructed functions, for the most part.

Finally, we can apply the program to one of these arrays:

$$B = p(A);$$

This will apply the program to all the elements in A and place the result in B. As it happens, this will execute in parallel.

Applying a function to an array is a very simple way of invoking a parallel computation, conceptually. But what really goes on in the platform to execute this operation efficiently, given everything that we have discussed so far?

Conceptually, the parallelism intrinsic to this example is of the form shown in Figure 1.

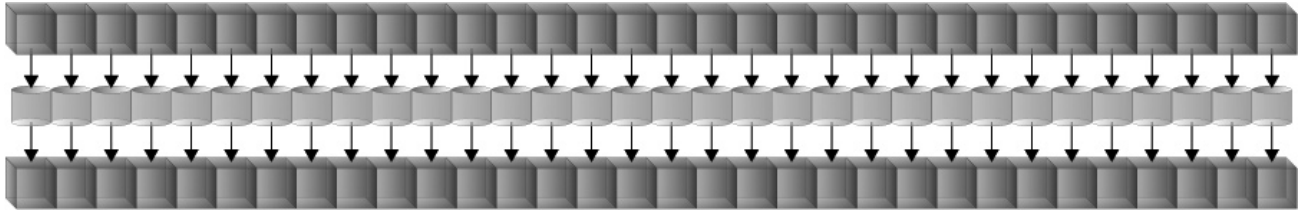


FIGURE 1: THE ABSTRACT PATTERN OF LATENT PARALLELISM SPECIFIED IN THE EXAMPLE

The important thing is that the semantics of program application provides a large amount of latent parallelism but has *not constrained the order in which these operations can be done or how they can be grouped*. Therefore the code generator and runtime system are free to reorganize them in any way that makes sense. For example, suppose we are targeting a two-core machine with a pipelined floating-point unit, four-way SIMD instructions, and two cores. The platform could then automatically organize this same computation as shown in Figure 2.

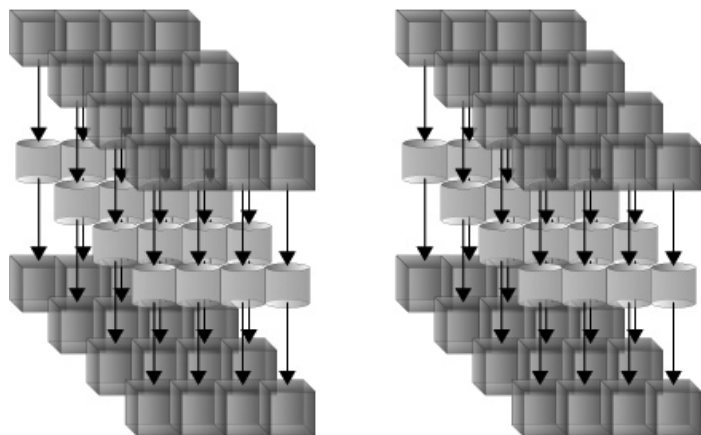


FIGURE 2: A COMBINATION OF CONCRETE PARALLEL MECHANISMS, INCLUDING SIMD INSTRUCTIONS, PIPELINING, AND MULTICORE EXECUTION, THAT COULD BE USED TO EXPLOIT THE LATENT PARALLELISM SPECIFIED IN THE EXAMPLE

Of course, if the hardware target changes, the code might have to be reorganized in a different way. For instance, a target with more cores, or a different SIMD width, might require a different decomposition. However, the code is portable, since the programmer has *not* constrained the computation to any particular ordering or decomposition. The code given here, for example, runs on various flavors of x86 multicore processors, the Cell BE SPUs, and

GPUs without change. Memory optimizations can also be made. The platform will break the work into blocks and prefetch one block into on-chip memory while working on another, work units can be broken into tiles that are suitable for the memory architecture and cache alignment, and arrays can be allocated with appropriate alignments and padding to avoid cache conflicts and false sharing. More complex code would require more complex transformations and management (e.g., control flow inside programs requires load balancing), but the same general principles apply.

Conclusion

Multicore processors are complex, but this complexity is in the form of several mechanisms that all fundamentally depend on two things: parallelism and data locality. It is possible to abstract away the complexity of multicore processors and still achieve high performance if abstractions are chosen that allow the programmer to focus on structuring computations around these two main concepts while not overconstraining the implementation. It is then possible to automatically reorganize the computation to exploit the various parallelism mechanisms available and optimize it for good memory behavior.

ADDITIONAL READING

The Editor suggests additional reading from past *login*: articles:

- [1] “Algorithms for the 21st Century,” by Steve Johnson:
www.usenix.org/publications/login/2006-10/openpdfs/johnson.pdf.
- [2] “Multi-Core Processors Are Here,” by Richard McDougall and James Loudon:
www.usenix.org/publications/login/2006-10/pdfs/mcdougall.pdf.
- [3] “Some Types of Memory Are More Equal Than Others,” by Diomedis Spinellis:
www.usenix.org/publications/login/2006-04/pdfs/spinellis.pdf.

DAVID N. BLANK-EDELMAN

practical Perl tools: back in timeline



David N. Blank-Edelman is the Director of Technology at the Northeastern University College of Computer and Information Science and the author of the O'Reilly book *Perl for System Administration*. He has spent the past 22+ years as a system/network administrator in large multiplatform environments, including Brandeis University, Cambridge Technology Group, and the MIT Media Laboratory. He was the program chair of the LISA '05 conference and one of the LISA '06 Invited Talks co-chairs.

dnb@ccs.neu.dnb@ccs.neu.edu

YOU KNOW, I WAS JUST MINDING MY own business, reading my email and stuff, when the following message from the SAGE mailing list came on my screen (slightly excerpted but reprinted with permission):

From: millerj@metro.dst.or.us
Date: January 9, 2008 2:10:14 PM EST
Subject: Re: [SAGE] crontabs vs /etc/
cron.[daily,hourly,*] vs /etc/cron.d/

On a more specific aspect of this (without regard to best practice), does anyone know of a tool that converts crontabs into Gantt charts? I've always wanted to visualize how the crontab jobs (on a set of machines) line up in time. Each entry would need to be supplemented with an estimate of the duration of the job (3 minutes vs 3 hours).

JM

I just love sysadmin-related visualization ideas. This also seemed like a fun project with some good discrete parts well-suited to a column. Let's build a very basic version of this project together. For the purpose of this discussion I'm going to make the assumption that you already know what a crontab file is, what it contains, and what it does for a living. If not, please consult your manual pages about them and cron (try typing something like `man 5 crontab` or just `man crontab`).

Chewing on the Crontab File

The first subtask that comes up with this project is the parsing and interpretation of a standard crontab file. The easy part will be to read in the file and have our program make sense of the individual fields in that file. Having a crontab sliced and diced into nice bite-sized (read: object) pieces doesn't help us all that much, because our end goal is to be able to plot what happens when cron interprets those pieces. Cron looks at that file and decides when a particular command should be run. We'll need some way to determine all of the times cron would have run a particular line during some set time period.

For example, let's say we take a very basic crontab file like this:

```
45 * * * * /priv/adm/cron/hourly
15 3 * * * /priv/adm/cron/daily
15 5 * * 0 /priv/adm/cron/weekly
15 6 1 * * /priv/adm/cron/monthly
```

Every 45 minutes, the `/priv/adm/cron/hourly` program is run, so we'll be plotting that event at 1:45, 2:45, 3:45, and so on. At 3:15 in the morning each day we run `/priv/adm/cron/daily`, and so on.

Figuring all of this out seems doable, but, truth be told, kind of a pain. Luckily we've been spared that effort because Piers Kent wrote and published the module `Schedule::Cron::Events`, which makes this subtask super easy. It calls upon another module to parse a crontab line (`Set::Crontab` by Abhijit Menon-Sen) and then provides a simple interface for generating the discrete events we'll need.

To use `Schedule::Cron::Events`, we'll need to pass it two pieces of information: the line from crontab we care about and some indication of when we'd like `Schedule::Cron::Events` to begin calculating the events created by that crontab line:

```
my $event = Schedule::Cron::Events( $cronline, Seconds => {some time} );
```

(where {some time} is provided using the standard convention of describing time as the number of seconds that have elapsed since the epoch).

Once you've created that object, each call to `$event->nextEvent()` returns back all of the fields you'd need to describe a date (year, month, day, hour, minutes, second).

Now that we understand how to deal with this subtask, let's move on to the others. We'll put everything together at the end.

Displaying the Timeline

Creating a pretty timeline is a nontrivial undertaking, so let's let someone else do the work here for us as well. There are decent Perl timeline representation (`Data::Timeline`) and display (`Graph::Timeline`) modules available, but there's one way to create timelines that are so spiffy that I'm actually going to forsake the pure-Perl solution. I think the Timeline (as they put it) "DHTML-based AJAXy widget for visualizing time-based events" project from the SIMILE project at MIT is very cool and a good fit for this project. More info on it can be found at <http://simile.mit.edu/timeline/>. To give you an idea of what Timeline's output looks like, see the excerpt from Monet's life shown in Figure 1.



FIGURE 1: TIMELINE MONET EXAMPLE SCREENSHOT

To make use of this widget we need to create two files: an HTML file that sucks in the widget from MIT, initializes it, and displays it and an XML file containing the events we want displayed. That last part will be our third

challenge, which we'll address in the next section. In the meantime, let me show you the HTML file in question. I should mention that my Javascript skills are larval at best; most of the following is cribbed from the tutorial found at the URL provided above. If this is all gobbledygook to you, feel free to just read the comments (marked as `<!-- -->` and `//`).

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01//EN">
<html>
  <head>
    <!-- Reference the widget -->
    <script src="http://simile.mit.edu/timeline/api/timeline-api.js" type="text/javascript">
    </script>

    <script type="text/javascript">
function onLoad() {
  // tl will hold the timeline we're going to create
  var tl;
  // get ready to specify where we'll get the data
  var eventSource = new Timeline.DefaultEventSource();

  // Create a timeline with two horizontal bars, one displaying
  // the hours, the other the days that contain the hours.
  // Note: both bands are set to display things relative
  // to my timezone (-5 GMT).
  var bandInfos = [
    Timeline.createBandInfo({
      eventSource:  eventSource,
      timeZone:    -5, // my timezone in Boston
      width:       "70%",
      intervalUnit: Timeline.DateTime.HOUR,
      intervalPixels: 100 }),
    Timeline.createBandInfo({
      timeZone:    -5,
      width:       "30%",
      intervalUnit: Timeline.DateTime.DAY,
      intervalPixels: 100 }),
  ];

  // keep the two bands in sync, highlight the connection
  bandInfos[1].syncWith = 0;
  bandInfos[1].highlight = true;

  // ok, create a timeline and load its data from output.xml
  tl = Timeline.create(document.getElementById("cron-timeline"), bandInfos);
  Timeline.loadXML("output.xml", function(xml, url) { eventSource.loadXML(xml, url); });
}

// boilerplate code as specified in the tutorial
var resizeTimerID = null;
function onResize() {
  if (resizeTimerID == null) {
    resizeTimerID = window.setTimeout(function() {
      resizeTimerID = null;
      tl.layout();
    }, 500);
  }
}
    </script>
    <title>My Test Cron Timeline</title>
  </head>
```

```

<!-- run our custom code upon page load/resize -->
<body onload="onLoad();" onresize="onResize();">

    <!-- actually display the timeline here in the document -->
    <div id="cron-timeline"
        style="height: 150px;
        border: 1px solid #aaa">

    </div>

</body>
</html>

```

To avoid repeating the explanation for each part of this file as it is described in the Timeline tutorial, let me just refer you to that Web page instead.

The one last non-Perl thing I need to show you to complete this subtask is an example of the event data we'll need (in a file called output.xml). This will give you an idea of which data the widget is expecting us to provide. Here's an example that assumes we're showing the cron events for January 2008:

```

<data>
<event start="Jan 01 2008 00:45:00 EST" title="/priv/adm/cron/hourly"></event>
<event start="Jan 01 2008 01:45:00 EST" title="/priv/adm/cron/hourly"></event>
<event start="Jan 01 2008 02:45:00 EST" title="/priv/adm/cron/hourly"></event>
<event start="Jan 01 2008 03:45:00 EST" title="/priv/adm/cron/hourly"></event>
...
<event start="Jan 01 2008 03:15:00 EST" title="/priv/adm/cron/daily"></event>
<event start="Jan 02 2008 03:15:00 EST" title="/priv/adm/cron/daily"></event>
<event start="Jan 03 2008 03:15:00 EST" title="/priv/adm/cron/daily"></event>
<event start="Jan 04 2008 03:15:00 EST" title="/priv/adm/cron/daily"></event>
...
<event start="Jan 06 2008 05:15:00 EST" title="/priv/adm/cron/weekly"></event>
<event start="Jan 13 2008 05:15:00 EST" title="/priv/adm/cron/weekly"></event>
<event start="Jan 20 2008 05:15:00 EST" title="/priv/adm/cron/weekly"></event>
<event start="Jan 27 2008 05:15:00 EST" title="/priv/adm/cron/weekly"></event>
<event start="Jan 01 2008 06:15:00 EST" title="/priv/adm/cron/monthly"></event>
</data>

```

Hmm, writing an XML data file: how do we do that? Read on.

XML Output with No Effort

So far we've vanquished the tricky parts of the project having to do with determining which data we need and what will consume this data. The last part is to make sure we format the data in a form that will work. In this case we're looking to create an XML file with specific tags and contents. There are a whole bunch of Perl ways to generate XML files, ranging from simple print statements to fairly complicated event-driven frameworks. The one that probably best serves our rather meager needs for this project is the use of the module XML::Writer. It makes it easy to produce XML that has properly matched tags, each with the correct attributes. This mostly requires code something like this:

```

# set up a place to put the output
my $output = new IO::File(">output.xml");

# create a new XML::Writer object with some pretty-printing turned on
my $writer
    = new XML::Writer( OUTPUT => $output, DATA_MODE => 1, DATA_INDENT => 2 );

```

```

# create a <sometag> start tag with the given attributes
$writer->startTag('sometag', Attribute1 => value, Attribute2 => value );

# just FYI: we could leave out the tag name here and it will try to
# figure out which one to close for us
$writer->endTag('sometag');

$writer->end();
$output->close();

```

Putting It All Together

Congrats: we've now seen all of major pieces and we're ready to show the "final" code. I'll only explicate the pieces of the code that are new to the discussion.

PART ONE: LOAD THE MODULES

```

use strict;
use Schedule::Cron::Events;
use File::Slurp qw( slurp );      # we'll read the crontab file with this
use Time::Local;                  # needed for date format conversion
use POSIX;                        # needed for date formatting
use XML::Writer;
use IO::File;

```

PART TWO: SET US UP CHRONOLOGICALLY

We're going to have to tell `Schedule::Cron::Events` where to begin its event iteration. Basically, we have to pick a start date. It seems as though it might be useful to display a timeline showing the events for the current month, so let's calculate the seconds from the epoch at the beginning of the first day of the current month:

```

my $currentmonth = ( localtime( time() ) )[4];
my $currentyear  = ( localtime( time() ) )[5];
my $monthstart   = timelocal( 0, 0, 0, 1, $currentmonth, $currentyear );

```

PART THREE: READ THE CRONTAB FILE INTO MEMORY

```

my @cronlines = slurp('crontab');
chomp(@cronlines);

```

PART FOUR: CREATE AND START THE XML OUTPUT FILE

```

my $output = new IO::File(">output.xml");
my $writer
    = new XML::Writer( OUTPUT => $output, DATA_MODE => 1,
                      DATA_INDENT => 2 );

$writer->startTag('data');

```

PART FIVE: LA MACHINE (THE ACTUAL WORK)

We've now hit the place in the code where the actual iterating over the contents of the crontab file takes place. As we iterate, we need to enumerate all of the events produced by each line we find. Because `Schedule::Cron::Events` is happy to provide `nextEvent()`s ad infinitum, we'll have to pick an arbi-

trary time to stop. As mentioned before, showing a month seems like a good timespan, so our code stops asking for `nextEvent()` as soon as that call returns something not in the current month.

Let's look at this iteration:

```
foreach my $cronline (@cronlines) {
    next if $cronline =~ /^#/;
    my $event
    = new Schedule::Cron::Events( $cronline, Seconds => $monthstart );
```

For each line in the crontab that is not a comment, we hand that line off to `Schedule::Cron::Events` with a start time of the beginning of the current month.

Then we iterate for as long as we're still in the current month:

```
while (1) {
    @nextevent = $event->nextEvent;
    # stop if we're no longer in the current month
    last if $nextevent[4] != $currentmonth;
```

For each event, we're going to want to generate an `<event>` element with the `start` attribute showing the time of that event and the `title` attribute listing the command cron would run at that time. We'll be calling the `strftime()` function from the `POSIX` module to get the date formatted the way the Timeline widget likes it:

```
$writer->startTag('event',
    'start' => POSIX::strftime('%b %d %Y %T %Z',@nextevent),
    'title' => $event->commandLine(),
);
$writer->endTag('event');
```

We could add an `end` attribute to this element if we knew how long each event would last. Unfortunately, there is no easy way to know or estimate the length of time a particular cron job takes (as suggested in the email that started this column). However, you could imagine writing more code to analyze past crontab logs to try to guess that information. Yes, this is one of those dreaded "This exercise is left to the reader" moments.

That's basically it. We now just need to close the Perl loops, close the outer tag in the XML file, stop `XML::Writer`'s processing, close the file itself, and we're done:

```
}
}
$writer->endTag('data');
$writer->end();
$output->close();
```

So, how's this look? Figure 2 shows a screenshot from the widget when loaded into a browser using our newly created data file.

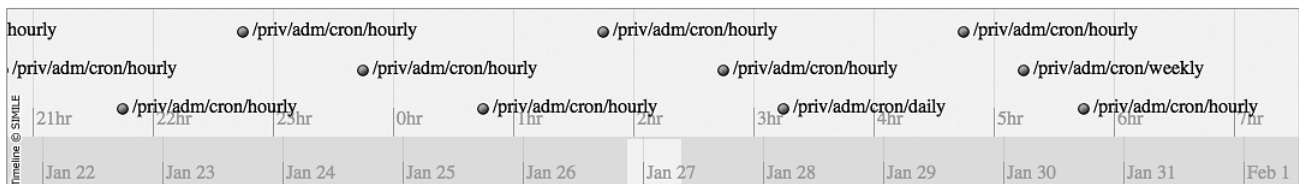


FIGURE 2: TIMELINE FROM A SIMPLE CRONTAB

Trust me, it's cooler in person, because you can scroll back and forth in the month.

I realize that this code doesn't fulfill the original correspondent's wishes because, number 1, it's not a GANTT chart (which would require analyzing the different cron jobs and seeing how they connect) and, number 2, it doesn't show multiple machines overlaid.

Defect number 1 turns out to be pretty hard to remedy. As Richard Chycoski pointed out in a follow-up to this message, dependency tracking in this context gets you into the fairly complex "batch processing" world, something we can't address in this column. Luckily, defect number 2 is pretty easy to fix; it just requires opening more than one crontab file and doing the same work on each file. That's actually a reasonable exercise for the reader with which to leave you without feeling guilty, so have at it.

Even with these defects the diagram seemed pretty spiffy to me. I wanted to see what would happen if I fed the script real-world data from another site. I contacted John, the writer of my opening email message, and he was kind enough to send me a set of crontabs including one that he described as follows: "These jobs are in use at Metro, producing space utilization reports for our NetApp, driving the cold backup sequence for Oracle databases, and other system tasks." Running my code against this crontab file (and changing the HTML file that displays it so it has a larger display area) yields the results in Figure 3, which John describes as "Sweet!"

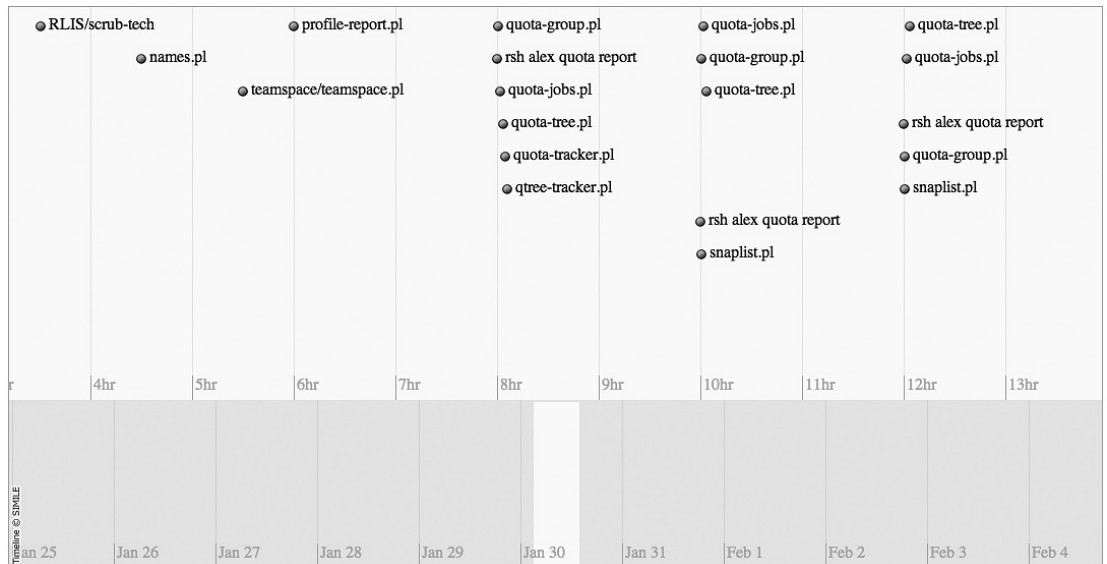


FIGURE 3: TIMELINE FROM A REAL-WORLD CRONTAB

Hopefully this fun little example has given you some tools both for working with crontabs and for creating timelines. I'm certain there are some more interesting offshoots of this idea just waiting for you to find them. Take care, and I'll see you next time.

PETER BAER GALVIN

Pete's all things Sun (PATS): the security sheriff



Peter Baer Galvin (www.galvin.info) is the Chief Technologist for Corporate Technologies, a premier systems integrator and VAR (www.cptech.com). Before that, Peter was the systems manager for Brown University's Computer Science Department. He has written articles and columns for many publications and is coauthor of the *Operating Systems Concepts* and *Applied Operating Systems Concepts* textbooks. As a consultant and trainer, Peter teaches tutorials and gives talks on security and system administration worldwide.

pbg@cptech.com

SECURITY IS A CONSTANT, ONGOING activity for most system administrators. While installations, patches, problems (fire fighting), and upgrades are usually in the sysadmin foreground, security is that constant background companion, at least at many facilities. In fact, the best sysadmins have a mental checklist they execute before hitting the <return> key on the command line, or pressing the <apply> button in a GUI (which brings up the question of whether the best sysadmins use any GUIs, but let's leave that for another time). The mental checklist is:

Checklist #1: Use before making a change.

- Is the syntax of the command correct?
- Is the command the right one to make the change?
- Is there a better way to make the change?
- Are the right options entered or selected?
- Is today Friday?
- Is today some other day on which it would be exceptionally bad to break something (such as the day before leaving for a vacation or conference)?
- What are the chances that executing this will break something?
- If this change would break something, can I undo the action?
- Is this a documented way to accomplish the task?
- If this is a new way to make a change, should I document it?
- And finally, what effect might this action have on security?

Only after this mental checklist is run would this mythical best sysadmin execute the action. Of course the best sysadmins would modify this list to suit their circumstances, site policies, and abilities.

If you care about security, and you are a good sysadmin, then not only do you consider your actions in a security context but you keep an eye open for ways to improve security without increasing your workload—which brings us to this month's topic.

The CIS Solaris Benchmark

Now, since you are the mythological “best sysadmin” that I've been talking about, you are already

going through another mental checklist. This is the one you execute when a new tool is proposed to you:

Checklist #2: Use before trying a new tool.

- Do I already have a better tool?
- Is it multiplatform or one-off?
- Does it work, or just cause more work?
- Is it kept up-to-date?
- Does it change too often, causing more work?
- How much does it cost?
- Do I already know it or is it at least easy to learn?
- Is it likely to break or break something? (Go back to checklist #1.)

In the case of the Center for Internet Security (CIS) [1], the answers to these questions are all the right ones. CIS publishes “benchmarks” for many operating systems and applications. They are reasonably priced for many uses and easy to use, and I believe they are among the best security tools that you can apply to your environment.

CIS is a nonprofit organization, but it does need funds to support its various activities. Membership is one form of funding and well worth considering. Organizations such as CIS are doing their part to improve the overall security of computing infrastructure. Many people feel security is too lax in general, and that lax core security wastes time and money as security is monitored and breaches are detected and mitigated. Membership in CIS provides you or your organization with a chance to help improve the state of security—in other words, a way to stop complaining and start helping. The benchmark documents are only for noncommercial use, but commercial use licenses are available. The benchmark tool is currently available for trial use; full use requires membership.

Each benchmark is platform-specific. For this column I will stick to the Solaris 10 Benchmark, but there are many others. Each benchmark comes as a document describing recommended security steps, plus an appendix including variations and more advanced security steps that are not recommended for all sites or all circumstances. Many benchmarks also come with a tool that runs an audit of a given system and calculates a security score. The resulting score can be compared to the score of the same system from a previous run, to the scores of other systems, or to the theoretical best score.

The tool included with the benchmark is “read-only” in that it should not make changes to the system. Rather, the benchmark documents and tools recommend changes that should be considered for improving security. Sun has taken the unusual step of supporting the use of the Solaris CIS Benchmark, in that any changes it recommends are supported by Sun. You can call Sun support if you have questions or problems regarding any changes you made based on the benchmark recommendations. (Glenn Brunette, a security-centric Sun Distinguished Engineer, has a nice blog posting about all of this [2].)

First Steps

CIS has many benchmarks available. Navigate the site to find the ones you are interested in. Before you can download any of the CIS assets, you must agree to its license and also fill out a form about you and your organization.

For Solaris, there are several available files to choose from. For Solaris 10 11/06 and 8/07, the best starting place is CIS_Solaris_Benchmark_v4.0. Included is the benchmark document containing recommendations and an

appendix with an overview of Solaris 10 security controls. Carole Fennelly edited the document with input from many security experts, and it is an excellent Solaris 10 security resource. (Full disclosure: Carole and I have worked together on projects, and I was among the beta reviewers of this document.)

The 89-page document is one of the best security documents available. It includes many recommendations on how to improve the out-of-the-box security of Solaris 10. Even though Solaris 10 is initially fairly secure, there are many steps recommended to improve that security. For each recommendation there is information about what hardware platforms it pertains to, if it is the OS default, if the change applies to zones or just the global zone, and if the Solaris Security Toolkit can be used to make the change. Also included is information on how the recommendation affects the security score of the system, how to implement the recommendation, and any notes regarding the recommendation. This completeness of information helps both novice and advanced sysadmins decide whether to implement the recommendation and, if so, how to do so.

Another tool to run through your mental checklists is “The Solaris Security Toolkit” [3], a freely available and supported tool from Sun. This tool not only audits but also can implement configuration changes. Its execution and configuration can be scripted to allow groups of systems to be configured similarly and checked for differences from that security configuration. For another useful site see [4], where custom scripts built around the toolkit are collected together.

You can obtain the Solaris Security Toolkit 4.2 documentation from [5] (assuming you are one of the mythical sysadmins who read documentation before mucking around with a tool!). Another nice guide to the toolkit comes in the form of Sun blueprints [6].

The Benchmark Tool

On the Solaris page of CIS, there are several older tools designed for specific Solaris releases. These tools are not supported by CIS and tend to be out of date and buggy. Rather than use those, head to the CIS front page [1] and navigate to CIS-CAT. This tool is written in Java, and it parses an XML file containing the tests to run for a given platform. The `ux-xml` tarball that comes with the benchmark holds the XML files. Using the same tool and the appropriate XML file gives you the flexibility to run multiple tests on multiple systems from this starting point. At the time of this writing the tool benchmarks the following platforms:

- SuSE
- Slackware
- Red Hat Enterprise Linux
- Solaris 10
- AIX
- Oracle 9i/10g (for Windows)
- Oracle 9i/10g (for UNIX)
- Windows XP
- Windows Server 2003

For my testing I used Solaris Nevada x86 build 81 (running within a VMWare virtual machine on top of Mac OS X Leopard). Use of the tool is very easy, and it is well documented by CIS. Start the tool with the provided shell script, use the “file” menu to load the appropriate XML file, and again use

the “file” menu to run the benchmark (Figure 1). A few minutes later (even in this virtualized environment) a report is generated showing the results of the run. The result is reported in XML and HTML. One feature not working in my testing was the “file->browse results” menu item in the benchmark tool. Rather, I manually viewed the results files in /SYS-CAT_Results/.

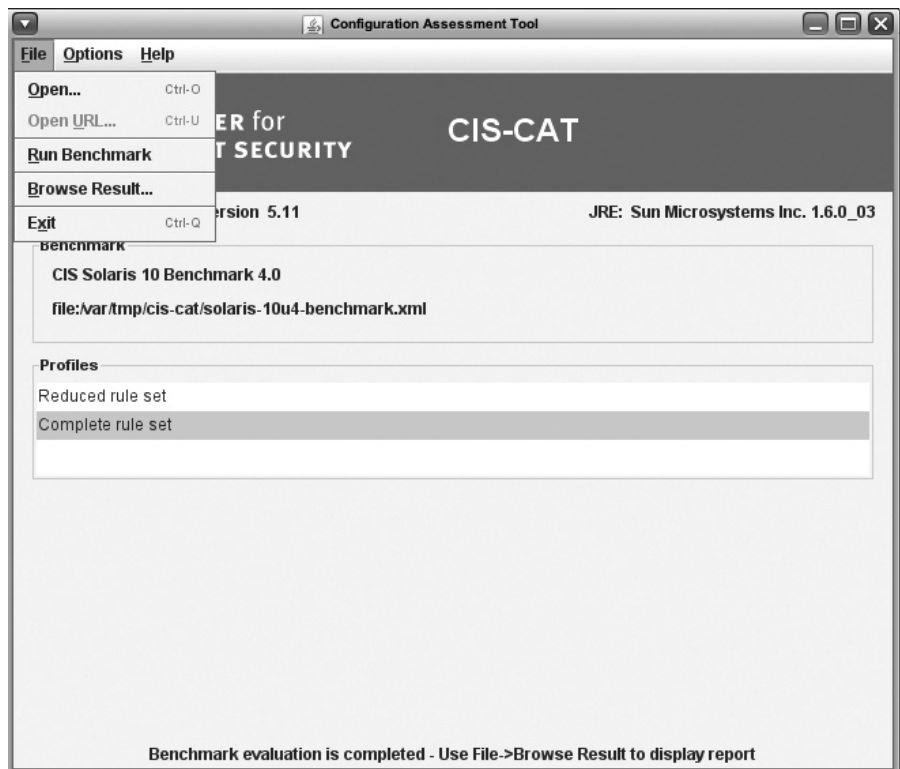


FIGURE 1: THE INITIAL SCREEN OF CIS-CAT

A tool is only as good as the usefulness of its results, and in the case of the CIS benchmark tool, the results are invaluable. Figure 2 shows the top of the report including summary information. Notice that on a generic current Solaris Nevada installation, the benchmark score is only 30%. Further down in the report are details on each test run, the results of the test, and a link to an explanation of the test, a link to the XML code that was executed, and recommendations to improve the security of the system based on the aspect tested (Figure 3). In essence, the report is a roadmap and how-to guide for improving security on a given system. Further, by being able to rerun the test and rescore the system, any changes made can be evaluated for their correctness and efficiency. This is a useful tool indeed.

Description	Items					Flat Model		
	P	F	U	j	Actual	Max	Score	
1 Install Patches and Additional Software	0	0	0	0	2	0.0	0.0	0%
2 Restrict Services	10	12	0	0	13	10.0	22.0	45%
2.1 Establish a Secure Baseline	0	0	0	0	1	0.0	0.0	0%
2.2 Disable Unnecessary Local Services	1	6	0	0	0	1.0	7.0	14%
2.3 Disable Other Services	9	5	0	0	0	9.0	14.0	64%
2.4 Enable Required Services	0	0	0	0	12	0.0	0.0	0%
2.5 id-2.5g (untitled)	0	1	0	0	0	0.0	1.0	0%
3 Kernel Tuning	0	5	0	0	0	0.0	5.0	0%
4 Logging	1	8	0	0	0	1.0	9.0	11%
5 File/Directory Permissions/Access	3	5	0	0	0	3.0	8.0	38%
6 System Access, Authentication, and Authorization	2	10	0	0	1	2.0	12.0	17%
7 User Accounts and Environment	8	8	0	0	0	8.0	16.0	50%
8 Warning Banners	0	7	0	0	0	0.0	7.0	0%
9 Appendix A: File Backup Script	0	0	0	0	0	0.0	0.0	0%
10 Appendix B: Service Manifest for /var/svc/method/cis_netconfig.sh	0	0	0	0	0	0.0	0.0	0%
11 Appendix C: Additional Security Notes	0	0	0	0	6	0.0	0.0	0%
12 References	0	0	0	0	0	0.0	0.0	0%
Total	24	55	0	0	22	24.0	79.0	30%

FIGURE 2: THE CIS-CAT REPORT SUMMARY TABLE

g	A	B	w	Benchmark Item	Result
1				Install Patches and Additional Software	
?	++			1.1 Apply Latest OS Patches	notchecked
?	++			1.2 Install Solaris 10 Encryption Kit	notchecked
2				Restrict Services	
2.1				Establish a Secure Baseline	
?	++			2.1.1 Establish a Secure Baseline	notchecked
2.2				Disable Unnecessary Local Services	
F	++	1.0		2.2.1 Disable Local CDE ToolTalk Database Server	fail
F	++	1.0		2.2.2 Disable Local CDE Calendar Manager	fail
F	++	1.0		2.2.3 Disable Local Common Desktop Environment (CDE)	fail
F	++	1.0		2.2.4 Disable Local sendmail Service	fail
F	++	1.0		2.2.5 Disable Local Web Console	fail
F	++	1.0		2.2.6 Disable Local WBEM	fail
P	++	1.0		2.2.7 Disable Local BSD Print Protocol Adapter	pass
2.3				Disable Other Services	
P	++	1.0		2.3.1 Disable RPC Encryption Key	pass
P	++	1.0		2.3.2 Disable NIS Server Daemons	pass
P	++	1.0		2.3.3 Disable NIS Client Daemons	pass
P	++	1.0		2.3.4 Disable NIS+ daemons	pass
P	++	1.0		2.3.5 Disable LDAP Cache Manager	pass
F	++	1.0		2.3.6 Disable Kerberos TGT Expiration Warning	fail
F	++	1.0		2.3.7 Disable Generic Security Services (GSS) daemons	fail
F	++	1.0		2.3.8 Disable Volume Manager	fail
P	++	1.0		2.3.9 Disable Samba Support	pass
F	++	1.0		2.3.10 Disable automount daemon	fail
P	++	1.0		2.3.11 Disable Apache services	pass
P	++	1.0		2.3.12 Disable Solaris Volume Manager Services	pass
P	++	1.0		2.3.13 Disable Solaris Volume Manager GUI	pass
F	++	1.0		2.3.14 Disable Local RPC Port Mapping Service	fail
2.4				Enable Required Services	
?	++			2.4.1 Enable Kerberos server daemons	notchecked
?	++			2.4.2 Enable NFS server processes	notchecked
?	++			2.4.3 Enable NFS client processes	notchecked
?	++			2.4.4 Enable telnet access	notchecked

FIGURE 3: A CIS-CAT REPORT DETAILS SECTION

Conclusions

Security is a necessary part of most sysadmins' lives. Generally, security is a complicated annoyance, involving keeping an eye open for frequently changing security recommendations and trying to make those changes on all of the systems within a facility. Multiply this challenge by the number of platforms, applications, and versions of all of the above, and even the best system administrators have a difficult time keeping up.

One way to improve the security situation is to apply a tool to the problem. The best sysadmins consider all aspects of the tool and its impact on their environment before going down that path. Certainly sites and priorities vary, but for most sites and most administrators, a tool such as the Center for Information Security benchmark (both the benchmark document and the Java tool) is a clear improvement over the status quo:

- It is low-cost or free.
- It covers many platforms.
- It is easy to install and use.
- Its results are very useful.
- It effects no changes to the system.
- It is written by experts on the subject.
- It comes with documentation that teaches improved security.
- It is updated frequently.

The CIS organization and its documentation and tool benchmarks get my highest recommendation for utility, practicality, and overall ability to help sysadmins improve site security and maintain that improved security.

As an aside, the best sysadmins tend to be an opinionated group. Feel free to send me your own mental checklists or improvements to the ones I included in this column. A future column collecting these checklists could be very enlightening.

Next Time

Given that the theme of the next *login*: issue is storage, and Solaris 10 comes with a free, open source, breakthrough file system, it seems fitting that ZFS should be the topic of PATS. As ZFS has been discussed previously in *login*:, I'll start by updating the status and features list, then discuss field experiences and use cases, and finish with a look into the future of ZFS.

REFERENCES

- [1] <http://www.cisecurity.org/>.
- [2] http://blogs.sun.com/gbrunett/entry/cis_solaris_10_security_benchmark1.
- [3] Available at <http://www.sun.com/security/jass>.
- [4] <http://www.ip-solutions.net/jass/>.
- [5] http://www.sun.com/products-n-solutions/hardware/docs/Software/enterprise_computing/systems_management/sst/index.html.
- [6] <http://www.sun.com/software/security/blueprints/index.html>.

DAVID JOSEPHSEN

iVoyeur: comply



David Josephsen is the author of *Building a Monitoring Infrastructure with Nagios* (Prentice Hall PTR, 2007) and is Senior Systems Engineer at DBG, Inc., where he maintains a gaggle of geographically dispersed server farms. He won LISA '04's Best Paper award for his coauthored work on spam mitigation, and he donates his spare time to the SourceMage GNU Linux Project.

dave-usenix@skeptech.org

I MISS THE BYGONE DAYS WHEN

everyone thought security was a product. You almost never had to deal with vendors beyond trying to convince management that openbsd, pf, and snort were the same no matter how pretty a box the vendor had found to put them on. When security was a product, all it took to placate panic-stricken knee-jerkers was a few tens of thousands of dollars on perimeter appliances, and, when they were placated, we were left alone, free to go back to the actual work of building secure infrastructure and maintaining secure systems.

Then Bruce had to go write that infernal book. You know the one I mean: the one that rhymes with “tree kits and pies.” Now everybody thinks security is a process. Now the vendors never go away. Worse, in some sick pantomime of vigilance we pay them to hang around, poking us with their brain-dead scanners and generating nonsensical 50-page lists of prioritized nonvulnerabilities. Suddenly we’re surrounded by standards. Now we have CoBIT, COSO, FFIEC, FISMA, GLBA, HIPAA, ISO17799/BS7799, NISPOM, PCI, and SOX to waste time on, incentivizing management to aspire to minimal baselines of system security and to postpone everything in the name of creating hundreds of pages of policies that no one (including the auditors) will ever read and that are hopelessly impossible to actually adhere to.

I’m terribly sorry, but I happen to be armpits-deep in PCI compliance, and I got a bit carried away there. I understand the intent of the policies and documentation (mostly). Lawyers are a wholly different and dangerous kind of script kiddie, and the documentation mitigates them. And of course I see what the community is trying to do in terms of third-party verification and base-line standardization. People can’t know what security looks like unless there are some guidelines. That’s all great and everything, except that it isn’t working. Security is not and has never been a product or a process; it’s a state of mind. At the end of the day, you “get it” or you don’t, and that mostly boils down to who you have working for you and what you’re trying to accomplish.

So while I see the point of the records-keeping aspects of the standards, I can’t help feeling sometimes that the systems security aspects are better left to the market. Companies that want to do busi-

ness with each other should ensure that they meet each other's internal criteria for systems security and leave it at that. Those that know what they're doing will do well and those that don't won't. Whatever sickeningly large amount of money is currently going to the myriad horde of mediocre security vendors would be better spent on getting, building, maintaining, and keeping good people.

In the "security as a process" universe, what happens instead is that companies partner with each other using PCIDSS compliance as the criterion and people like you and me end up having to deal with business partners who don't know what it means to verify an ssh key fingerprint. In the "security as a process" universe, companies still buy security; it just costs a lot more, they never stop paying, and it's much more verbose. My biggest gripe with standardization in this context has to be that, in the minds of the executives, compliance equates to security. Meet the minimal requirements of the standard and you are—by definition—secure (and I think this idea is actively encouraged by the aforementioned mediocre security vendors). Here's a hint: If you think you can safely carry out transactions of sensitive data with me because I said "yes" when you asked, "Are you PCI compliant?" you have some security problems. They are severe in scope, trivial to exploit, and of a type that won't show up on the port scans.

There I go getting carried away again. It's been a bad week. Sorry. Anyway, I'm sure compliance is a challenge for pretty much everyone out there, and the company I currently work for is no exception. Being a rather small operation, we're challenged mostly by the documentation aspects of the standards, but we've also run into quite a few requirements that assume a much larger body of employees than we possess. In each of these cases, one of the various monitoring systems we currently employ has satisfied the requirement handily. This of course brings me (in only six paragraphs) to the actual point of this particular article: monitoring tools that can help you comply.

Most of the requirements I'm talking about are audit-related, and much of the advice I'm about to give is probably advice you've heard before about tools you've heard about before. It's all obvious stuff in little pieces, but when it all comes together it loses transparency, so my intent here is to provide a short list of things you should have implemented before going into an audit in order to ensure that as many requirements as possible are met in an automated fashion. The PCIDSS, for example, has a slew of requirements around making sure the policies are being followed by manually auditing various aspects of the environment. Most of these requirements are written in such a way as to suggest humans should be performing audits quarterly, but in my experience so far, they can all be met programmatically and still make the auditors happy—well, not happy, but satisfied.

Account Auditing

The security standards that I'm aware of all have in common some user-account-related auditing requirements. Among these are requirements for detecting old accounts and enforcing password strength policies. How you do this depends a lot on your environment and the size of your organization.

Most companies of any size use some sort of LDAP-like directory system for maintaining user IDs and passwords. The general idea here is to keep a list of valid accounts. If you're large, you probably also have something such as PeopleSoft or SAP. In this case you're looking to write an LDAP diff between your HR system and your directory system. Accounts that don't exist in the former probably shouldn't exist in the latter, with the exception of system

and administrator-related accounts. You'll want to enumerate the exceptions once and monitor changes from then on. Any monitoring system that can execute arbitrary scripts can fulfill this role.

If you're small like us, then you may be using local credentials. In this case a simple list of valid accounts to diff against is probably sufficient. Either way the strategy is pretty much the same. A Nagios plug-in could be used to check accounts on the servers against a list of valid accounts. A file system integrity checker such as Samhain [1] or a configuration management engine could be useful for larger organizations. The auditors are going to want to see how the valid account list is managed and whatever policies you have for incident response in the event a rogue account is encountered or an unexpected change is made.

Change Auditing

You will be required to display policies and procedures for making changes to production systems. These need to be backed up with a change-detection methodology of some type. For large installs configuration management engines are certainly the best way to go here. Unfortunately, I haven't taken that particular plunge myself as of yet, so I can't speak intelligently about it beyond observing that if LISA attendees are any indication, the "big three" appear to be (in alphabetical order) bcfg2, cfengine, and puppet.

If you're like me (writing articles telling other people what to do when you should be implementing configuration management), then you can meet the criteria with a filesystem integrity checker such as Samhain [1]. You'll also need something like RANCID [2] for your network gear, even if you do use a configuration management engine. For those of you who aren't familiar with it, RANCID is an ingenious bit of glue among CVS, expect, and SSH. It logs into your network gear, dumps the configuration, and maintains a revision history of the dumps with CVS for you automatically. You'll probably want to send change notifications from these systems to /bin/logger so they get sent to your centralized syslog server, which brings me to . . .

Incident Detection

The PCI DSS wants you to have a policy for collecting and auditing logs from systems and security appliances. It's not enough to have the logs; you need to show how you audit them and what you do when your log audits encounter something unexpected and possibly bad. You can meet these criteria with automated log parsing or event correlation software if you implement it well enough.

The first thing you're going to need to do is to get your security-related logs in a single place and in a single format. The auditors know that the log-watching software is only as good as the logs it has to watch, so they're going to want to see what information you have, how you're getting it there, and what's preventing it from being modified by a malicious entity. Centralized syslog infrastructure works well for this for most organizations. I'd suggest a syslog daemon that supports the newer (RFC 3195) reliable delivery mechanisms such as SDSC Syslog [3] if you're going to go the syslog route.

There are several implementations of the syslog protocol for Windows, including EventReporter [4] and Snare [5], if you're blessed with Windows machines. I can't think of any network appliances that don't have native support for syslog, but if you've managed to find one, there are several SNMP to syslog translators [6] available. Creating a centralized syslog architecture is

beyond the scope of this article, but a great place to start is Tina Bird's excellent log analysis portal [7].

Once you have it all in one spot there are several ways to parse it. Most people I know are fond of either logsurfer [8] or SEC [9] for this purpose. Both of these tools are fodder for future articles, and both do an excellent job of mining log data in real time for interesting events, and both will meet the audit criteria. I tend to mostly use logsurfer because it's written in C and therefore has a smaller footprint, saving SEC for one-off or more complex situations. The auditors will want to see what you're parsing for and what you're doing with the alerts. (Hint: This should be detailed in an incident response policy, and the policy should actually be followed.)

Another tool that's gained much popularity in the past few years is Splunk [10]. Having played with the open source version, I'm convinced this isn't wholly to do with the company's cool t-shirts. Splunk is, in fact, a fantastic tool. It won't be replacing grep, logsurfer, or SEC in my environment, but it certainly augments them, and since Splunk added support for taking arbitrary actions based on regex-style criteria, it certainly bears mentioning in this context. Meeting compliance requirements is actually a stated goal of the software, according to Splunk's Web site, so it seems the company had some experience in this regard.

For bonus points, consider implementing a nonaddressable monitoring system for forensics purposes. Logs cannot be modified by a malicious entity if they are on a box that cannot be accessed via the network (probably). Passive network taps work well here. I've personally had good experiences with NetOptics [11] aggregating network taps. One type of tap we use can listen to four 10/100 networks and aggregate them all to a single gigabit port. The box connected to that port need not actually be on any network at all.

System and Service Discovery

The auditors will want to know how you're managing network access, both via WiFi and via rj45 wall sockets. Even if you use a NAC system, they'll want you to audit the network for unauthorized systems as well as new services running on existing systems. Of course, whatever you use needs to be backed up with a policy document. PCI wants a firewall policy, for example, which details the services that are authorized for use on the network and written justification for services it considers insecure.

There are several Nagios plug-ins that can help here. Many of them are wrappers around nmap and some do their own scanning and system discovery. I've recently started using OSSIM [12] for this sort of stuff and I have to say I'm pretty happy with it so far. OSSIM is like a portal for all things security in your organization. It's one of the better tools I've come across for pooling output from existing tools and presenting it in a way that is actually pretty flexible. The best thing about it is that it seems to have silenced the auditors with its built-in support for scanning tools such as nmap, p0f, Pads, and Nessus.

Having spent the better part of two months in PCI-land, what have I implemented to make my workplace more secure? Well, nothing, actually. All of the systems security and monitoring currently in place was in place pre-audit, and I've spent 100% of my time churning out policies outlawing bit-torrent in the requisite Microsoft Word format. If we are more secure as a result of that then I'm glad, but I kind of doubt that we are. Bruce himself once said, "Amateurs hack systems; professionals hack people," and I'm beginning to think I've been hacked by a gaggle of standards bodies. I'm not

sure what the answer to the problem of organizational security is, but this can't be it. What we have in the standards is nothing but a license for vendors to take our money and give us in return someone else's static definition of an utterly subjective concept.

I don't harbor any hope that ranting about it here has convinced anyone of anything, but I do hope you found a few tools in there that will help you with your particular set of auditors. If not, at least you know I feel your pain.

Take it easy.

REFERENCES

- [1] <http://www.la-samhna.de/samhain>.
- [2] <http://www.shrubbery.net/rancid>.
- [3] <http://sourceforge.net/projects/sdscsyslog>.
- [4] <http://www.eventreporter.com>.
- [5] <http://www.intersectalliance.com>.
- [6] <http://snmptt.sourceforge.net>.
- [7] <http://www.loganalysis.org>.
- [8] <http://www.dfn-cert.de/eng/logsurf>.
- [9] <http://simple-evcorr.sourceforge.net>.
- [10] <http://www.splunk.com>.
- [11] <http://www.netoptics.com>.
- [12] <http://www.ossim.net>.

HEISON CHAK

VoIP and IPv6



Heison Chak is a system and network administrator at SOMA Networks. He focuses on network management and performance analysis of data and voice networks. Heison has been an active member of the Asterisk community since 2003.

heison@chak.ca

IP ADDRESS SHORTAGE, TRAFFIC PRIORITIZATION, end-to-end security, and NAT issues are problems with VoIP that can be addressed by IPv6. This article will discuss some of these issues. It will also outline some of the hurdles in migrating to the next version of the Internet Protocol.

IPv4 Exhaustion

At a consumption of 5–8% per year, it is predicted that the remaining 25% of available IPv4 addresses could be exhausted as early as July 15, 2011. With Japan having made IPv6 adoption a mandate since 2001, Asia sits in a leading spot in the push for the new protocol. IPv4 addresses are 32-bit, normally written as four decimal numbers.

Example: 192.168.1.10

IPv6 addresses are 128-bit, represented as eight fields, separated by colons, of up to four hexadecimal digits each.

Example: 3ffe:ffff:101::230:6eff:fe04:d9ff

The symbol :: is a special syntax used to represent multiple 16-bit groups of continuous zeroes. The large number of addresses (2^{128}) allows a hierarchical allocation of addresses that may make routing and renumbering simpler. Separate address spaces exist for ISPs and for hosts, which is inefficient in use of address space bits but efficient for operational needs.

Third-generation (3G) wireless both in Europe and North America had once been viewed as a big push toward IPv6, since the protocol can facilitate more IP addresses, end-to-end QoS/security, and mobility between 3G and other networks. However, with the slow adoption of 3G networks, ISPs found that they didn't need as many IP addresses as they had once thought. With the exhaustion date closing in, these perceptions may change rapidly in the next few years. The United States government is mandating its agencies' networks to interface with new IPv6 backbones by June 2008, and China plans to showcase its largest IPv6 network at the 2008 Olympics. Commonly known as 6CDO, the IPv6 EU-Chinese Digital Olympics project will demonstrate IPv6 applications in many facets at the Summer Games. This will certainly be an exciting year for IPv6.

End-to-End Communications

With IPv4, NAT is often used to enable multiple hosts on a private network to access the Internet using a single public IP address. Many find this Layer 3 technique convenient and use it widely. Some higher-layer protocols, such as SIP, send network-layer address information inside application payloads. For example, embedded private IP addresses can often be seen in SDP (Session Description Protocol) embedded as a SIP payload. NAT operates only in Layer 3, so the embedded private IP address will not be translated, because it is in Layer 4. Because the private address often become unreachable from the receiving end, the effect could be SIP calls that fail to establish, failed touch-tone inputs, one-way audio, or simply no audio.

Instead of fixing the problem from the root, that is, by not sending embedded Layer 3 addresses in a non-Layer 3 protocol, workarounds are invented to change the embedded private IP address to match the public Internet address on the router. On the endpoint equipment, it may support a static entry of the border router's external IP address or one of the automatic discovery protocols: STUN (Simple Traversal of UDP through NATs), ICE (Interactive Connectivity Establishment), or Traversal Using Relay NAT (TURN). On the router, a SIP-capable ALG (Application Layer Gateway) may be running to examine each SIP/SDP packet and alter the embedded IP.

Although these techniques are widely used to assist devices behind a NAT firewall or router with their packet routing, such altering is actually one of the biggest offenders in data integrity. If you want to implement end-to-end security with IPsec, using one of these techniques will be a challenge, because an ALG on the router altering packets will cause IPsec Authentication Signatures to fail.

With IPv6, end-to-end communication permits nodes to communicate without NAT in a secure fashion. In addition, quality of service can be maintained between IPv4 and IPv6, since there is no difference in QoS for the two protocol versions. There is only a slightly different header definition in IPv6.

Migrating VoIP to IPv6

Unlike the migration from NCP to IPv4 in the early 1980s, IPv4 and IPv6 will interoperate during and after the transition. With the new API (RFC3493, RFC3542) having been available since Linux 2.4, FreeBSD 4.x, Mac OS X 10.2, Windows XP, and Solaris 8, OSes can leave the details of supporting the two versions to the API. Many network vendors (e.g., Cisco, Juniper, Checkpoint) and open source applications (e.g., Apache, Sendmail, Postfix, OpenSSH) also feature IPv6 support.

In order for VoIP to take advantage of IPv6, any VoIP equipment that may be connecting to a network should be made IPv6-aware. Application servers, gateways, and communication end nodes must incorporate the new API. They need to be able to handle both IPv4 and IPv6 traffic, understand how to parse IPv6 URLs, and be able to store the lengthier IPv6 addresses. It's best if they are version-independent whenever possible, parsing addresses and URLs to support both the IPv4 and the IPv6 address syntax required for networking, logging, and SIP URL parsing.

IPv6 address syntax:
0::C0A8:010A # IPv4-compatible address (192.168.1.10)
1:2:3:4:5:6:7:8/16 # denotes the address to be /16
0:0:0:0:0:0:0:1 # loopback
::1 # loopback (shorthand)
http://[1:2:3:4:5:6:7:8]:80/index.html # port 80 URL

Open Source VoIP and IPv6

In March 2007, Viagenie in Canada conducted a VoIP call to Consulintel in Spain using CounterPath eyeBeam (previously known as X-Lite) through Asterisk with the IPv6 patch. "Asterisk-IPv6 shows the power of VoIPv6 by avoiding all issues regarding NAT traversal when using IPv4. The presence of NAT for VoIPv4 results in users issues such as non-connecting calls, one-way audio, non-working DTMF. Asterisk-IPv6 solves all these issues and also brings, together with IPv6, true IP mobility, security and autoconfiguration of VoIPv6 phones," states Marc Blanchet, president of Viagenie. Despite efforts made by Viagenie to make Asterisk IPv6-aware when using the SIP protocol, however, the current version of Asterisk is still not IPv6-ready. SER (SIP Express Router) does seem to be further ahead when it comes to IPv6 support.

ROBERT G. FERRELL

/dev/random



Robert G. Ferrell is an information security geek biding his time until that genius grant finally comes through.

rgferrell@gmail.com

I THINK THE CHIEF PROBLEM WITH IPV6, other than the widespread allergy people seem to have toward adopting it, is that the overall implementation just isn't ambitious enough. There's no reason to stop with simply assigning unique IP addresses to all the network-capable machines on the planet—with 128 bits of address space we could enzymatically splice IP headers into every gene of every human DNA molecule in existence. Never again will you be bored in a hotel: With your laptop and wireless bio-interface you can experience endless hours of entertainment, changing your eye color from brown to blue at will, tweaking your vocal chords to sound like Cyndi Lauper inhaling helium, or growing hobbit-hair on your feet. Forget dieting and exercise—lose weight the easy way by cranking that metabolism off the chart and then watching the pounds just melt right away. Be sure to call down to the front desk first for a plastic tarp to make cleanup easier.

Combined with the latest in GPS technology, genetic IP labeling could revolutionize the dating scene. Punch in the attributes you'd like in your ideal mate, overlay the resulting genome map on a street map, and start scanning. You could use the cell phone network to pinpoint your best-fit candidates in a matter of minutes, ranked according to degree of genetic compatibility. The commercial spin-off possibilities are myriad: *Google Birth* if you're looking to reproduce; *Google Mirth* if you want someone with a sense of humor; *Google Worth* if you crave a well-heeled mate. Search Engines might have to be expanded to Search and Rescue Engines.

I see, in my dilithium crystal ball, political parties, religious cults, or professional/trade associations based on genomic traits. Advertising will get even more precisely targeted when some hapless schmuck strolls up to the mall directory and the genome scanalyzer hidden deep inside launches into a spiel tailored to his personal physical shortcomings. Randomly shotgunned ads for Viagra and Cialis will seem quaint and harmless compared with the ruthless efficiency of a machine that knows not only what's wrong with you right now but what might be crouched around the corner patiently waiting to spring in a decade or three.

There will be a whole new exciting avenue for diversion opening up when the horrors people can look forward to in later life as a result of their chromosomal baggage begin to pop up in unexpected places and say "boo" at

them. Consider those little self-diagnosis chairs at the neighborhood pharmacy, for example. Check your blood pressure, measure your heart rate, and plan for genetic doom while-u-wait, splashed for your convenience on the wide-screen, high-definition plasma suspended over the prophylactic aisle and brought to you by the makers of those little blue capsules you take to stay regular. This innovation in pharmaceutical marketing would represent the final annihilation of what few shreds of personal privacy still remain, but the way we're going that's inevitable no matter what technologies we decide to embrace.

Life insurance companies will, of course, adore this new development. They can scan you as you walk in the door and have a rate quote (or more likely a declination of coverage) ready by the time you reach the agent's desk. Those people who do manage to meet the minimum standards for genetic soundness can expect to be chased down in the street by policy pushers eager to sign them up. Employers and the military will be able to customize the health insurance packages of recruits to exclude any genetic surprises that might develop into expensive but now inarguably preexisting conditions at some future stage. It could be argued, in fact, that virtually all human maladies are the result of genetics in one form or another. No matter where you go, where are you?

We might want to put in a little overtime in the security arena if we decide to head down this path, however. I'm thinking that it wouldn't be too pleasant if black hats figured out a way to crack the Direct On-Demand Genetic Expression Encryption algorithm. I can envision all sorts of possible attacks: mRNA-in-the-middle, cross-linkage scripting, STS-injection, denial of sequence, you name it. If you're handy with Genetisploit, you might even be able to make everyone in the hotel WiFi cloud sprout tiny horns from their foreheads overnight: 802.666.

Not that all who hack thusly will be content with mere whimsy, alas. We'll need more robust authentication to deal with these rascals, methinks: Maybe upgrade the current multifactor paradigm to *something you are, something you have, something you know, and something you secrete*. That may not halt potential attackers totally, but they'll at least have to slow down long enough to towel off.

If only we could identify the "terrorism" gene and suppress it universally, this whole "obsoleting the concept of privacy and ignoring the Constitution" fad might ooze back into the slime-saturated crack of damnation from whence it slithered. My innate human talent for pattern recognition suggests that we'd just find some other justification for carrying the process to its logical conclusion, though. Once set into motion, bloated bureaucracies (if I may be excused the redundancy) take "juggernaut" to a whole new level. Inertia, thy name is government.

I fully expect our genes will one day just cut us out of the loop entirely and communicate directly with other genes. That's pretty much where evolution seems to be headed, anyway, as we've never been particularly reliable or efficient as recombination vectors.

Homo sapiens: the pinnacle of terrestrial evolution, or merely a deep gouge in the fossil record? You decide.

book reviews



ELIZABETH ZWICKY, WITH TONY DEL PORTO, NICK STOUGHTON, AND SAM STOVER

SECURITY DATA VISUALIZATION

Greg Conti

No Starch Press, 2007. 230 pages.

ISBN 978-1-59327-143-5

VISUALIZING DATA

Ben Fry

O'Reilly, 2007. 382 pages.

ISBN 978-0-59651-455-6

If you are a fan of visualization, or perhaps you're looking for a new hobby and have some numbers you're interested in, buy both these books. If you already have the numbers ready to whip into shape and you have tools you love to do it with, you might skip *Visualizing Data*. If you have no interest in security or networks, or you're just starting from scratch, you could skip *Security Data Visualization*. But probably, you want both of them. You want *Security Data Visualization* if you are drowning in network-oriented data, even if you don't want to be a fan of visualization.

Visualizing Data is a tour through the process of taking a question, finding the numbers that go with it, and producing an interactive visualization of the answer that you can put on a Web page easily. It uses Processing, which is my current favorite play language, but it also talks in passing about other tools and languages. The techniques are applicable to any language, although if you haven't got a language you're really fluent in for this purpose, I'd recommend going ahead and learning Processing. It's easy to learn, and the ability to publish to the Web without hassle is priceless.

My favorite thing about *Visualizing Data* is that it tackles the whole process in all its blood, guts, and gore. It

starts with finding the data and cleaning it up. Many books assume that the data fairy is going to come bring you data, and that it will either be clean, lovely data or you will parse it carefully into clean, lovely data. This book assumes that a significant portion of the data you care about comes from some scuzzy Web page you don't control and that you are going to use exactly the minimum required finesse to tear out the parts you care about. It talks about how to do this, and how to decide what the minimum required finesse would be. (Do you do it by hand? Use a regular expression? Actually bother to parse XML?)

Visualizing Data also shows a couple of cool visualization techniques, but it is primarily a process book; it's designed to take people with a casual interest in visualizing data and walk them through the whole process, end-to-end, from finding the data to ending up with a refined, interactive way of looking at it. It also teaches Processing, on the side.

Security Data Visualization has some discussion of the process, but it's mostly a catalog of ways you haven't thought of to look at network and security data. It's most interesting to people who care about network data (even if they don't care about security), but if you're into data visualization, there's a lot there even if networks and security aren't your area. These are interesting data sets in a lot of ways. Obviously, anything that involves people trying to break into computers has inherent dramatic value, but network security data sets are interesting as a problem, as well. They're big and complicated, and people are intentionally hiding stuff in them. Techniques that work here can be adapted to many other large, complicated, multivariate data sets.

If you do work with security data, particularly network data, there's an in-depth explanation of how to use visualization tools to illuminate characteristics of your data, along with a really great references section to point you to more information. It's not a cookbook, but it's not an area where the recipes have been found yet, either.

HANDBOOK OF NETWORK AND SYSTEM ADMINISTRATION

Jan Bergstra and Mark Burgess, editors

Elsevier, 2007. 997 pages.

ISBN 978-0-444-52198-9

The preface to this book says it is written by researchers, for researchers, educators, and advanced practitioners. In this case, you should take this very seriously. It is intended for academic researchers first and foremost, with educators as a strong secondary audience, it

and a hope that a few practitioners (people you might consider nonacademic researchers, for instance) might have some interest. It's the sort of book where eigenvectors need no explanation.

That makes it hard to evaluate. When I review, I hold in my head a picture of the sort of people I'm reviewing for. For most of those people, for people who do programming and system administration for a living, this book is not of interest. If you spend most of your time administering computer systems, as opposed to doing research and writing papers, what you need to know here is that the field has reached a point where a major academic publisher will publish a serious academic book, and it looks all impressive and stuff. You don't want to own it, unless you want to bludgeon people with it, metaphorically or literally. Since it weighs probably twice what my laptop does, it makes an excellent weapon, and if you'd like people to believe in the field's seriousness, the sheer density of the text ought to convince them.

Furthermore, this kind of book is not going to be welcoming and engaging to anybody. That's not the reason for its existence, and that's not the style academic researchers call for. You can't fault it for that. You also can't fault an anthology for being uneven, and although you can fault some of the individual chapters for being not particularly well focused, you can't be surprised. If you tell somebody to write down all the interesting theoretical stuff you might want to know about system administration and topic X, most of the time you are going to get something that doesn't have a clear focus or audience. The best authors in *Handbook of Network and System Administration* transcend this, but it's difficult to do.

So I am left with a book that I think is excellent in parts and that I think is a fine example of its type, but that I mostly didn't enjoy and that most readers of this review are not going to want to own. I did enjoy several chapters, including, surprisingly, the one on graph theory and, less surprisingly, Matt Blaze's chapter on security models, and I found several others to be of theoretical interest, not to mention the inherent amusement value in statements such as "The belief that one must control specific files seems to arise from a lack of trust in the software base being managed." There is a reason I do not trust the software base I manage. It is the same reason that psychiatric nurses do not trust patients to behave rationally. I have both practical and theoretical cause to believe that software is in no way trustworthy.

If, by happenstance, you are interested in this book and the data visualization books, read this one first,

or let a long time elapse between them. Otherwise, the contrast between the beautiful data visualizations and the charts and graphs in this book will drive you mad. They are not, objectively, horrible charts and graphs. But they are for the most part to proper data visualization as Little League baseball is to the World Series.

LINUX FIREWALLS: ATTACK DETECTION AND RESPONSE WITH IPTABLES, PSAD, AND FWSNORT

Michael Rash

No Starch Press, 2007. 290 pages.

ISBN 978-1-59327-141-1

Here's another stark contrast. This is the exact opposite of a theoretical book. It does mention the relevant concepts, but it's primarily a step-by-step tour of iptables, psad, and fwsnort. If you're building a Linux firewall and want to know what all the bells and whistles are, when you might want to set them off, and how to hook them together, here you go.

[Editor's Note: I also read this book. Mike has written several articles for ;login: about topics he covers in considerable detail in this book: psad, fwknop, and fwsnort. Running on Linux, psad monitors iptables logfiles and can send email alerts and even add in reactive filters in response to network events. Mike wrote psad as a replacement for the aging portsentry Perl script, used to detect scans. But psad does this and much more. With fwknop you can open ports in your firewall in response to a cryptographic request sent to the firewall. The most ambitious tool of all is fwsnort; it turns a Linux iptables ruleset into an IDS using many Snort rules.

Linux Firewalls is more a book about using these tools than an iptables primer. But if you ever wanted to learn more about iptables' less familiar features, such as pattern matching within application data or logging minutia, this is the book for you.]

THE BOOK OF PF: A NO-NONSENSE GUIDE TO THE OPENBSD FIREWALL

Peter N.M. Hansteen

No Starch Press, 2008. 158 pages.

ISBN 978-1-59327-165-7

REVIEWED BY TONY DEL PORTO

PF is the OpenBSD packet filter. I've used PF for a variety of things since its release with OpenBSD 3.0 and like it for its intuitive syntax and comprehensive feature set. Although the system manual and FAQ are complete in describing PF's features, figuring out

how to translate those features into a functioning set of rules has required frequent trips to my favorite search engine. *The Book of PF* aims to provide an overview of the things that can be done with PF, with enough examples to get the reader started. It is explicitly not a how-to book or cookbook, but, rather, a less terse (and more enjoyable to read) explanation of PF's features.

After a bit of overview the author covers PF rule syntax in two short chapters and then goes on to describe some of the more interesting features of the current version of PF. In addition to the standard things a stateful packet filter is expected to do, PF has been adapted to deal with a variety of issues that plague networked systems. PF provides greylisting and tarpitting to deal with spam, packet queueing to deal with oversubscribed bandwidth, and failover via CARP. There is an authentication scheme via ssh for network access, NAT and port redirection, proxies for ftp, tftp, and even the initial connection between hosts via the synproxy facility. PF is available on the various *BSDs and the author notes the differences in feature implementation where appropriate. The author ends the book with monitoring and debugging tools and some suggestions for choosing suitable hardware. He also touches briefly on performance, which has been the subject of a previous *;login:* article ("Linux vs. OpenBSD: A Firewall Performance Test," by Adamo and Tablo, December 2005).

On the whole the book is a great resource and has me eager to rewrite my aging rulesets to take advantage of PF's more recent features. In particular spamd (not spamassassin) has my attention. I will pick a few nits though. I'm a little perplexed by the author's decision to place the chapter on wireless networks early in the book and between chapters on more commonly used features, as well as why the debugging section was presented so late. One of the most frequent challenges I've faced is the question, "Why doesn't this network service work?" The debugging section is key in providing tools to answer that question, at least as far as PF is concerned. I would have preferred the organization to be more bread-and-butter up front and icing later on.

For future editions, I would love to see an appendix with some real-world rulesets demonstrating how the features of PF can be combined to express a network policy. The author provides a copious list of both online and dead-tree resources for finding such examples; however, a book that aims to be "a stand-alone document to enable you to work on your machines with only short forays into man pages and occasional reference to the online and printed resources" should

include a few multipage rulesets, especially if the book is to be consulted when a ruleset is broken and the interwebs are not reachable for consultation. At 145 pages, the book has room to grow, and I look forward to reading the next edition.

CROSS-PLATFORM DEVELOPMENT IN C++: BUILDING MAC OS X, LINUX, AND WINDOWS APPLICATIONS

Syd Logan

Addison Wesley, 2007. 576 pages.

ISBN: 978-0-321-51437-0

REVIEWED BY NICK STOUGHTON

If I were ever to write a book, there is a very good chance that I would produce a volume that looks a lot like Syd Logan's *Cross-Platform Development in C++*. There is a fundamental principle that my good friend Stephen Walli puts as, "There's no such thing as a portable application, merely applications that have been ported." Syd starts out in the introduction with the point that even a simple "hello, world" application produces different output on Mac OS X and a Windows platform (that pesky line-ending difference from DOS).

The book goes on to describe a set of "Items"—maxims to help improve code portability. Each item is well expounded, and the text is littered with examples. If it were not for the examples, the book need not have had the "in C++" as part of its title. Indeed, the C++ used is sufficiently close to C in almost every case that I would certainly recommend this book to C developers, and quite possibly to developers who work in any similar computer language.

Syd is not actively involved in the same standards committees as I am, and so he leaves "Use Standards based APIs" to item 16, whereas I would have had this as item 1, or at worst 2. He also isn't as up-to-date with them as he might be. For example, he states that the "GNU C library implements all the functions specified in ISO/IEC 9945-1:1996"; although this is true, it also implements all the functions in the 2001 and 2008 editions, and many of the new functions in the 2008 edition have come from the GNU C library. He uses the old `_POSIX_SOURCE` feature test macro, which was superseded by the `POSIX_C_SOURCE` macro in 1996. But the correct points are made, and the overall advice is sound.

Syd also recommends using a platform abstraction library; his background is from Netscape, and not surprisingly he pushes the Netscape Portable Runtime Library (NSPR). There are other such libraries, most notably Boost, and although Boost rates a mention, the relative strengths and weaknesses of the differing approaches are not discussed.

All in all, this is a highly recommended book for anyone involved in software development. Do not be put off by the C++ in the title. Do not expect it to answer every question you've ever had. But do expect to be provoked into writing better, more portable code.

METASPLOIT TOOLKIT FOR PENETRATION TESTING, EXPLOIT DEVELOPMENT, AND VULNERABILITY RESEARCH

David Maynor, K.K. Mookhey, Jacopo Cervini, Fairuzan Roslan, and Kevin Beaver

Syngress, 2007. 352 pages.

ISBN: 978-1-59749-074-0

REVIEWED BY SAM STOVER

Metasploit Toolkit is a funny book—not “ha ha” funny, but oddly put together. Chapter 1 is 63 pages long, Chapter 2 is 11 pages, and Chapter 3 is a mere 6. See what I mean? Funny. OK, now that I've covered the only negative thing I can come up with, let's get to the important bits. To put it as concisely as I possibly can, if you are in any way interested in a book on Metasploit, this is the one to get. That's it: Stop looking around and just find this book. Don't even bother browsing through it at the bookstore; that would be a waste of time. If I could figure out a way to mainline text, I'd recommend that.

So, why do I like this book so much? Well, for starters, it is one of the first books by Syngress that didn't overwhelm me with spelling and grammar mistakes. This could be due to one of two reasons: Either there weren't any, or I was so engrossed in the material presented by the book that I didn't notice them. I'm betting on the former, but YMMV. However, you say there are other Metasploit books out there. This is true, and I've even reviewed some of them, but the ones I've seen don't deal with Metasploit v3.x. Ah, but you counter that the rewrite in Ruby for v3.x didn't

affect the user experience, since it was all under the hood. Wrong and *wrong*. I've been using Metasploit pretty much since it came out, but v3.x finally makes it easy to discover your own vulnerabilities, and this book shows you how. This is a huge step for Metasploit, one that truly allows it to compete with the likes of CANVAS and IMPACT. (What's with the all caps? Are they yelling at us? Should Metasploit be METASPLOIT? Someone tell HD he needs to change the name.)

Setting aside the odd chapter structure, you'll find a large portion of this book (over 100 pages' worth) dedicated to case studies. I've said it before, and I'll say it again: I love case studies. They let me set things up, run through the process, and learn from it. Any book that keeps my fingers on the keyboard as much as it keeps my eyes on the page is a keeper. Following the case studies are three glossaries. If you are a veteran user, there's plenty in the first couple of chapters that you can ignore, such as how to install and use Metasploit, but there's also a fair bit of detail on how the 3.x changes will impact your experience. If you are a Metasploit noob, reading the book from cover to cover will effectively migrate you out of noobdom. Each Metasploit component is addressed in sufficient detail to give either the new or the experienced user what they need—the noob learns how things are, and the vet learns how things have changed.

Overall, Metasploit 3.x is a pretty exciting advance in the exploit/vuln landscape. This book is the perfect guide for getting started and/or learning what is new. I honestly can't recommend it enough: it was written precisely the way I like books to be written: clear, concise, and with lots of examples. If you want to learn anything about Metasploit, I can't think of a better place to start.

USENIX notes

USENIX MEMBER BENEFITS

Members of the USENIX Association receive the following benefits:

FREE SUBSCRIPTION to *;login:*, the Association's magazine, published six times a year, featuring technical articles, system administration articles, tips and techniques, practical columns on such topics as security, Perl, networks, and operating systems, book reviews, and summaries of sessions at USENIX conferences.

ACCESS TO ;LOGIN: online from October 1997 to this month:
www.usenix.org/publications/login/

DISCOUNTS on registration fees for all USENIX conferences.

DISCOUNTS on the purchase of proceedings and CD-ROMs from USENIX conferences.

SPECIAL DISCOUNTS on a variety of products, books, software, and periodicals: www.usenix.org/membership/specialdisc.html.

THE RIGHT TO VOTE on matters affecting the Association, its bylaws, and election of its directors and officers.

FOR MORE INFORMATION regarding membership or benefits, please see www.usenix.org/membership/ or contact office@usenix.org.
Phone: 510-528-8649

USENIX BOARD OF DIRECTORS

Communicate directly with the USENIX Board of Directors by writing to board@usenix.org.

PRESIDENT

Michael B. Jones,
mike@usenix.org

VICE PRESIDENT

Clem Cole,
clem@usenix.org

SECRETARY

Alva Couch,
alva@usenix.org

TREASURER

Theodore Ts'o,
ted@usenix.org

DIRECTORS

Matt Blaze,
matt@usenix.org

Rémy Evard,
remy@usenix.org

Niels Provos,
niels@usenix.org

Margo Seltzer,
margo@usenix.org

OPEN PUBLIC ACCESS TO ALL USENIX CONFERENCE PROCEEDINGS

On March 12, 2008, the USENIX Association Board and staff announced publicly that access to all of our conference proceedings would now be available free of charge to the general public. In making this move we hope to set the standard for open access to information, an essential part of our mission.

This significant decision will allow universal access to some of the most important technical research in advanced computing. As members may know, USENIX has delivered innumerable industry "firsts" at past conferences, including ONYX, the first attempt at UNIX hardware; the launch of the first UNIX product by Digital Equipment Corporation; the first paper on Sendmail by Eric Allman; the first Perl presentation by Tom Christiansen; and the first report by James Gosling on Oak, which later became Java.

We are well aware that we could not achieve such goals without the support and dedication of our membership. We urge you to encourage others to join USENIX to support initiatives such as this one—and, of course, to receive *;login:*, along with other member benefits.

Thanks to USENIX and SAGE Corporate Supporters

USENIX Patrons

Google
Microsoft Research
NetApp

USENIX Benefactors

Hewlett-Packard
Linux Pro Magazine

USENIX & SAGE Partners

Ajava Systems, Inc.
DigiCert® SSL Certification
FOTO SEARCH Stock Footage
and Stock Photography

Raytheon
rTIN Aps
Splunk
Taos
Tellme Networks
Zenoss

USENIX Partners

Cambridge Computer
Services, Inc.
cPacket Networks
EAGLE Software, Inc.
GroundWork Open
Source Solutions
Hyperic

IBM
Infosys
Intel
Interhack
Oracle
Ripe NCC
Sendmail, Inc.
Sun Microsystems, Inc.
UUNET Technologies, Inc.
VMware

SAGE Partners

MSB Associates

PROFESSORS, CAMPUS STAFF, AND STUDENTS—

DO YOU HAVE A USENIX REPRESENTATIVE ON YOUR CAMPUS?

IF NOT, USENIX IS INTERESTED IN HAVING ONE!

The USENIX Campus Rep Program is a network of representatives at campuses around the world who provide Association information to students, and encourage student involvement in USENIX. This is a volunteer program, for which USENIX is always looking for academics to participate. The program is designed for faculty who directly interact with students. We fund one representative from a campus at a time. In return for service as a campus representative, we offer a complimentary membership and other benefits.

A campus rep's responsibilities include:

- Maintaining a library (online and in print) of USENIX publications at your university for student use
- Distributing calls for papers and upcoming event brochures, and re-distributing informational emails from USENIX
- Encouraging students to apply for travel grants to conferences
- Providing students who wish to join USENIX with information and applications
- Helping students to submit research papers to relevant USENIX conferences
- Providing USENIX with feedback and suggestions on how the organization can better serve students

In return for being our “eyes and ears” on campus, representatives receive a complimentary membership in USENIX with all membership benefits (except voting rights), and a free conference registration once a year (after one full year of service as a campus rep).

To qualify as a campus representative, you must:

- Be full-time faculty or staff at a four year accredited university
- Have been a dues-paying member of USENIX for at least one full year in the past

For more information about our Student Programs, see <http://www.usenix.org/students>

USENIX contact: Anne Dickison, Director of Marketing, anne@usenix.org

writing for ;login:

Writing is not easy for most of us. Having your writing rejected, for any reason, is no fun at all. The way to get your articles published in ;login:, with the least effort on your part and on the part of the staff of ;login:, is to submit a proposal first.

PROPOSALS

In the world of publishing, writing a proposal is nothing new. If you plan on writing a book, you need to write one chapter, a proposed table of contents, and the proposal itself and send the package to a book publisher. Writing the entire book first is asking for rejection, unless you are a well-known, popular writer.

;login: proposals are not like paper submission abstracts. We are not asking you to write a draft of the article as the proposal, but instead to describe the article you wish to write. There are some elements that you will want to include in any proposal:

- What's the topic of the article?
- What type of article is it (case study, tutorial, editorial, mini-paper, etc.)?
- Who is the intended audience (sysadmins, programmers, security wonks, network admins, etc.)?
- Why does this article need to be read?

- What, if any, non-text elements (illustrations, code, diagrams, etc.) will be included?
- What is the approximate length of the article?

Start out by answering each of those six questions. In answering the question about length, bear in mind that a page in ;login: is about 600 words. It is unusual for us to publish a one-page article or one over eight pages in length, but it can happen, and it will, if your article deserves it. We suggest, however, that you try to keep your article between two and five pages, as this matches the attention span of many people.

The answer to the question about why the article needs to be read is the place to wax enthusiastic. We do not want marketing, but your most eloquent explanation of why this article is important to the readership of ;login:, which is also the membership of USENIX.

UNACCEPTABLE ARTICLES

;login: will not publish certain articles. These include but are not limited to:

- Previously published articles. A piece that has appeared on your own Web server but not been posted to USENET or slashdot is not considered to have been published.
- Marketing pieces of any type. We don't accept articles about products. "Marketing" does not include being enthusiastic about a new tool or software that you can download for free, and you are encouraged to write case

studies of hardware or software that you helped install and configure, as long as you are not affiliated with or paid by the company you are writing about.

- Personal attacks

FORMAT

The initial reading of your article will be done by people using UNIX systems. Later phases involve Macs, but please send us text/plain formatted documents for the proposal. Send proposals to login@usenix.org.

DEADLINES

For our publishing deadlines, including the time you can expect to be asked to read proofs of your article, see the online schedule at <http://www.usenix.org/publications/login/sched.html>.

COPYRIGHT

You own the copyright to your work and grant USENIX permission to publish it in ;login: and on the Web. USENIX owns the copyright on the collection that is each issue of ;login:. You have control over who may reprint your text; financial negotiations are a private matter between you and any reprinter.

FOCUS ISSUES

In the past, there has been only one focus issue per year, the December Security edition. In the future, each issue may have one or more suggested focuses, tied either to events that will happen soon after ;login: has been delivered or events that are summarized in that edition.



Announcement and Call for Papers **USENIX**

8th USENIX Symposium on Operating Systems Design and Implementation (OSDI '08)

Sponsored by USENIX in cooperation with ACM SIGOPS

<http://www.usenix.org/osdi08>

December 8–10, 2008

San Diego, CA, USA

Important Dates

Paper submissions due: *May 8, 2008, 9:00 p.m. PDT*

Submissions acknowledged: *May 12, 2008*

Notification of acceptance: *July 29, 2008*

Papers due for shepherding: *Mid-September 2008*

Final papers due: *October 7, 2008*

Posters due: *October 15, 2008*

WiP reports due: *November 19, 2008*

Conference Organizers

Program Co-Chairs

Richard Draves, *Microsoft Research*

Robbert van Renesse, *Cornell University*

Program Committee

Marcos Aguilera, *Microsoft Research*

Lorenzo Alvisi, *University of Texas, Austin*

Remzi Arpaci-Dusseau, *University of Wisconsin, Madison*

Eric Brewer, *University of California, Berkeley, and Intel Research Berkeley*

Brad Chen, *Google, Inc.*

Fred Douglass, *IBM Research*

Greg Ganger, *Carnegie Mellon University*

Galen Hunt, *Microsoft Research*

Anthony Joseph, *University of California, Berkeley*

Dina Katabi, *MIT*

Kim Keeton, *HP Labs*

Idit Keidar, *Technion*

Terence Kelly, *HP Labs*

Dejan Kostić, *Ecole Polytechnique Fédérale de Lausanne*

Philip Levis, *Stanford University*

David Lie, *University of Toronto*

Jack Lo, *VMware*

Dahlia Malkhi, *Microsoft Research*

Erich Nahum, *IBM Research*

Fernando Pedone, *University of Lugano*

Ian Pratt, *University of Cambridge*

Dave Presotto, *Google, Inc.*

Krithi Ramamritham, *IIT Bombay*

Wolfgang Schröder-Preikschat, *University of Erlangen-Nürnberg*

Marvin Theimer, *Google, Inc.*

Leendert van Doorn, *AMD*

Geoffrey M. Voelker, *University of California, San Diego*

Jim Waldo, *Sun Microsystems, Inc.*

Helen Wang, *Microsoft Research*

Steering Committee

Brian Bershad, *University of Washington*

Jay Lepreau, *University of Utah*

Jeff Mogul, *HP Labs*

Margo Seltzer, *Harvard University*

Ellie Young, *USENIX*

Overview

The eighth OSDI seeks to present innovative, exciting research in computer systems. OSDI brings together professionals from academic and industrial backgrounds in what has become a premier forum for discussing the design, implementation, and implications of systems software.

The OSDI Symposium emphasizes innovative research as well as quantified or insightful experiences in systems design and implementation. OSDI takes a broad view of the systems area and solicits contributions from many fields of systems practice, including, but not limited to, operating systems, file and storage systems, distributed systems, mobile systems, secure systems, embedded systems, virtualization, networking as it relates to operating systems, and the interaction of hardware and software development. We particularly encourage contributions containing highly original ideas, new approaches, and/or groundbreaking results.

Submitting a Paper

Submissions will be judged on originality, significance, interest, clarity, relevance, and correctness. Accepted papers will be shepherded through an editorial review process by a member of the program committee.

A good paper will:

- ◆ consider a significant problem
- ◆ propose an interesting, compelling solution
- ◆ demonstrate the practicality and benefits of the solution
- ◆ draw appropriate conclusions

- ◆ clearly describe what the authors have done
- ◆ clearly articulate the advances beyond previous work

Papers accompanied by nondisclosure agreement forms will not be considered. All submissions will be treated as confidential prior to publication in the Proceedings.

In addition to citing relevant, published work, authors should relate their OSDI submissions to relevant submissions of their own that are simultaneously under review for other venues. The OSDI PC reserves the right to ask authors to provide copies of related simultaneously submitted papers.

Simultaneous submission of the same work to multiple venues, submission of previously published work, and plagiarism constitute dishonesty or fraud. USENIX, like other scientific and technical conferences and journals, prohibits these practices and may, on the recommendation of a program chair, take action against authors who have committed them. In some cases, program committees may share information about submitted papers with other conference chairs and journal editors to ensure the integrity of papers under consideration. If a violation of these principles is found, sanctions may include, but are not limited to, barring the authors from submitting to or participating in USENIX conferences for a set period, contacting the authors' institutions, and publicizing the details of the case.

Authors uncertain whether their submission meets USENIX's guidelines should contact the program chairs, osdi08chairs@usenix.org, or the USENIX office, submissionspolicy@usenix.org.

Authors of accepted papers will be expected to provide both PDF and HTML versions of their paper, for inclusion in the Web and CD-ROM versions of the Proceedings. Authors of accepted papers will also be expected to sign a Consent Form, agreeing not to publish their papers elsewhere within 12 months of acceptance, except for electronic access as permitted in the Consent Form. One author per paper will receive a registration discount of \$200. USENIX will offer a complimentary registration upon request.

Deadline and Submission Instructions

Authors are required to submit full papers by 9:00 p.m. PDT on May 8, 2008. This is a hard deadline—no extensions will be given.

Submitted papers must be no longer than 14 single-spaced 8.5" x 11" pages, including figures, tables, and references, using 10 point type on 12 point (single-spaced) leading, two-column format, Times Roman or a

similar font, within a text block 6.5" wide x 9" deep. Papers not meeting these criteria will be rejected without review, and no deadline extensions will be granted for reformatting. Pages should be numbered, and figures and tables should be legible in black and white, without requiring magnification. Papers so short as to be considered "extended abstracts" will not receive full consideration.

Papers must be in PDF format and must be submitted via the submission form on the OSDI '08 Call for Papers Web site, www.usenix.org/osdi08/cfp.

The title and author name(s) and affiliation(s) should appear on the first page of the submitted paper. (Reviewing is not blind.)

For more details on the submission process, and for templates to use with LaTeX, Word, etc., authors should consult the detailed submission requirements at <http://www.usenix.org/events/osdi08/cfp/requirements.html>.

All submissions will be acknowledged by May 12, 2008. If your submission is not acknowledged by this date, please contact the program chairs promptly at osdi08chairs@usenix.org.

Outstanding Paper Awards

The program committee will, at its discretion, give out awards for outstanding papers. Papers of particular merit will be forwarded to *ACM Transactions on Computer Systems* for possible publication in a special issue.

Work-in-Progress Reports

Are you doing new, interesting work that has not been previously presented and that is still in too early a phase for publication? The OSDI attendees could provide valuable feedback to you. We are particularly interested in the presentation of student work. Details on submitting Work-in-Progress session proposals will be available on the Web site by August 2008. The deadline is November 19, 2008.

Poster Session

We plan to hold a poster session in conjunction with a social event at the Symposium. Details on submitting posters for review will be available on the Web site by August 2008. The deadline is October 15, 2008.

Registration Materials

Complete program and registration information will be available in August 2008 on the conference Web site. If you would like to receive the latest USENIX conference information, please join our mailing list at <http://www.usenix.org/about/ mailing.html>.

2nd USENIX Workshop on Offensive Technologies (WOOT '08)

Sponsored by USENIX, the Advanced Computing Systems Association

<http://www.usenix.org/woot08>

July 28, 2008

San Jose, CA

WOOT '08 will be co-located with the 17th USENIX Security Symposium (USENIX Security '08), which will take place July 28–August 1, 2008.

Important Dates

Submissions due: *June 1, 2008, 11:59 p.m. PDT*

Notification of acceptance: *June 28, 2008*

Electronic files due: *July 14, 2008*

Workshop Organizers

Program Chairs

Dan Boneh, *Stanford University*

Tal Garfinkel, *VMware*

Dug Song, *Zattoo*

Program Committee

Pedram Amini, *Tipping Point*

Martin Casado, *Stanford University*

Chris Eagle, *Naval Postgraduate School*

Halvar Flake, *Zynamics*

Trent Jaeger, *Pennsylvania State University*

Nate Lawson, *Root Labs*

Charlie Miller, *Independent Security Evaluators*

Matt Miller, *Leviathan Security Group*

HD Moore, *BreakingPoint Systems*

Tim Newsham, *Information Security Partners, LLC*

Vern Paxson, *International Computer Science Institute
and Lawrence Berkeley National Laboratory*

Niels Provos, *Google*

Hovav Shacham, *University of California, San Diego*

Adam Shostack, *Microsoft*

Alex Sotirov, *VMware*

Giovanni Vigna, *University of California, Santa
Barbara*

Overview

Progress in the field of computer security is driven by a symbiotic relationship between our understandings of attack and of defense. The USENIX Workshop on Off-

sive Technologies aims to bring together researchers and practitioners in system security to present research advancing the understanding of attacks on operating systems, networks, and applications.

Instructions for Authors

Computer security is unique among systems disciplines in that practical details matter and concrete case studies keep the field grounded in practice. WOOT provides a forum for high-quality, peer-reviewed papers discussing tools and techniques for attack.

Submissions should reflect the state of the art in offensive computer security technology—either surveying previously poorly known areas or presenting entirely new attacks.

We are interested in work that could be presented at more traditional, academic security forums, as well as more applied work that informs the field about the state of security practice in offensive techniques.

A significant goal is producing published artifacts that will inform future work in the field. Submissions will be peer-reviewed and shepherded as appropriate.

Submission topics include:

- Vulnerability research (software auditing, reverse engineering)
- Penetration testing
- Exploit techniques and automation
- Network-based attacks (routing, DNS, IDS/IPS/firewall evasion)
- Reconnaissance (scanning, software, and hardware fingerprinting)
- Malware design and implementation (rootkits, viruses, bots, worms)
- Denial-of-service attacks
- Web and database security
- Weaknesses in deployed systems (VoIP, telephony, wireless, games)
- Practical cryptanalysis (hardware, DRM, etc.)

Workshop Format

The attendees will be authors of accepted position papers/presentations as well as invited guests. Each author will have 25 minutes to present his or her idea. A limited number of grants are available to assist presenters who might otherwise be unable to attend the workshop. Paper files will be available on the USENIX Web site to participants before the workshop and will be made generally accessible after the workshop.

Submission Instructions

Papers must be received by 11:59 p.m. Pacific time on Sunday, June 1, 2008. This is a hard deadline—no extensions will be given. Submissions should contain six or fewer two-column pages, excluding references, using 10 point fonts, standard spacing, and 1 inch margins. Please number the pages. All submissions will be electronic and must be in either PDF (preferred) or PostScript. Author names and affiliations should appear on the title page. Submit papers using the Web form on the WOOT '08 Call for Papers Web site, <http://www.usenix.org/woot08/cfp>.

Given the unique focus of this workshop, we expect that work that has been presented previously in an unpublished form (e.g., Black Hat presentations) but that is well suited to a more formal and complete treat-

ment in a published, peer-reviewed setting will be submitted to WOOT, and we encourage such submissions (with adequate citation of previous presentations).

Simultaneous submission of the same work to multiple venues, submission of previously published work, and plagiarism constitute dishonesty or fraud. USENIX, like other scientific and technical conferences and journals, prohibits these practices and may, on the recommendation of a program chair, take action against authors who have committed them. In some cases, program committees may share information about submitted papers with other conference chairs and journal editors to ensure the integrity of papers under consideration. If a violation of these principles is found, sanctions may include, but are not limited to, barring the authors from submitting to or participating in USENIX conferences for a set period, contacting the authors' institutions, and publicizing the details of the case.

Authors uncertain whether their submission meets USENIX's guidelines should contact the workshop organizers at woot08chairs@usenix.org or the USENIX office, submissionspolicy@usenix.org.

Papers accompanied by nondisclosure agreement forms will not be considered. All submissions will be treated as confidential prior to publication in the Proceedings.

3rd USENIX Workshop on Hot Topics in Security (HotSec '08)

Sponsored by USENIX, the Advanced Computing Systems Association

<http://www.usenix.org/hotsec08>

July 29, 2008

San Jose, CA

HotSec '08 will be co-located with the 17th USENIX Security Symposium (USENIX Security '08), July 28–August 1, 2008.

Important Dates

Position paper submissions due: *May 28, 2008, 11:59 p.m. PDT*

Notification of acceptance: *June 25, 2008*

Final files due: *July 14, 2008*

Workshop Organizers

Program Chair

Niels Provos, *Google*

Program Committee

Matt Blaze, *University of Pennsylvania*

Martin Casado, *Stanford University*

Wenke Lee, *Georgia Institute of Technology*

Patrick McDaniel, *Pennsylvania State University*

Michalis Polichronakis, *FORTH-ICS*

Moheeb Rajab, *Johns Hopkins University*

Tara Whalen, *Dalhousie University*

Overview

Position papers are solicited for the 3rd USENIX Workshop on Hot Topics in Security (HotSec '08).

HotSec is intended as a forum for lively discussion of aggressively innovative and potentially disruptive ideas in all aspects of systems security. Surprising results and thought-provoking ideas will be strongly favored; complete papers with polished results in well-explored research areas are discouraged. Papers will be selected for their potential to stimulate discussion in the workshop.

HotSec '08 will be a one-day event, Tuesday, July 29, 2008, co-located with the 17th USENIX Security Symposium in San Jose, California.

Workshop Format

Attendance will be by invitation only, limited to 35–50 participants, with preference given to the authors of accepted position papers/presentations.

Each author will have 10–15 minutes to present his or her idea, followed by 15–20 minutes of discussion with the workshop participants.

Instructions for Authors

The goal of the workshop is to stimulate discussion of and thinking about aggressive ideas and issues in systems security.

Position papers are expected to fit into one of the following categories:

- Fundamentally new techniques for and approaches to dealing with current security problems
- New major problems arising from new technologies that are now being developed or deployed
- Truly surprising results that cause rethinking of previous approaches

While our goal is to solicit ideas that are not completely worked out, we expect submissions to be supported by some evidence of feasibility or preliminary quantitative results.

Topics

Possible topics of interest include but are not limited to:

- Secure operation, management, and event response of/for ultra-large-scale systems
- Designing secure large-scale systems and networks
- Self-organizing and self-protecting systems
- Security assurance for non-expert users
- Approaches and technologies to improve security in programming
- Balancing security and privacy/anonymity
- Interactions between security technology and public policy

Submission Instructions

Submitted position papers must be no longer than six (6) single-spaced 8.5" x 11" pages, including figures, tables, and references. Author names and affiliations should appear on the title page.

Submissions must be in PDF and must be submitted via the Web submission form on the HotSec '08 Call for Papers Web site, <http://www.usenix.org/hotsec08/cfp>.

Authors will be notified of acceptance by June 25, 2008. Authors of accepted papers will produce a final PDF and the equivalent HTML by July 14, 2008. All papers will be available online to participants prior to the workshop and will be generally available online after the workshop.

Simultaneous submission of the same work to multiple venues, submission of previously published work, and plagiarism constitute dishonesty or fraud. USENIX, like other scientific and technical conferences and journals, prohibits these practices and may, on the recommendation of a program chair, take action against authors who have committed them. In some cases, pro-

gram committees may share information about submitted papers with other conference chairs and journal editors to ensure the integrity of papers under consideration. If a violation of these principles is found, sanctions may include, but are not limited to, barring the authors from submitting to or participating in USENIX conferences for a set period, contacting the authors' institutions, and publicizing the details of the case.

Note, however, that we expect that many position papers accepted for HotSec '08 will eventually morph into finished, full papers presented at future conferences.

Authors uncertain whether their submission meets USENIX's guidelines should contact the workshop organizers at hotsec08chair@usenix.org or the USENIX office, submissionspolicy@usenix.org.

Papers accompanied by nondisclosure agreement forms will not be considered. All submissions will be treated as confidential prior to publication in the Proceedings.

Workshop on Cyber Security Experimentation and Test (CSET '08)

Sponsored by USENIX, the Advanced Computing Systems Association

<http://www.usenix.org/cset08>

July 28, 2008

San Jose, CA

CSET '08 will be co-located with the 17th USENIX Security Symposium (USENIX Security '08), which will take place July 28–August 1, 2008.

Important Dates

Paper submissions due: *May 15, 2008*

Notification of acceptance: *June 30, 2008*

Electronic files due: *July 15, 2008*

Workshop Organizers

General Chair

Terry Benzel, *USC Information Sciences Institute*

Program Chairs

Sonia Fahmy, *Purdue University*

Jelena Mirkovic, *USC Information Sciences Institute*

Program Committee

Andy Bavier, *Princeton University*

Bob Braden, *USC Information Sciences Institute*

Tom Daniels, *Iowa State University*

Alefiya Hussain, *SPARTA*

Anthony Joseph, *University of California, Berkeley*

Sean Peisert, *University of California, Davis*

Peter Reiher, *University of California, Los Angeles*

Robb Ricci, *University of Utah*

Stephen Schwab, *SPARTA*

Mark Stamp, *San Jose State University*

Angelos Stavrou, *George Mason University*

Nick Weaver, *ICSI*

Vinod Yegneswaran, *SRI International*

Overview

Security challenges constantly grow in complexity and scale. To meet these challenges, security professionals need safe experiment environments, tools, and methodologies to:

- capture new threats,
- study threats through interactive experimentation,
- dissect and reassemble malware,

- pit new attacks against proposed defenses, and
- test defensive technologies in a large-scale, realistic setting.

This workshop aims to gather both researchers who use testbeds for security experimentation and testbed developers to share their ideas and results and to discuss open problems in this area. While we particularly invite papers that deal with security experimentation, we are also interested in papers that address general testbed/experiment issues that have implications on security experimentation such as traffic and topology generation, large-scale experiment support, experiment automation, etc.

Topics

Topics of interest include but are not limited to:

- Security experimentation
 - Experiments for Internet infrastructure protection (e.g., DNS, BGP)
 - Experiments with distributed denial-of-service attacks
 - Experiments with botnets and malware
 - Experiments that evaluate existing or novel defenses
 - Other testbed-based security experiments
- Testbeds and methodologies
 - Tools, methodologies, and infrastructure that support risky and/or realistic experimentation
 - Supporting experimentation at a large scale through virtualization or federation, or by scaling down problems while preserving realism and experiment fidelity
 - Experience in designing or deploying secure testbeds
 - Tools for realistic traffic generation
 - Instrumentation and automation of experiments; their archiving, preservation, and visualization
 - Diagnosis of and methodologies for dealing with experimental artifacts

- Fair sharing of testbed resources and experiment federation
- Hands-on security classes
 - Experiences teaching security classes that use testbeds for homework, in-class demonstrations, or class projects
 - Organizing red team/blue team exercises in classes

Submission Instructions

Submissions must be no longer than 6 pages—including tables, figures and references—in 2-column format, using 10 point fonts. Text outside a 6.5" by 9" block will be ignored. Submit your paper in PDF via the Web submission form on the CSET '08 Call for Papers Web site, <http://www.usenix.org/cset08/cfp>. We encourage authors to follow the U.S. National Science Foundation's guidelines for preparing PDF grant submissions:

- https://www.fastlane.nsf.gov/documents/pdf_create_pdfcreate_01.jsp

Each submission should have a contact author who should provide full contact information (email, phone, fax, mailing address). One author of each accepted paper will be required to present the work at the workshop.

Simultaneous submission of the same work to multiple venues, submission of previously published work, and plagiarism constitute dishonesty or fraud. USENIX, like other scientific and technical conferences and journals, prohibits these practices and may, on the recommendation of a program chair, take action against authors who have committed them. In some cases, program committees may share information about submitted papers with other conference chairs and journal editors to ensure the integrity of papers under consideration. If a violation of these principles is found, sanctions may include, but are not limited to, barring the authors from submitting to or participating in USENIX conferences for a set period, contacting the authors' institutions, and publicizing the details of the case.

Authors uncertain whether their submission meets USENIX's guidelines should contact the workshop organizers at cset08chairs@usenix.org or the USENIX office, submissionpolicy@usenix.org.

Papers accompanied by nondisclosure agreement forms will not be considered. All submissions will be treated as confidential prior to publication in the Proceedings.



Announcement and Call for Participation

22nd Large Installation System Administration Conference (LISA '08)

Sponsored by USENIX and SAGE

<http://www.usenix.org/lisa08>

November 9–14, 2008

San Diego, CA, USA

Important Dates

Extended abstract and paper submissions due: *May 8, 2008, 11:59 p.m. PDT*

Invited talk and workshop proposals due: *May 20, 2008*

Guru Is In and Hit the Ground Running proposals due: *May 31, 2008*

Notification to authors: *Mid-June 2008*

Poster proposals due, first round: *July 16, 2008*

Notification to poster presenters, first round: *July 23, 2008*

Final papers due: *August 20, 2008*

Poster proposals due, second round: *October 22, 2008*

Notification to poster presenters, second round: *October 29, 2008*

Conference Organizers

Program Chair

Mario Obejas, *Raytheon*

Program Committee

Paul Anderson, *University of Edinburgh*

Derek Balling, *Answers Corporation*

Travis Campbell, *AMD*

Narayan Desai, *Argonne National Laboratory*

Aleen Frisch, *Exponential Consulting*

Peter Baer Galvin, *Corporate Technologies*

Brent Hoon Kang, *University of North Carolina at Charlotte*

Chris McEniry, *Sony Computer Entertainment America*

David Parter, *University of Wisconsin*

David Plonka, *University of Wisconsin*

Melanie Rieback, *Vrije Universiteit*

Kent Skaar, *Bladologic*

Chad Verbowski, *Microsoft*

Invited Talks Coordinators

Rudi van Drunen, *Competa IT/Xlexit*

Philip Kizer, *Estacado Systems*

Workshops Coordinator

Lee Damon, *University of Washington*

Guru Is In Coordinator

John "Rowan" Littell, *California College of the Arts*

Hit the Ground Running Coordinator

Adam Moskowitz, *Permabit Technology Corporation*

Work-in-Progress Reports and Posters Coordinators

Brent Hoon Kang, *University of North Carolina at Charlotte*

Gautam Singaraju, *University of North Carolina at Charlotte*

Overview

Since 1987, the annual LISA conference has become the premier meeting place for professional system and network administrators. System administrators of all ranks, from novice to veteran, and of all specialties meet to exchange ideas, sharpen skills, learn new techniques, debate current issues, and mingle with colleagues and friends.

Attendees are diverse, a rich mix of nationalities and of educational, government, and industry backgrounds. We work in the full spectrum of computing environments (e.g., large corporations, small businesses, academic institutions, government agencies). We include full- and part-time students engaged in internships, as well as students and faculty deeply involved in

system administration research. Whereas many attendees focus on practical system administration, others focus on speculative system administration research. We support a broad range of operating systems (e.g., Solaris, Windows, Mac OS X, HP-UX, AIX, BSD, Linux) and commercial and open source applications, and we run them on a variety of infrastructures.

The conference's diverse group of participants are matched by a broad spectrum of conference activities:

- A training program for both beginners and experienced attendees covers many administrative topics, ranging from basic procedures to using cutting-edge technologies.
- Refereed papers present the latest developments and ideas related to system and network administration.
- Workshops, invited talks, and panels discuss important and timely topics in depth and typically include lively and/or controversial debates and audience interaction.
- Work-in-Progress Reports (WiPs) and poster sessions provide brief looks ahead to next year's innovations.
- The Hit the Ground Running track presents multiple important topics in single sessions, distilled down to a few solid points.
- LISA also makes it easy for people to interact in more informal settings:
 - Noted experts answer questions at Guru Is In sessions.
 - Participants discuss/celebrate/commiserate about a shared interest at Birds-of-a-Feather (BoF) sessions.
 - Vendors answer questions and offer solutions at the Exhibition.

Finally, we strongly encourage informal discussions among participants on both technical and nontechnical topics in the famous "hallway track." LISA is a place to learn and to have fun!

Get Involved!

The theme for LISA '08 is "Real World System Administration."

Experts and old-timers don't have all the good ideas. We welcome participants who will provide concrete ideas to immediately implement, as well as those whose research will forge tomorrow's computing infrastructures. We are particularly keen to showcase novel solutions or new applications of mature technologies. This is your conference, and we want you to participate. Here are examples of ways to get involved in this 22nd LISA conference:

- Submit a draft paper or extended abstract for a refereed paper.
- Suggest an invited talk or panel discussion.
- Propose a short Hit the Ground Running presentation.
- Share your experience by leading a Guru Is In session.
- Create and lead a workshop.
- Propose a tutorial topic.
- Present a Work-in-Progress Report (WiP) or submit a poster.
- Organize a Birds-of-a-Feather (BoF) session.
- Email an idea to the Program Chair: lisa08ideas@usenix.org.

Refereed Papers

Effective administration of a large site requires a good understanding of modern tools and techniques, together with their underlying principles—but the human factors involved in managing and applying these technologies in a production environment are equally important. Bringing together theory and practice is an important goal of the LISA conference, and practicing system administrators, as well as academic researchers, all have valuable contributions to make. A selection of possible topics for refereed papers appears in a

separate section below, but submissions are welcome on any aspect of system administration, from the underlying theory of a new configuration technique to a case study on the management of a successful site merger.

Whatever the topic, it is most important that papers present results in the context of current practice and previous work: they should provide references to related work and make specific comparisons where appropriate. The crucial component is that your paper present something new or timely; for instance, something that was not previously available, or something that had not previously been published. Careful searching for publications on a similar theme will help to identify any possible duplication and provide pointers to related work; the USENIX site contains most previous LISA conference proceedings, which may provide a starting point when searching for related publications: <http://www.usenix.org/events/byname/lisa.html>.

Cash prizes will be awarded at the conference for the best refereed paper as well as for the best refereed paper for which a student is the lead author; a special announcement will also be made about these two papers.

Proposal and Submission Details

Anyone who would like help in writing a proposal should contact the program chair at lisa08chair@usenix.org. The conference organizers are keen to make sure that good work gets published, and we are happy to help at any stage in the process.

Proposals may be submitted as draft papers or extended abstracts.

- **Draft papers:** This is the preferred format. A draft paper proposal is limited to 16 pages, including diagrams, figures, references, and appendices. It should be a complete or near-complete paper, so that the Program Committee has the best possible understanding of your ideas and presentation.
- **Extended abstracts:** An extended abstract proposal should be about 5 pages long (at least 500 words, not counting figures and references) and should include a brief outline of the final paper. The form of the full paper must be clear from your abstract. The Program Committee will be attempting to judge the quality of the final paper from your abstract. This is harder to do with extended abstracts than with the preferred form, draft papers, so your abstract must be as helpful as possible in this process to be considered for acceptance.

Paper authors are also invited to submit posters, as outlined below, to accompany their presentations; these provide an overview of the work and a focal point for delegates to meet with the author.

General submission rules:

- All submissions must be electronic, in ASCII or PDF format only. Proposals must be submitted using the Web form located on the LISA '08 Call for Papers Web site, <http://www.usenix.org/lisa08/cfp>.
- Submissions should include a list of appropriate topic keywords or tags above the body text of the draft paper or extended abstract. For example:
Tags: security, research, IPv6
Suggested tags include security, research, case study, backups, configuration management, database, Web, printing, filesystem, authentication, and VMs. Authors may include additional tags as well.
- Submissions whose main purpose is to promote a commercial product or service will not be accepted.
- Submissions may be submitted only by the author of the paper. No third-party submissions will be accepted.
- All accepted papers must be presented at the LISA conference by at least one author. One author per paper will receive a registration discount of \$200. USENIX will offer a complimentary registration for the technical program upon request.
- Authors of an accepted paper must provide a final paper for publication in the conference proceedings. Final papers are limited to 16 pages, including diagrams, figures, references, and appendices. Complete instructions will be sent to the authors of accepted papers. To aid authors in creating a paper suitable for LISA's audience, authors of accepted proposals will be assigned one or more shepherds to help with the process of completing the paper. The shepherds will read one or more intermediate drafts and provide comments before the authors complete the final draft.
- Simultaneous submission of the same work to multiple venues, submission of previously published work, and plagiarism constitute dishonesty or fraud. USENIX, like other scientific and technical conferences and journals, prohibits these practices and may, on the recommendation of a

program chair, take action against authors who have committed them. In some cases, to ensure the integrity of papers under consideration, program committees may share information about submitted papers with other conference chairs and journal editors. If a violation of these principles is found, sanctions may include, but are not limited to, barring the authors from submitting to or participating in USENIX conferences for a set period, contacting the authors' institutions, and publicizing the details of the case. Authors uncertain whether their submission meets USENIX's guidelines should contact the program chair, lisa08chair@usenix.org, or the USENIX office, submissionspolicy@usenix.org.

- Papers accompanied by nondisclosure agreement forms will not be considered. All submissions will be treated as confidential prior to publication in the Proceedings.

For administrative reasons, every submission must list:

1. Paper title, and names, affiliations, and email addresses of all authors. Indicate each author who is a full-time student.
2. The author who will be the contact for the Program Committee. Include his/her name, affiliation, paper mail address, daytime and evening phone numbers, email address, and fax number (as applicable).

For more information, please consult the detailed author guidelines at <http://www.usenix.org/events/lisa08/cfp/guidelines.html>. **Paper and extended abstract submissions are due by 11:59 p.m. PDT on May 8, 2008.** Authors will be notified by mid-June whether their papers have been accepted.

Training Program

LISA offers state-of-the-art tutorials from top experts in their fields. Topics cover every level from introductory to highly advanced. You can choose from over 50 full- and half-day tutorials ranging from Linux-HA, through performance tuning, Solaris, Windows, Perl, Samba, network troubleshooting, security, network services, filesystems, backups, Sendmail, spam, and legal issues, to professional development.

To provide the best possible tutorial offerings, USENIX continually solicits proposals and ideas for new tutorials, especially on subjects not yet covered. If you are interested in presenting a tutorial or have an idea for a tutorial you would like to see offered, please contact the Education Director, Daniel V. Klein, at tutorials@usenix.org.

Invited Talks

An invited talk discusses a topic of general interest to attendees. Unlike a refereed paper, this topic need not be new or unique but should be timely and relevant or perhaps entertaining. A list of suggested topics is available in a separate section below. An ideal invited talk is approachable and possibly controversial. The material should be understandable by beginners, but the conclusions may be disagreed with by experts. Invited talks should be 60–70 minutes long, and speakers should plan to take 20–30 minutes of questions from the audience.

Invited talk proposals should be accompanied by an abstract of less than one page in length describing the content of the talk. You can also propose a panel discussion topic. It is most helpful to us if you suggest potential panelists. Proposals of a business development or marketing nature are not appropriate. Speakers must submit their own proposals; third-party submissions, even if authorized, will be rejected.

Please email your proposal to lisa08it@usenix.org. **Invited talk proposals are due May 20, 2008.**

The Guru Is In

Everyone is invited to bring perplexing technical questions to the experts at LISA's unique Guru Is In sessions. These informal gatherings are organized around a single technical area or topic. Email suggestions for Guru Is In sessions or your offer to be a Guru to lisa08guru@usenix.org. **Guru Is In proposals are due May 31, 2008.**

Hit the Ground Running

This track consists of five high-speed presentations packed into each 90-minute session. The presentations are intended to give attendees a "brain dump" on a new technology, new features in an existing protocol or service, an overview of the state of the art of a technique or practice, or an introduction to an existing technology that is becoming more widely used.

HTGR proposals should be accompanied by an abstract of less than one page in length describing the content of the talk. Proposals of a business development or marketing nature are not appropriate. Speakers must submit their own proposals; third-party submissions, even if authorized, will be rejected. Suggestions for desired HTGR presentations are also welcome (if possible, accompanied by a suggestion for a speaker).

Please email your proposal to lisa08htg@usenix.org. **HTGR proposals are due May 31, 2008.**

Workshops

One-day workshops are hands-on, participatory, interactive sessions where small groups of system administrators have an opportunity to discuss a topic of common interest. Workshops are not intended as tutorials, and participants normally have significant experience in the appropriate area, enabling discussions at a peer level. However, attendees with less experience often find workshops useful and are encouraged to discuss attendance with the workshop organizer.

A workshop proposal should include the following information:

- Title
- Objective
- Organizer name(s) and contact information
- Potential attendee profile
- Outline of potential topics

Please email your proposal to lisa08workshops@usenix.org. **Workshop proposals are due May 20, 2008.**

Posters

This year's conference will include a poster session. This is an opportunity to display a poster describing recent work. The posters will be on display during the conference, and fixed times will be advertised when authors should be present to discuss their work with anyone who is interested. This provides a very good opportunity to make contact with other people who may be interested in the same area. Student posters, practitioners sharing their experiences, and submissions from open source communities are particularly welcome.

To submit a poster, please send a 1–5 page proposal or 6–12 PowerPoint slides in PDF to lisa08posters@usenix.org. Please include your name, your affiliation, and the title of your poster. There will be two rounds of submission and review of poster proposals. You may submit your poster during either the first or the second round.

The first deadline for submissions is July 16, 2008. Please submit your poster by this deadline if you plan to apply for a student conference grant or will be traveling to LISA '08 from outside the United States and need to allow time for visa preparation. Accepted poster authors from the first round will be notified by July 23.

The second deadline for submissions is October 22. Accepted poster authors from the second round will be notified by October 29. Completed posters from both rounds will be required by the start of the conference.

Poster presenters who would also like to give a short presentation may also register for a WiP as below.

Work-in-Progress Reports (WiPs)

A Work-in-Progress Report (WiP) is a very short presentation about current work. It is a great way to poll the LISA audience for feedback and interest. We are particularly interested in presentations of student work. To schedule a short presentation, send email to lisa08wips@usenix.org or sign up on the first day of the technical sessions.

Birds-of-a-Feather Sessions (BoFs)

Birds-of-a-Feather sessions (BoFs) are informal gatherings organized by attendees interested in a particular topic. BoFs will be held in the evening. BoFs may be scheduled in advance by emailing bofs@usenix.org. BoFs may also be scheduled at the conference.

Possible Topics for Authors and Speakers

Technical Challenges

- Authentication and authorization: “Single sign-on” technologies, identity management

- Autonomic computing: Self-repairing systems, zero administration systems, fail-safe design
- Configuration management: Specification languages, configuration deployment
- Data center design: Modern methods, upgrading old centers
- Data management: DBMS management systems, deployment architectures and methods, real world performance
- Email: Mail infrastructures, spam prevention
- Grid computing: Management of grid fabrics and infrastructure
- Hardware: Multicore processor ramifications
- Mobile computing: Supporting and managing laptops and remote communications
- Multiple platforms: Integrating and supporting multiple platforms (e.g., Linux, Windows, Macintosh)
- Networking: New technologies, network management
- Security: Malware and virus prevention, security technologies and procedures, response to cyber attacks targeting individuals
- Service
- Standards: Enabling interoperability of local and remote services and applications
- Storage: New storage technologies, remote filesystems, backups, scaling
- Web 2.0 technologies: Using, supporting, and managing wikis, blogs, and other Web 2.0 applications
- Virtualization: Managing and configuring virtualized resources

Professional Challenges

- Budgeting: Definitions and methods
- Communication: Tools and procedures for improving communication between administrators and users, distribution organizations, or teams
- Consolidation: Merging and standardizing infrastructures and procedures
- Devolution: Managing dependence on devolved services (calendars, mail, Web 2.0, etc.) and users
- Ethics: Common dilemmas and outcomes
- Flexibility: Responding effectively to changes in technology and business demands
- In-house development: The (dis)advantages and pitfalls of in-house technology development
- Legislation: Security, privacy
- Management: The interface and transition between “technical” and “managerial”
- Metrics: Measuring and analyzing the effectiveness of technologies and procedures
- Outsourcing/offshoring system administration: Is it possible?
- Proactive administration: Transitioning from a reactive culture
- Standardizing methodologies: Sharing best practice
- Training and staff development: Developing and retaining good system administrators; certifications
- User support: Systems and procedures for supporting users

Contact the Chair

The program chair, Mario Obejas, is always open to new ideas that might improve the conference. Please email your ideas to lisa08ideas@usenix.org.

Final Program and Registration Information

Complete program and registration information will be available in August 2008 at the conference Web site, <http://www.usenix.org/lisa08>. If you would like to receive the latest USENIX conference information, please join our mailing list at <http://www.usenix.org/about/mailling.html>.

Sponsorship and Exhibit Opportunities

The oldest and largest conference exclusively for system administrators presents an unparalleled marketing and sales opportunity for sponsoring and exhibiting organizations. Your company will gain both mind share and market share as you present your products and services to a prequalified audience that heavily influences the purchasing decisions of your targeted prospects. For more details please contact exhibits@usenix.org.



CALL FOR PAPERS

The Middleware conference is a forum for the discussion of important innovations and recent advances in the design, construction and uses of middleware. Middleware is distributed-systems software that resides between the applications and the underlying operating systems, network protocol stacks, and hardware. Its primary role is to functionally bridge the gap between application programs and the lower-level hardware and software infrastructure in order to coordinate how application components are connected and how they interoperate. Following the success of past conferences in this series, the 9th International Middleware Conference will be the premier event for middleware research and

technology in 2008. The scope of the conference is the design, implementation, deployment, and evaluation of distributed system platforms and architectures for future computing and communication environments. Highlights of the conference will include a high quality technical program, invited speakers, an industrial track, poster and demo presentations, a doctoral symposium, and workshops. Submissions on a diversity of topics are sought, particularly ones that identify new research directions. The topics of the conference include, but are not limited to

Sponsored by:
(pending)



Platforms and Architectures:

- *Middleware for Web services and Web-service composition*
- *Middleware for cluster and grid computing*
- *Peer-to-peer middleware solutions*
- *Event-based, publish/subscribe, and message-oriented middleware*
- *Communication protocols and architectures*
- *Middleware for ubiquitous and mobile computing*
- *Middleware for embedded systems and sensor networks*
- *Middleware for next generation telecommunication platforms*
- *Semantic middleware*
- *Service-oriented architectures*
- *Standard middleware architectures*

- *Reconfigurable, adaptable, and reflective middleware approaches*

Systems issues:

- *Advanced middleware support for high confidence dynamic integrated systems*
- *Reliability, fault tolerance, and quality-of-service in general*
- *Scalability of middleware: replication and caching*
- *Systems management, including solutions for autonomic and self-managing middleware*
- *Middleware feedback control solutions for self-regulation*
- *Real-time solutions for middleware platforms*
- *Information assurance and security*
- *Evaluation techniques for middleware solutions*

- *Middleware support for multimedia streaming*
- *Middleware solutions for (large scale) distributed databases*

Design principles and tools:

- *Formal methods and tools for designing, verifying, and evaluating middleware*
- *Model-driven architectures*
- *Software engineering for middleware*
- *Engineering principles and approaches for middleware*
- *Novel development paradigms, APIs, and languages*
- *Existing paradigms revisited: object models, aspect orientation, etc.*
- *On-the-fly management and configuration of middleware*

The conference also strongly encourages submission of industry-focused and use case studies; full papers should be submitted to the main program, where they will be reviewed using appropriate criteria (e.g. emphasizing experience and system evolution), and accepted papers will be published in the main conference proceedings. Additionally, short industry-focused papers may be submitted to a special industrial track; accepted short papers will be presented at the conference and published in the ACM Digital Library. Details on the industrial track will be available shortly. Note that submissions to the main program may indicate a willingness to be referred to the industrial track if a paper is not accepted to the main program.

Proceedings:

The proceedings of Middleware 2008 will be published as a Springer-Verlag volume in the Lecture Notes in Computer Science Series.

Submission Guidelines:

Papers must not exceed 20 pages, including abstract, all figures, all tables, and references. Papers should include a short abstract and up to 6 keywords. Submitted papers should follow the formatting instructions of the Springer LNCS Style (for style and formatting guidelines Please check the Information for Authors page at Springer at <http://www.springer.de/comp/lncs/authors.html>).

Submitted papers may not be submitted for conference publication, journal publication, or be under review for any other conference or journal. For any questions regarding this matter, please contact the program chairs. Submissions will be handled via the conference management system that will be announced later.

Important Dates: (** Note that deadlines will not be extended **)

23 April 2008	Abstract registration (hard deadline)
30 April 2008	Paper submission (hard deadline)
16 July 2008	Acceptance notification
31 August 2008	Camera ready copy due
1 December 2008	Conference begins

Organization:

Conference Chairs:	Wouter Joosen (K.U.Leuven, Belgium) Yolande Berbers (K.U.Leuven, Belgium)	Industry Chair:	Fred Douglass (IBM, USA)
PC Chairs:	Valerie Issarny (INRIA, France) Rick Schantz (BBN Technologies, USA)	Work In Progress Chair:	Cecilia Mascolo (University of Cambridge, UK)
Workshop & Tutorials Chairs:	Frank Eliassen (University of Oslo, Norway) Hans-Arno Jacobsen (University of Toronto, Canada)	Publicity Chair:	Michael Atighetchi (BBN Technologies, USA)
		Posters and Demos Chair:	Sonia Ben Mokhtar (UCL, UK)
		Doctoral Symposium Chair:	Bert Lagaisse (K.U.Leuven, Belgium)
		Local Arrangements Chair:	Sam Michiels (K.U.Leuven, Belgium)
			Davy Preuveneers (K.U.Leuven, Belgium)

Programme Committee:

Gustavo Alonso (ETH Zurich, Switzerland)	Frank Eliassen (University of Oslo, Norway)	Mark Linderman (Air Force Research Laboratory, USA)
Christiana Amza (University of Toronto, Canada)	Markus Endler (PUC-Rio, Brazil)	Joe Loyall (BBN Technologies, USA)
Jean Bacon (University of Cambridge, UK)	Pascal Felber (University of Neuchatel, Switzerland)	Cecilia Mascolo (University of Cambridge, UK)
Dave Bakken (Washington State University, USA)	Paulo Ferreira (INESC ID / Tech. Univ. of Lisbon, Portugal)	Satoshi Matsuoka (Tokyo Institute of Technology, Japan)
Guruduth Banavar (IBM Research, India)	Nikolaos Georgantas (INRIA, France)	Elie Najm (ENST Paris, France)
Alberto Bartoli (University of Trieste, Italy)	Chris Gill (Washington University, USA)	Bala Natarajan (Symantec Corp., India)
Christian Becker (Universitat Mannheim, Germany)	Paul Grace (Lancaster University, UK)	Gian Pietro Picco (University of Trento, Italy)
Gordon Blair (Lancaster University, UK)	Indranil Gupta (University of Illinois at Urbana-Champaign, USA)	Alexander Reinefeld (Zuse Institute Berlin, Germany)
Roy H Campbell (University of Illinois at Urbana Champaign, USA)	Qi Han (Colorado school of Mines, USA)	Luis Rodrigues (INESC-ID/IST, Portugal)
Renato Cerqueira (PUC-Rio, Brazil)	Peter Honeyman (CITI, University of Michigan, USA)	Antony Rowstrom (Microsoft Research, UK)
Angelo Corsaro (PrismTech, USA)	Gang Huang (Peking University, China)	Douglas C. Schmidt (Vanderbilt University, USA)
Paolo Costa (Vrije Universiteit Amsterdam, Netherlands)	Shanika Karunasekera (University of Melbourne, Australia)	Jean-Bernard Stefani (INRIA, France)
Geoff Coulson (Lancaster University, UK)	Bettina Kemme (McGill University, Canada)	Gautam Thaker (Lockheed Martin Adv. Tech. Labs, USA)
Jan De Meer (SmartSpaceLab, Germany)	Fabio Kon (University of Sao Paulo, Brazil)	Peter Triantafillou (University of Patras, Greece)
Fred Douglass (IBM T.J. Watson Research Center, USA)	Doug Lea (Oswego State University, USA)	Apostolos Zarras (University of Ioannina, Greece)
Naranker Dulay (Imperial College London, UK)	Rodger Lea (University of British Columbia, Canada)	

There's a whole lot of technology in the queue. are you ready?

What's next?

acm



Get ready with **ACM Queue**—the technology magazine focused on problems that don't have easy answers—yet.

Queue dissects the challenges of emerging technologies. **Queue** targets the problems and pitfalls just ahead. **Queue** helps you plan for the future. **Queue** poses the hard questions you'd like to ask.

Isn't that what you've been looking for?

www.acmqueue.org

Subscribe now at **ACM Queue's** special, limited-time charter subscription rate of \$19.95 for ACM members. Use the subscription card in this issue or go to the **ACM Queue** web site at www.acmqueue.org

www.acmqueue.org

LINUX+

DVD

The best source for Linux users
www.linuxmagazine.org/en

Check it out at the nearest
Barnes&Noble and Borders stores!

WANTED

LINUX+ 2DVDs CentOS Linux 5.0 Fedora 7.0 Open Office 2.2.1

LINUX+

LINUX ENVIRONMENT FOR EXPERTS Vol. 2 No. 1 Issue 4 (2007/4) Price \$19.95 US \$19.95 CAN \$29.95

DVD

EXPLORE CENTOS

Install & Configure Step-by-Step

+ FEDORA 7.0
Closer Look at RedHat Distributions

Using IP Tables
How to use Netfilter's logging capabilities

Find an Easy Way to Share Information
Set-up an eGroupWare Server with Zenserver

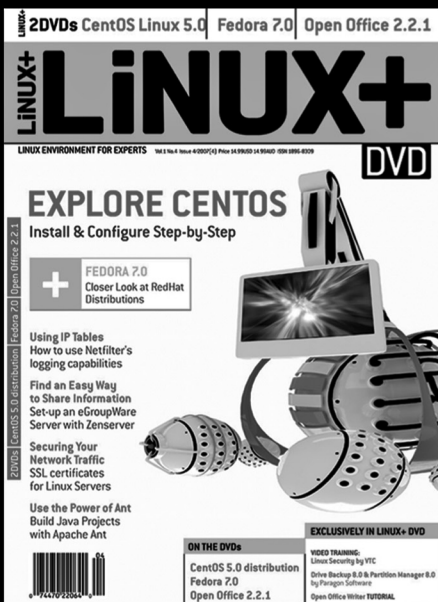
Securing Your Network Traffic
SSL certificates for Linux Servers

Use the Power of Ant
Build Java Projects with Apache Ant

ON THE DVDs
CentOS 5.0 distribution
Fedora 7.0
Open Office 2.2.1

EXCLUSIVELY IN LINUX+ DVD

VIDEO TRAINING:
Linux Security by VTC
Drive Backup & D.D. Partition Manager 8.0 by Paragon Software
Open Office Writer TUTORIAL



LINUX+ 2DVDs Mandriva One 2008 & Extras

LINUX+

LINUX ENVIRONMENT FOR EXPERTS Vol. 2 No. 1 Issue 4 (2007/4) Price \$24.95 US \$24.95 CAN \$34.95

DVD

Mandriva Uncovered

Install & Configure - Ins and Outs

+ GREAT multimedia BOX SECTION
LINUX WORLD OF GAMES

IP Spoofing Attacks
How to prevent IP spoofing attacks

Kernel Security
Additional level of Linux security

Amanda-Backup Solution
How to keep your files safe

PCLinuxOS
Linux hard to use?
Find out the truth

ON THE DVDs
MANDRIVA ONE 2008
MANDRIVA LINUX WORLD OF GAMES
INCLUDE: TORQUE

PLUS
AN DEFINITIVE SECURITY
FOR MANDRIVA
OS/ME/MS/MSX SERVERS



LINUX+ 2DVDs Debian 4.0 plus Extras Filezilla 3.0.4

LINUX+

LINUX ENVIRONMENT FOR EXPERTS Vol. 2 No. 1 Issue 4 (2007/4) Price \$24.95 US \$24.95 CAN \$34.95

DVD

HANDS-ON WITH DEBIAN

Installation & Configuration Guide

+ GRAPHICS CARDS - CONSUMERS' CHOICE
CHOOSE THE BEST CARD FOR LINUX

ON THE DVDs
DEBIAN 4.0
Mandriva Linux World of Games
MANDRIVA ONE 2008
EXCLUSIVELY
FOR LINUX+ DVD
4-HOUR VIDEO TRAINING
MYSQL BY VTC

Linux video drivers
Which driver is right for you


Debian Package Management
How to keep your system safe & up-to-date

Introduction to Installing and Using Annix
Provide a secure, fast and reliable server operating system

MySQL backup with ZRM
Recover your data in a safe way

A Portable Web Database Server
Create a portable system yourself

Hard disk recovery
An easy way to recover your data



Dr. Dobb's®
**ARCHITECTURE
& DESIGN WORLD**

JULY 21-24, 2008
HYATT REGENCY MCCORMICK PLACE
CHICAGO, IL

**SAVE
THE
DATE!**

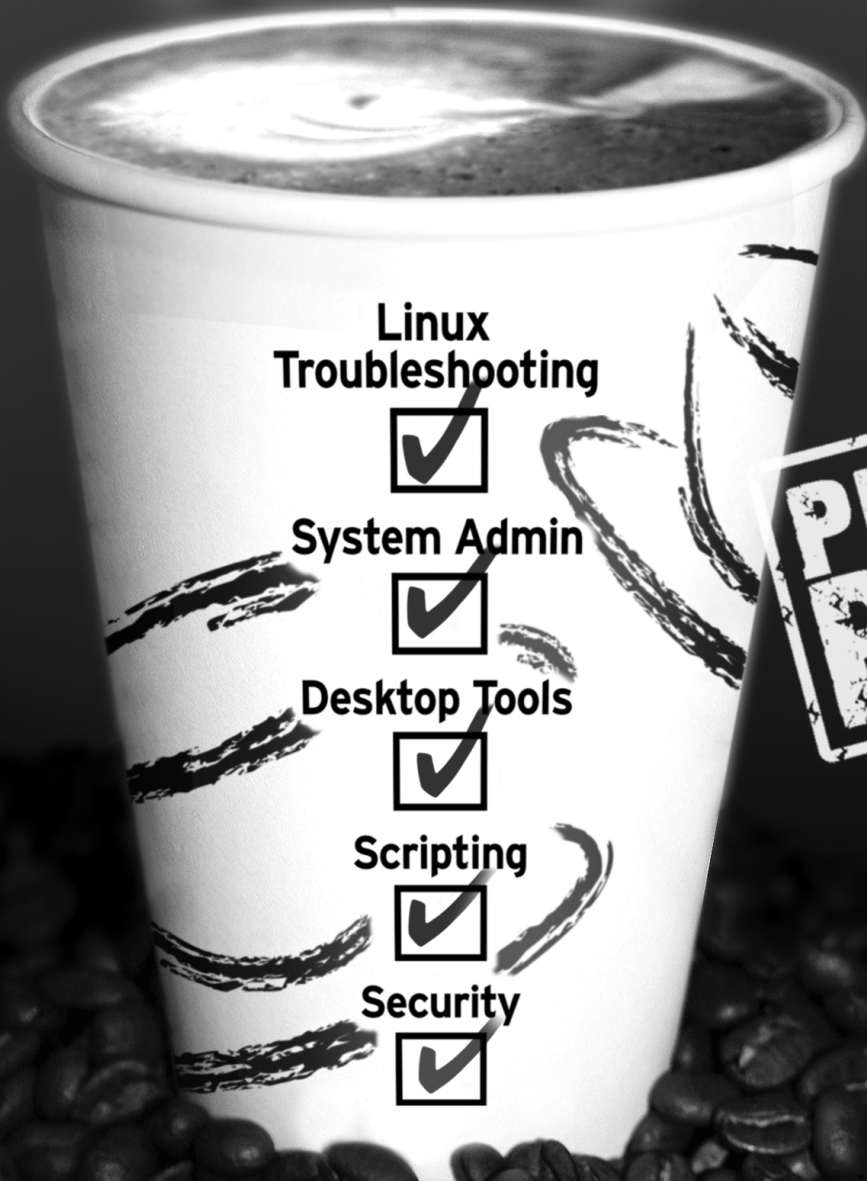
OVER 75 SESSIONS IN 6 FOCUSED TRACKS

- Agile Development & Methods
- Modeling & Design
- Roll-Up-Your-Sleeves
- Service-Oriented Architecture
- Solutions/Technical Architecture
- Traditional Development & Methods

PLUS:

Keynotes, Case Studies, Panels, Birds-of-a-Feathers, Tabletop Exhibits, Sponsored Technical Sessions and Special Events

**CONFERENCE DETAILS AND ONLINE REGISTRATION WILL BE
AVAILABLE AT WWW.SDEXPO.COM IN MID-APRIL 2008.**



Linux
Troubleshooting



System Admin



Desktop Tools



Scripting



Security



**PREMIUM
BLEND**

LINUX PRO
MAGAZINE

If you like the taste
of Linux, why not treat
yourself to the best?

Linux Pro Magazine delivers real-world solutions for the technical reader. In every issue, you'll find advanced techniques for configuring and securing Linux systems. Learn about the latest tools and discover the secrets of the experts in Linux Pro. Each issue includes a full Linux distribution on DVD.

www.linuxpromagazine.com

USENIX '08

2008 USENIX ANNUAL TECHNICAL CONFERENCE

June 22–27, 2008 • Boston, MA

USENIX '08 will feature:

- **An extensive Training Program, covering crucial topics and led by highly respected instructors**
- **Keynote Address, "The Parallel Revolution Has Started: Are You Part of the Solution or Part of the Problem?" by David Patterson, *Director, University of California, Berkeley, Parallel Computing Laboratory***
- **Technical Sessions, featuring the Refereed Papers Track, Invited Talks, Guru Is In Sessions, and a Poster Session**
- **Plus workshops, BoFs, and more!**

Join the community of programmers, developers, and systems professionals in sharing solutions and fresh ideas.

Register by June 6, 2008, and save!

<http://www.usenix.org/usenix08>

;login:

USENIX Association
2560 Ninth Street, Suite 215
Berkeley, CA 94710

POSTMASTER
Send Address Changes to ;login:
2560 Ninth Street, Suite 215
Berkeley, CA 94710

PERIODICALS POSTAGE
PAID
AT BERKELEY, CALIFORNIA
AND ADDITIONAL OFFICES
