

LINUX JOURNAL™

Since 1994: The Original Magazine of the Linux Community

OpenSSH
Under the Hood

Linux on PlayStation 3

SECURITY

- » SELINUX
MULTI-CATEGORY
SECURITY
- » PACKETFENCE NETWORK
ACCESS CONTROL
- » SINGLE PACKET
AUTHORIZATION



iptables Primer

Asterisk Time-Zone Processing

Python Manipulates ODF

Magic with Inkscape and XLST

APRIL 2007 | ISSUE 156
www.linuxjournal.com



BelltownMedia

USA \$5.00
CAN \$6.50

0 71486 03102 4

0 4

Enterprise and High-Performance Computing Under Your Control



Industry Leading 4P x86 Computing

Innovative server technology with outstanding performance and memory scalability

4-Way **XtremeWorkstation**™



- AMD Opteron™ 8000 Series processors
- Up to 128GB of DDR2 533/667 memory
- Up to 6.0TB SATA or 2.4TB SAS
- 2 PCI-E x16 slots for high-end graphics card
- Hot-swappable drives
- Windows® or Linux OS

4-Way 3U **XtremeServer**™



- AMD Opteron™ 8000 Series processors
- Up to 128GB of DDR2 533/667 memory
- Up to 4.5TB SATA or 1.8TB SAS
- 2 PCI-E x16 and 3 PCI-X slots
- Redundant power supplies and fans
- Hot-swappable drives
- ServerDome Management – IPMI 2.0
- Windows® or Linux OS

Leveraging Xen Virtualization with the Appro XtremeServer

Go to [http://www.appro.com/whitepaper/White Papers.asp](http://www.appro.com/whitepaper/White%20Papers.asp)

- AMD Opteron™ Processors:
- Quad-Core Ready - increase capacity without altering datacenter infrastructure
 - Best performance per-watt with energy-efficient DDR2
 - Optimized system performance with Direct Connect Architecture

For more information, please visit www.appro.com
or call Appro Sales at 800-927-5464 or 408-941-8100.



Today, Dan configured a switch in London, rebooted servers in Sydney, and watched his team score the winning goal in St. Louis.

With Avocent data center solutions, the world can finally revolve around you. Avocent puts secure access and control right at your finger tips – from multi-platform servers to network routers, your local data center to branch offices, across the hall or around the globe. Let others roll crash carts to troubleshoot – with Avocent, trouble is on ice.

To learn more, visit us at www.avocent.com/ice to download Data Center Control: Guidelines to Achieve Centralized Management whitepaper or call 866.277.1924 for a demo today.



CONTENTS

APRIL 2007
Issue 156



ILLUSTRATION ©ISTOCKPHOTO.COM/EMERAH TURUDU

FEATURES

51 Single Packet Authorization
Want something better than port knocking?
Michael Rash

54 eCryptfs: a Stacked Cryptographic Filesystem
How does your filesystem security stack up?
Mike Halcrow

60 Multi-Category Security in SELinux in Fedora Core 5
Not in the military and still want SELinux?
Russell Coker

64 PacketFence
An open-source solution to manage your security.
Ludovic Marcotte and Dominik Gehl

ON THE COVER

- *OpenSSH Under the Hood*, p. 74
- *Linux on PlayStation 3*, p. 70
- *SELinux Multi-Category Security*, p. 60
- *PacketFence Network Access Control*, p. 64
- *Single Packet Authorization*, p. 51
- *iptables Primer*, p. 78
- *Asterisk Time-Zone Processing*, p. 82
- *Python Manipulates ODF*, p. 91
- *Magic with Inkscape and XLST*, p. 86

The competition doesn't stand a chance.



If you base deployment decisions on performance and price, Coyote Point's for you. We've cornered that market.

To prove it we asked The Tolly Group to evaluate our E350si application traffic manager against the competition. The results speak for themselves.

Throughput? Almost 40% more than others in our space. Cost of transactions per second? Up to four times less. Connection rate? In some cases, one-sixth the cost. One-sixth! And we're told Coyote Point is the #1 choice for today's open source networks.

But don't just take our word for it. Get the facts. Call 1.877.367.2696 or write info@coyotepoint.com for your free copy of the full Tolly Report.



CoyotePoint
Systems Inc



CONTENTS

APRIL 2007

Issue 156

COLUMNS

- 22** REUVEN LERNER'S
AT THE FORGE
Dojo Events and Ajax
- 26** MARCEL GAGNÉ'S
COOKING WITH LINUX
Security for Your Data—It's
Totally Mondo!
- 32** DAVE TAYLOR'S
WORK THE SHELL
Displaying Image Directories
in Apache
- 34** MICK BAUER'S
PARANOID PENGUIN
Linux Firewalls for Everyone
- 38** JON "MADDOG" HALL'S
BEACHHEAD
The Outer Banks
- 40** DOC SEARLS'
LINUX FOR SUITS
Why an iPhone When We Can
Make Our Own OpenPhone?
- 96** NICHOLAS PETRELEY'S
/VAR/OPINION
Do Not Forget What People Fetch

QUICK TAKES

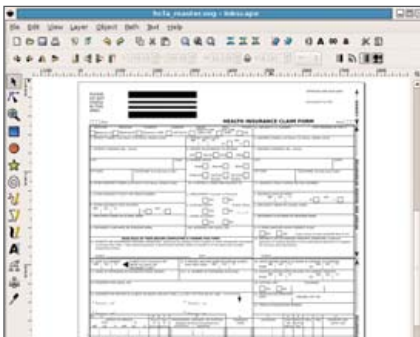
- 46** MYSQL DESERVES A
DOUBLE TAKE
Reuven M. Lerner

IN EVERY ISSUE

- 8** LETTERS
12 UPFRONT
20 TECH TIPS
44 NEW PRODUCTS
81 ADVERTISERS INDEX

INDEPTH

- 70** NEED FOR SPEED:
PS3 LINUX!
It's a miracle; it's a dog.
Dave Taylor
- 74** THE OPENSSSH PROTOCOL
UNDER THE HOOD
What's SSH all about?
Girish Venkatachalam
- 78** STARTING A LINUX
FIREWALL FROM SCRATCH
How do you start using iptables?
Dinil Divakaran
- 82** TIME-ZONE PROCESSING
WITH ASTERISK, PART II
When is a good time to call?
Matthew Gast
- 86** USE INKSCAPE AND XSLT TO
CREATE CROSS-PLATFORM
REPORTS AND FORMS
Dynamic forms using Inkscape
with XSLT
Chad Files



- 91** EXTRACT AND PARSE ODF
FILES WITH PYTHON
Want to use Python to dissect
an ODF file?
Kamran Husain



70 LINUX ON PLAYSTATION 3

Next Month

AJAX/WEB SERVICES

Next month, we'll have everything you need to help you graduate to Web 2.0, or for the more experienced, take Web 2.0 to the max. Haven't got your feet wet yet? We'll introduce you to the raw basics of Ajax in one article, and then take you deeper in another. Already swimming in the deep end? We'll tell you about the new IDE Aptana, the development tool MochiKit, and get you working with four great Ajax plugins for Wordpress. And, wait until you get a glimpse of Zimbra, which takes Ajax to the max.

As always, there's much more. We'll get you working with Single Packet Authorization in part two of our series on the topic and give you the lowdown on PostgreSQL.

Are you
shocked
by the
high cost
of iSCSI &
Fibre Channel
storage?



AoE is your answer!

ATA-over-Ethernet = simple, low cost, expandable storage.

www.coraid.com



EtherDrive® SR1520

- RAID enabled 3U appliance with 15 slots for hot swap SATA disks
- Check out our other Storage Appliances and NAS Gateway



1. Ethernet Storage – without the TCP/IP overhead!
2. Unlimited expandability, at the lowest possible price point!!
3. You want more storage...you just buy more disks – it's that simple!!!

Visit us at www.coraid.com
for more information.



1.706.548.7200

The Linux Storage People

www.coraid.com

axigen

MAIL SERVER

ISP EDITION



The ISP Solution of Choice
for a Carrier Class Mail Server

CLUSTERING SYSTEM

High Availability
(RHCS integration)

Distributed mailboxes
(LDAP routing)

ADVANCED SECURITY

Message Acceptance/Routing policies
Antivirus & Antispam integration
Anti-impersonation function

REPORTING ENGINE
(over 100 personalized reports)

ONLINE BACKUP AND
RESTORE SYSTEM

PERSONALIZED SUPPORT

Available 24x7x365

Priority response

Dedicated Support Engineer

On-site installation assistance

www.axigen.com/Lj

Visit now to start your free trial!

LINUX JOURNAL

Editor in Chief

Nick Petreley, ljeditor@linuxjournal.com

Executive Editor	Jill Franklin jill@linuxjournal.com
Senior Editor	Doc Searls doc@linuxjournal.com
Art Director	Garrick Antikajian garrick@linuxjournal.com
Products Editor	James Gray newproducts@linuxjournal.com
Editor Emeritus	Don Marti dmarti@linuxjournal.com
Technical Editor	Michael Baxter mab@cruzio.com
Senior Columnist	Reuven Lerner reuven@lerner.co.il
Chef Français	Marcel Gagné mggagne@salmar.com
Security Editor	Mick Bauer mick@visi.com

Contributing Editors

David A. Bandel • Greg Kroah-Hartman • Ibrahim Haddad • Robert Love • Zack Brown • Dave Phillips • Marco Fioretti • Ludovic Marcotte • Paul Barry • Paul McKenney • Dave Taylor

Proofreader Geri Gale

Publisher	Carlie Fairchild publisher@linuxjournal.com
General Manager	Rebecca Cassity rebecca@linuxjournal.com
Director of Sales	Laura Whiteman laura@linuxjournal.com
Regional Sales Manager	Joseph Krack joseph@linuxjournal.com
Regional Sales Manager	Kathleen Boyle kathleen@linuxjournal.com
Circulation Director	Mark Irgang mark@linuxjournal.com
Marketing Coordinator	Lana Newlander mktg@linuxjournal.com
System Administrator	Mitch Frazier sysadm@linuxjournal.com
Webmaster	Keith Daniels webmaster@linuxjournal.com
Accountant	Candy Beauchamp acct@linuxjournal.com

Linux Journal is published by, and is a registered trade name of, Belltown Media, Inc.
PO Box 980985, Houston, TX 77098 USA

Editorial Advisory Board

Daniel Frye, Director, IBM Linux Technology Center
Jon "maddog" Hall, President, Linux International
Lawrence Lessig, Professor of Law, Stanford University
Ransom Love, Director of Strategic Relationships, Family and Church History Department,
Church of Jesus Christ of Latter-day Saints
Sam Ockman, CEO, Penguin Computing
Bruce Perens
Bdale Garbee, Linux CTO, HP
Danese Cooper, Open Source Diva, Intel Corporation

Advertising

E-MAIL: ads@linuxjournal.com
URL: www.linuxjournal.com/advertising
PHONE: +1 713-344-1956 ext. 2

Subscriptions

E-MAIL: subs@linuxjournal.com
URL: www.linuxjournal.com/subscribe
PHONE: +1 713-589-3503
FAX: +1 713-589-2677
TOLL-FREE: 1-888-66-LINUX
MAIL: PO Box 980985, Houston, TX 77098 USA
Please allow 4-6 weeks for processing address changes and orders
PRINTED IN USA

LINUX is a registered trademark of Linus Torvalds.



TYANPSC™

Hands On Supercomputing



Typhoon™ 600 Series Personal Supercomputer

TyanPSC's Typhoon™, the next generation turnkey Personal Supercomputer has the power to blow away all your computational needs! Purpose-built for office and laboratory environments, easy to deploy and use, the Typhoon™ provides intense computational power in remote or constrained places, works like a PC and is whisper quiet.

High Performance Computing Just Got Cooler

Clusters of Typhoons / Low Power
Small size Form Factor / Under Mobility

Typhoon

T-630 DX / T-650 QX Series

- Up to 186 / 256 GFlops at your desk!
- Turnkey, Easy-to-Deploy, and Easy-to-Use
- Integrated 5 node cluster - up to 20 / 40 processor cores in a box!
- Plugs into standard wall outlet - only uses 15 Amps
- Microsoft® Windows® Compute Cluster Server 2003 pre-installed
- High Performance in Constrained spaces: Office, Remote, Plane, Boat, etc
- RAID capable



Windows Compute
Cluster Server 2003

TYANPSC™

Personal Supercomputer

Tyan Computer USA

3288 Laurelview Court
Fremont, CA 94538 USA
Tel: +1-510-651-8868 Fax: +1-510-651-7688
Pre-Sales Tel: +1-510-651-8868 x5120
Email: marketing@tyan.com

For More Information, please visit
www.tyanpsc.com

Letters



Regarding the Golden Age of *Linux Journal*

After having seen in the past few issues of *LJ* readers who are disenchanted with the magazine's content, I feel I must give counterpoint.

Much like Mr Silverton's setup in the January 2007 issue, I began using Linux in 1995, failing first to install Red Hat and settling in on the astounding Slackware distribution from untold numbers of hit-or-miss floppies. Likewise, I used Linux in a rather amateur fashion for a number of years. At that time, I subscribed to *Linux Journal* but found only one or two articles that rapt my attention. Later, I started a Linux-based company, designing Linux-based solutions for a number of types of businesses. Even after becoming a Linux "professional", I was no more drawn to the overall quality of *Linux Journal*.

For many months now, however, I have been simply astonished by the quality of most every article. I now read the magazine cover to cover. Even the articles that are "beneath" me often have a gem of information that I may find intriguing.

Even more appealing is the uncanny timeliness of recent issue themes. *LJ* has never really been bleeding-edge. After all, it is a monthly publication. Recently, though, I have had several issues in a row that have matched very closely my personal contemporary interests, and I have been delighted by the breadth of your coverage. You cover amply the needs of both the amateur (which I am,

with respect to more desktop-type issues) and the professional (which I am, with respect to the development and server issues).

In summary, I have never, in ten years of subscribership, been so happy with the diversity, scope, depth and accuracy of *Linux Journal*. Keep up the excellent work.

--
Sean C. McCord

What's a Few Zeros between Friends?

Quote from *Linux for Suits*, February 2007:
"The 15mbps they reserve for their Internet service is less than 1% of that capacity."

15 millibits per second really is pathetic.

--
Ian Stirling

Obviously, that should have been Mbps.—Ed.

What's a Few Letters between Friends?

In "White Box Phone" [February 2007], you said that the OpenMoko phone came from Funambol. I think you meant to say FIC.

Funambol is the company behind SyncML and push mobile e-mail.

--
Bill Weinberg

Someone Else May Have to Decipher Your Code Someday

I've already made comments about Dave Taylor's column, and you know I'm behind his doing this column and what it teaches. My comment is not a criticism of that column but what I hope is viewed as an addendum to it.

The urge to put an entire sequence of operations on "one line" (chained together via pipes) is noble and quite understandable when one is "in a hurry" or doing something "quick and dirty". It isn't the wisest of choices for beginners or "semi-production" work.

Yes, this is arguable, but I'd like to reference something else in your magazine, page 14, "They Said It" [February 2007], the third quotation, "Coding up the simplest thing...."

Over time, I've learned that as "The next

guy", trying to read and understand code written by "the guy before me", that breaking a sequence of steps like this down into the constituent parts makes maintainability and supportability significantly easier.

While the original state is "easy enough" to work with, especially for a demonstration, in the work place, we should encourage everyone to remember "the next guy" (and there will always be a next guy) when we write scripts.

I would, in an environment where this might be used by someone else and not run every five minutes, rewrite this script with these additions:

```
# Comment this line out for debugging.
trap "/bin/rm -f $Tmp1 $Tmp2 $Tmp3 $Tmp4"
exit
```

```
Tmp1=/tmp/tmp.1.$$
Tmp2=/tmp/tmp.2.$$
Tmp3=/tmp/tmp.3.$$
Tmp4=/tmp/tmp.4.$$
```

```
grep 'google.com/search' $ACCESSLOG | \
awk '{print $11}' > $Tmp1
```

```
< $Tmp1 cut -d\? -f2 | cut -d\& -f1 >
$Tmp2
```

```
sed 's/+/ /g;s/%22/" /g;s/q=/' < $Tmp2
>$Tmp3
```

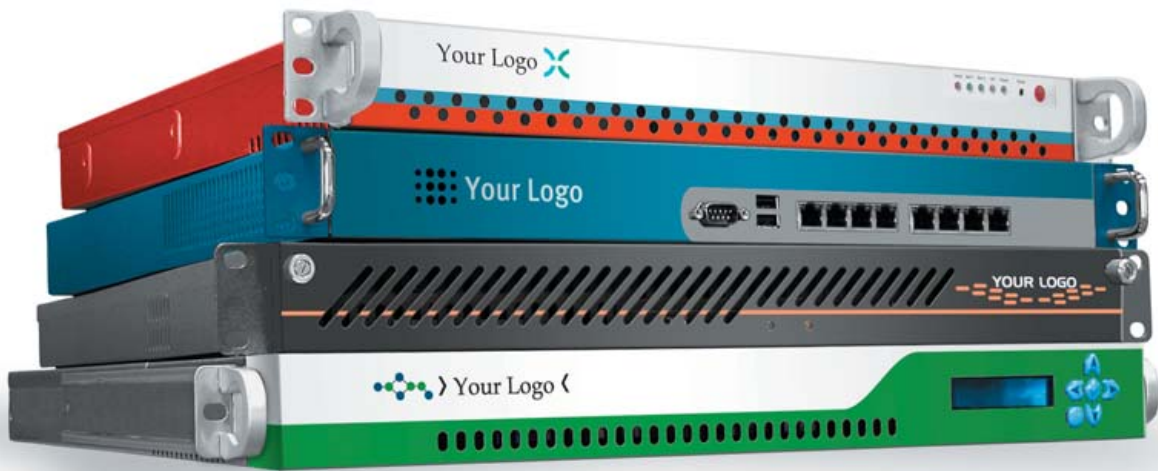
```
sort $Tmp3 | uniq -c > $Tmp4
```

```
sort -rn $Tmp4 | head -5
```

Yes, this seems wasteful. (I mention the five minutes above to point out that if run more than, say every half hour on a moderately busy box, the methods may need to be adjusted.) Many people will point out a number of issues with this method of operation, and one of them is going to be "writing to disk so much wastes a lot of time". Yes, it does. I've never been a member of the "save the PID" crowd or the "make it hard to troubleshoot for job security" group either. Writing to disk a lot is much cheaper than the rate I get paid. If it takes an extra 20% to run, who cares as long as it gives the right answer reliably?

Using the sequence of steps I show above,

The Industry Leader for Server Appliances



Custom server appliances or off the shelf reference platforms,
built with your image and software starting under \$1,000.

From design to deployment, we handle it all.

Delivering an appliance requires the right partner. MBX Systems is the right partner. We understand that by putting your name on our hardware, you're putting your reputation in our hands. We take that seriously. We provide the services you need to support your customers. Better than the competition. You never even

need to touch the hardware. Engineering. Design. Deployment. We handle it all, so you can focus on what's important to you. Your software. Your sales. Your success.

Visit us at www.mbx.com or call 1-800-681-0016 today.

MBX
systems

www.mbx.com | 1-800-681-0016

NEW!

Tiny WiFi Controller
boots Linux in 1.1 seconds



\$129
CPU board only

\$249
as shown

quantity
discounts
start at
10 units

200 MHz CPU

- TS-7400 CPU board
- Low power, low heat, long life
- Up to 128MB on-board Flash
- Up to 128M SDRAM
- SD Flash Card socket
- 1 external USB port
- 1 10/100 Ethernet
- 802.11g internal WiFi option
- One piece, rugged aluminum enclosure option measures 1.1" x 4.9" x 3.1"

Design your solution with
one of our engineers

- Over 20 years in business
- Never discontinued a product
- Engineers on tech support
- Custom configurations and designs w/ excellent pricing and turn-around time
- Most products stocked and available for next day shipping

See our website for options,
peripherals and x86 SBCs



We use our stuff.

visit our TS-7200 powered website at

www.embeddedARM.com/7400wifi

(480) 837-5200

[LETTERS]

the people who have to figure out what the he** I did can quickly get to the root of the problem. Now, all they have to do is comment out the trap line and look at the temp files' output to see what is happening and how to fix it. The trap line executes the rm of the temp files when the script exits (that is, it cleans up after itself). They also can see the flow of the process better and gain quicker understanding.

Thank you for the great articles and helpful insights!

--
Michael C. Tiernan

Now You See Them, Now You Don't

In */var/opinion* [February 2007], Nick Petreley suggests tactics to move further into the mass desktop market. Among other things, he says, "don't remove features".

I heartily agree. KDE has seen a steady stream of added features, but unfortunately some old ones were removed. It is no longer possible to move a maximised window partly out of the viewport. You have to de-maximise it first, just like Microsoft. Nor can you double-click on the bar at the top and reduce it to just the bar, as used to be possible. Konqueror no longer allows you to right-click on a file and see an option to delete it, merely one to move it to trash. (Fortunately, there is a way to do this: select the file and press Shift-Delete.)

Presumably, the strategy is to converge on Microsoft—up to a point. If desktop-level Linux becomes identical, why should people buy it rather than the ready-installed Microsoft product?

When Microsoft moved from command-line (MS-DOS) to Windows, Linux expanded in the same direction, retaining the old facilities. This is part of its strength. The same logic applies to desktop Linux: if we remove features, we play to lose.

--
Chris Trayner

As far as I can tell, you're right about moving files to the trash. But, I just moved a maximized window partially out of the virtual desktop, and I also shaded (reduced a window to its title) by double-clicking on the title bar. I obtained KDE 3.5.6 from a KDE mirror and run it on Ubuntu 6.10 AMD64.—Ed.

LINUX JOURNAL

At Your Service

MAGAZINE

PRINT SUBSCRIPTIONS: Renewing your subscription, changing your address, paying your invoice, viewing your account details or other subscription inquiries can instantly be done on-line, www.linuxjournal.com/subs. Alternatively, within the U.S. and Canada, you may call us toll-free 1-888-66-LINUX (54689), or internationally +1-713-589-2677. E-mail us at subs@linuxjournal.com or reach us via postal mail, Linux Journal, PO Box 980985, Houston, TX 77098-0985 USA. Please remember to include your complete name and address when contacting us.

DIGITAL SUBSCRIPTIONS: Digital subscriptions of *Linux Journal* are now available and delivered as PDFs anywhere in the world for one low cost. Visit www.linuxjournal.com/digital for more information or use the contact information above for any digital magazine customer service inquiries.

LETTERS TO THE EDITOR: We welcome your letters and encourage you to submit them to ljeditor@linuxjournal.com or mail them to Linux Journal, 1752 NW Market Street, #200, Seattle, WA 98107 USA. Letters may be edited for space and clarity.

WRITING FOR US: We always are looking for contributed articles, tutorials and real-world stories for the magazine. An author's guide, a list of topics and due dates can be found on-line, www.linuxjournal.com/author.

ADVERTISING: *Linux Journal* is a great resource for readers and advertisers alike. Request a media kit, view our current editorial calendar and advertising due dates, or learn more about other advertising and marketing opportunities by visiting us on-line, www.linuxjournal.com/advertising. Contact us directly for further information, ads@linuxjournal.com or +1 713-344-1956 ext. 2.

ON-LINE

WEB SITE: Read exclusive on-line-only content on *Linux Journal's* Web site, www.linuxjournal.com. Also, select articles from the print magazine are available on-line. Magazine subscribers, digital or print, receive full access to issue archives; please contact Customer Service for further information, subs@linuxjournal.com.

FREE e-NEWSLETTERS: Each week, *Linux Journal* editors will tell you what's hot in the world of Linux. Receive late-breaking news, technical tips and tricks, and links to in-depth stories featured on www.linuxjournal.com. Subscribe for free today, www.linuxjournal.com/newsletters.



Your integrated neighborhood
just got friendlier



With Pogo's Network Attached Storage appliance, maximum interoperability with the most popular operating systems is easier. Take control of your ever-expanding storage needs with an easy to use graphical management interface, multiple snapshots, built-in back up, IP failover, and enhanced security. Powered by Intel Dual-Core Xeon processors and a fast 64-bit OS. Call today for more information.

NAB2007

THE WORLD'S LARGEST ELECTRONIC MEDIA SHOW

APRIL 14-19, 2007

LAS VEGAS

Visit us in the North Hall, Booth #3938

**FREE VIP
EXHIBIT PASS
CONTACT POGO LINUX**

www.pogolinux.com

pogo [linux]TM

Experience, Imagination, and Support.

Pogo Linux is Hardware Solutions Built for Linux, Built for You.

To get started, contact us at 888.828.POGO or inquiries06@pogolinux.com

Pogo Linux, Inc. 701 Fifth Ave. Suite 6850, Seattle, WA 98104



Intel, Intel logo, Intel Inside logo, Pentium, Xeon, and Xeon Inside are trademarks or registered trademarks of Intel corporation or its subsidiaries in the United States and other countries. For additional terms and conditions please visit www.pogolinux.com

diff -u

WHAT'S NEW IN KERNEL DEVELOPMENT

The kernel may soon support a **larger compiled binary size**, if a few remaining problems can be sorted out. **Eric Biederman** has had some patches to accomplish this floating around for a while, and **Vivek Goyal** at IBM has been testing them out. According to Vivek, the time is now ripe to give Eric's patches (with modifications and Vivek's own fixes) to a wider audience and include them in **Andrew Morton's** -mm tree. A lot of folks seem happy to see these patches, and much additional work is going into them from various kernel people. Some of that work is in the form of cleanups, but some folks also are working on the tricky interactions between this code and the **swsusp software suspend code**. It does seem as though these patches, sooner or later, will be complete and accepted into the official tree.

The open-source **ar5k wireless driver** has been cleared of any copyright problems, at least according to the **Software Freedom Law Center** (SFLC). The driver, developed for OpenBSD, primarily by **Reyk Floeter**, had lived under a cloud of suspicion by **Atheros**, the maker of the wireless chipsets it supported. And, although nothing was ever proven against the ar5k developers, the "scandal" was enough to prevent Linux folks from incorporating the driver into the Linux kernel. With the SFLC opinion, the worst objections have been set aside, if not entirely eliminated, and now the biggest complaint from kernel folks is that the ar5k driver may just be badly writ-

ten or take the wrong approach. Currently, some folks support incorporating ar5k into the kernel in its current form or with slight modifications, and some think the whole driver should be scrapped and just used as a hardware reference for an entirely new driver. Whichever way that debate falls, it's clear that Atheros wireless chipsets will soon have a free alternative to Atheros' own freely available though proprietary MadWifi driver.

A lot of patches have come along to remove broken or unmaintained parts of the kernel or to schedule their removal after a period of depreciation. Andrew Morton has scheduled **FUTEX_FD** for removal in June 2007, saying the code has "unfixable races"—exactly the charge leveled against DevFS in the old days. **Rusty Russell**, who originally wrote the patch, has no interest in fighting to keep the code. Moreover, he says that once it's removed, the futex code could be made a lot simpler. With friends like these, FUTEX_FD needs no enemies, and it is doubtful the code will last even as long as Andrew has given it.

At the same time, **Adrian Bunk** has posted patches to remove a bunch of old drivers that have been marked as broken for several years. These include the **VIDEO_ZR36120** and **SKMC** drivers; and the **MAC89x0**, **ATARI_BIONET** and **ATARI_PAMSNET** drivers. Of these latter, **Geert Uytterhoeven** may have a patch, by **Matthias Urlichs**, to fix the MAC89x0 driver. But the patch still needs to be tested and signed off on. Meanwhile, Adrian has posted more patches to remove the **FB_CYBER FB_VIRGE**, **FB_RETINAZ3**, **FB_ATARI**, **FB_SUN3** and **FB_PM3** drivers,

but it looks as though **James Simmons** might take over maintainership of those, and Geert may already have a patch to fix the FB_ATARI driver. So, those framebuffer drivers may not be taken out after all.

At the same time, the **sysctl code**, which to all appearances was on the chopping block for real, seems to have found a reprieve. Eric Biederman, who had scheduled the code for removal, found some legitimate sysctl users. As a result, not only is the code no longer marked as deprecated, but it also will be compiled into all kernels by default for the near future. Eric will work with the various distributions to eliminate the sysctl uses gradually and, in perhaps a year or two, will once again consider removing the code.

The **Sparse C code parser** never had an official release under its original maintainer, so it becomes one of a very small group of projects to have an initial release only after changing hands. **Linus Torvalds** originally created the tool in 2003 for his own use to help spot bugs in kernel patches. And, he made the tool available for download without ever giving it a version number or doing any kind of organized release. When users started asking for features that Linus didn't himself desire, he suggested that someone else might take over the project. **Josh Triplett** stepped up, officially releasing Sparse version 0.1 with several feature enhancements. A version 0.2 followed closely after that, this time mainly adding bug fixes. Is the project old, having begun in 2003, or new, having just put out its first release? You decide.

—ZACK BROWN

Governments Vote for Linux Security

Linux is the fastest growing operating system in the world, in large part because customers in every industry are demanding highly secure information technology environments, particularly those customers in the public sector.

Worldwide, more than 225 IBM government customers are embracing Linux to lower the total cost of computing, consolidate workloads, increase efficiency and enact e-government transformation. But, it's the inherent higher level of security that is pro-

viding government agencies the confidence to expand Linux beyond edge-of-network applications into the heart of the enterprise to run mission-critical applications.

Today, thousands of developers and members of the open community worldwide work on Linux, constantly making it better and more secure. IBM, for instance, supports the growth of Linux through the work of IBM's Linux Technology Center, made up of more than 600 engineers in 40 locations worldwide, of whom more

than 300 work full-time on Linux as part of the Open Source community.

Open Community Committed to Security

The inherent Linux security advantage is the added layer of community approval, which closed systems are unable to provide as they rely completely on internal teams to avert security breaches. Only after close examination and approval by a body of peers does a Linux solution reach

a customer. This same talent pool quickly addresses emerging security concerns, plugging holes and releasing patches before a company's infrastructure is ever compromised. With other operating systems, bug fixes and security patches are more likely to be lumped together and released based on specific timelines that accommodate the vendor, not the users.

But as e-citizen applications become the norm, the users rule the roost. Citizens are becoming more accustomed to self-service applications, such as renewing a driving license on-line or paying a citation. As the number of citizens accessing information on-line grows, so does the incidence of cyber crimes. Crackers, phreakers and identity thieves keep abreast of technology advances, increasing their sophistication.

This now poses significant new threats to governments where the chief concern includes safeguarding national security and information privacy. As citizens share sensitive information, including names, date of birth, addresses and social security numbers with e-government systems, there needs to be a check in place to ensure this information is guarded by the highest levels of security.

As a result, advances in Linux security continue to improve, especially in the affordability of implementing higher-level security architectures. Open-source software provides more transparency and user control, allowing users to identify and fix security vulnerabilities as and when they happen, as compared to waiting for vendors to fix the security flaws.

Common Criteria—Early Standard for Security

Many government agencies worldwide require IT vendors to adhere to Common Criteria security certification standards. The Common Criteria is recognized internationally by IT professionals as the ISO standard (ISO/IEC 15408) used by the United States government and other organizations to assess security and assurance of technology products. The world's largest Linux enterprise server distribution vendors, Novell and Red Hat, for instance, have achieved high levels of security certification that enable Linux to be adopted by governments and government agencies running on multiple IBM software and hardware platforms.

UK Cabinet Office Pioneers Security-Enhanced Linux

Advances in security continue to improve, especially in the affordability of implementing higher-level security architectures. For use within governments, one resource has been the use of Security-Enhanced (SE) Linux. The Mandatory Access Control provides access to information on a need-to-know basis, protecting governments better from potential cracker and virus threats.

IBM has been a key enabler, consistently supporting the growth of SELinux while simultaneously ensuring that security within the platform is hardened, maintained and enhanced without compromising access and ease of use. IBM recently worked with the UK Cabinet Office and partners Tresys and Belmin Group on the first pilot for Mandatory Access Control through Security-Enhanced Linux in an e-procurement application at a National Health Service hospital trust in England. Through the use of this technology, any organization will have the ability to contain crackers, provide the necessary confinement for its applications and minimize damage to the enterprise.

For years, customers have demanded secure, open, interoperable platforms that are easier to manage and more flexible in running different workloads in a heterogeneous architecture that often comprises disparate products from multiple IT vendors. This trend is particularly pervasive in emerging markets.

India: A Shining Star for Linux and Open Source

India has been a shining star for open-source Linux software, with growth rates for sales doubling in the past year alone. The government and defense sectors have been one of the key drivers for Linux adoption in India, which combined accounted for 38% of the Linux market during 2005–2006 according to Dataquest. In India, customers are flocking to Linux because it provides a low-cost, highly secure and customizable alternative to closed systems and vendor lock-in.

Just this month (January 2007), it was reported that Tamil Nadu, one of the largest and most industrialized states in India with a population the size of the UK, is moving away from a closed system in favor of open source. They also are training 30,000 government officials on Linux, a key move because

technical skills have been one of the barriers to further adoption of open-source software. Recently, the state of Kerala, which boasts a 100% literacy rate, announced it was migrating to Linux as well.

Project Higgins and User-Centric Management

With the growth of on-line government services, citizens are benefiting from new open-source software security initiatives that are aimed at helping protect their personal information. IBM, Harvard Law School's Berkman Center for Internet and Society, Novell and Parity Communications are working on an open-source initiative, code-named Higgins, that will spawn a new generation of security software, giving people more control over personal on-line identity information.

The Higgins initiative is developing software for user-centric identity management, an emerging trend in security software. User-centric identity management enables individuals to manage and control their on-line personal information actively, such as bank account numbers, medical records, telephone numbers and credit-card numbers—rather than institutions managing that information as they do today. People will decide what information they want shared with trusted on-line Web sites that use the software.

The user will play a key role in the future in helping enterprises and institutions comply with various e-governance regulations. As users have more and more access to e-citizen services, user-centric management will help in identifying the information individuals want to share, for instance to their health-care provider. A health-care provider does not necessarily need to know the user's personal information and will have access to information relevant to the patient's medical history, thus limiting the exposure to data loss.

The Higgins initiative moves on-line security to the next stage by creating an open, highly secure and flexible software platform that essentially puts the user in the center of the identity management cosmos.

Linux is continuing to become an unstoppable force in the public sector, helping provide government agencies and the citizens they serve worldwide a computer operating system both can trust.

—DANIEL FRYE

Just Say No to OpenXML

A couple of months ago, Microsoft and another software company, Novell, signed a technical cooperation agreement. A part of this agreement may turn out as highly dangerous for almost all citizens, but it is not the one that the majority of critics addressed.

Summary of the Previous Season

With this agreement, Microsoft commits not to sue, for patent infringement and other “intellectual property” violations, the users of the specific version of Linux packaged and distributed by Novell. One second after the announcement, Linux and Free Software supporters worldwide started to explain to each other how absurd and laughable it would be to believe, even for a single moment, that such violations actually exist or that any court could ever sentence Linux end users for them. This week, a handful of public statements (not even from Microsoft) quickly showed how little such arguments matter in the real, nongEEK world. Wal-Mart’s Chief Technology Officer announced that the company finally will be able to use Linux to expand its global Web presence, because “questions over intellectual property are a ‘huge problem’”, but “the intellectual property protections in the Novell deal give Wal-Mart more confidence in using Linux more broadly”. After such a confirmation from Wal-Mart, which business owner (or Wal-Mart supplier) is going to listen to *geeks* swearing that Linux has no hidden legal bombs?

Regardless of Wal-Mart, the worst, most dangerous part of this deal may not be the patent suits part, but something else with an even bigger impact on the culture, economy and ownership of public documents.

Interoperability, Please!

Interoperability is enormously important in the computer world. If the documents produced with a computer program are not completely and surely readable with any other program of the same category, those documents cannot be exchanged, become unusable after only a few years or remain available only by paying much more than would be fair to the producer of the original software.

In order to avoid this, it is necessary to store and exchange documents in a nonproprietary file format. As explained in “Everybody’s Guide to OpenDocument”, “if computer programs are pens, then think of file formats as alphabets. There is nothing wrong in selling overpriced pens, as long as

Resources

Statement from Wal-Mart CTO:

news.com.com/Wal-Mart+eyes+Microsoft+for+Web+build-out/2100-1017_3-6152247.html

Why Redmond feels so threatened by ODF:

computerworld.co.nz/news.nsf/tech/CBE417F838EAB4A0CC25717A001A0EC9

“Everybody’s Guide to OpenDocument”: www.linuxjournal.com/article/8616

Introduction to OpenDocument: opendocumentfellowship.org/introduction

How to hire Guillaume Portes: www.robweir.com/blog/2006/01/how-to-hire-guillaume-portes.html

Novell statement on file formats for office applications: www.novell.com/products/desktop/fileformats.html

Is Office OpenXML A One-Way Standard? Ask Microsoft: blogs.adobe.com/shebanation/2006/12/open_xml_one-way.html

Is OpenXML a one-way specification for most people? www.sutor.com/newsite/blog-open/?p=1145

A game of Zendo: www.robweir.com/blog/2006/07/game-of-zendo.html

Microsoft Office to get a dose of OpenDocument:

news.com.com/Microsoft+Office+to+get+a+dose+of+OpenDocument/2100-1013_3-6069188.html?tag=nefd.top

OpenDocument Viewer: opendocumentfellowship.org/odfviewer

cheap models also exist. But the whole thing is contingent on everybody using the same alphabet, without needing to pay fees or learn special secrets.”

Today, two file formats are competing for all our office files. One is the nonproprietary, completely open by design, internationally ratified standard called OpenDocument. The other is OpenXML, the format used in the next version of Microsoft Office, which aims to reach the same status but already has been defined by several experts as something explicitly created to be usable only in Microsoft Office.

Here Comes the Real Trap

In spite of all this, the file format part of the agreement explicitly says that it is “designed to ensure that customers using OpenOffice.org will continue to be able to read and write documents using future Microsoft Office file formats...[Novell will] ensure that file formats used by future versions of Microsoft Office are well

specified and available to all to implement.”

Can you see the problem now? First, the agreement lasts five years—it cannot ensure anything after 2011. Next, it is officially *not* meant to make OpenDocument usable in Microsoft Office. Instead, it is going to make sure that OpenOffice.org users can continue to be slaves to a proprietary format that is very hard to support completely in other software programs.

99% of existing office files are Microsoft-locked, and almost all current desktops are Microsoft. The agreement will actively work to preserve this situation, practically making OpenDocument (and the very concept of nonproprietary formats) irrelevant and unused in any large organization, no matter what its technical and openness merits and certifications are.

Think of all the large companies and public administrations where most of the existing partners, customers or suppliers use Microsoft

SUPERMICRO®

Blue Skies Green Earth

...And We Want to Keep It that Way.



Help Preserve the Earth with Supermicro High Efficiency Servers

Supermicro continues to lead the way in server technology with power-saving, earth-friendly concepts in its full line of rackmount and tower servers with high-efficiency power supplies as a standard option.

With high efficiency ratings reaching an impressive 90%+, Supermicro power supplies can save up to 30% energy per system compared to competitive traditional power supplies. This can equal up to \$300 savings over a 3 yr. period.

1U Twin™
2 Nodes in 1U
Double Density
90%+ Efficiency Power Supply

www.supermicro.com/products/nfo/1UTwin.cfm

Already a Supermicro Customer?
Ask us for your Supermicro Earth
Friendly Lapel Pin.



Office formats. Laziness and the wish to ignore what software is are very powerful. In such a scenario, the first time a manager sends an OpenDocument file from an employee or supplier back with an "I can't open this" note, the sender will set the default file format of OpenOffice.org to OpenXML and never go back. Why compromise a career or a sale annoying people in this way, especially if "I can still use this cool free software, can't I?" Sure—until the agreement expires and the next version of OpenXML breaks compatibility.

This is the real danger; there won't be any need to sue anybody for using Linux, because millions of business and public files will remain in a one-way format, made to order for Microsoft Office.

This is not interoperability. OpenDocument is interoperability. Working on OpenXML support in OpenOffice.org also makes it look like multiple applications *are going* to support this format. It is a prerequisite to ISO standardization—that is, for government acceptance. Government pressure to require/migrate to OpenDocument also would be much easier to fight or ignore: "Why spend public money on training, software migration and so on, now that our (current) format is an ISO standard and even Linux can open the formats of 99.999% of the existing office files successfully? See how good we were to encourage competition and give you choice?" All this could sensibly delay the adoption of OpenDocument (not forever, don't worry: just four or five years). It wouldn't be the first time that something goes into limbo because Microsoft says that it will support it in some way and then drops it because "there is no market demand".

The Solution

Even if you don't care at all about computers, OpenDocument is an occasion too important to miss—to save tax money if for nothing else. Please refuse to distribute or accept office files in OpenXML formats. It's too risky. Use whatever software you like best, but just say no to OpenXML, and ask your friends, coworkers and government to do the same. This is going to be immensely easier to do than in the past. A plugin to read and save OpenDocument files in Microsoft Office and a cross-platform viewer (similar to an Acrobat Reader for OpenDocument) are being actively developed. Very soon, there won't be any valid excuse not to use OpenDocument. Don't miss this opportunity.—MARCO FIORETTI

Widening the Analog Hole



The entertainment industry embraced the digital revolution by making digital goods behave like analog ones—that is, scarce and hard to reproduce. It narrowed reproducibility of its "content" until all that remained was what it ironically called "the analog hole". Only through the analog hole could scarified digital goods still be moved with relative ease, and without being stopped by the DRM police. The moved goods would not be of identical digital quality, but they risked looking and sounding good enough to please the user.

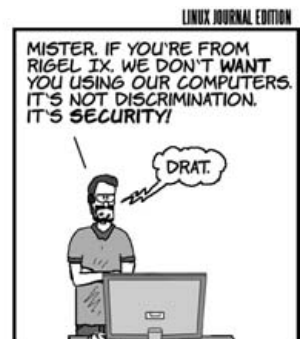
Naturally, user demand has turned the analog hole into a marketplace. Front and center are the products by Chicago-based Neuros, which loves Linux and works to bring the free and open Linux value system to the world of media production and reproduction.

The Neuros Recorder 2 is an MPEG-4 recorder that "works like a mini-digital VCR". It connects to sources over RCA cables and records in real time on standard removable Flash memory cards. Content can be transferred to other devices manually or over a USB 2.0 connection.

The company's latest product is the Neuros OSD, "the first open-source Linux-based embedded media center". It lets you record from cable, satellite TV, DVD, TiVo/DVR, camcorder or VCR over RCA or S-Video cables, and to distribute or play back recordings over LAN (Ethernet), memory card, USB or RCA cables. The architecture is wide open, and it's constantly improved by Neuros and a growing community of developers and users. The price is \$229 US.



Check them out at neurostechnology.com.—DOC SEARLS



OPPORTUNITIES IN LEADERSHIP COMPUTING

The Oak Ridge National Laboratory in East Tennessee is engaging a world-class team to make dramatic advancements — fielding new capabilities and new opportunities for application to high-end science and high performance computing.

Solving Problems Beyond the Reach of Today's Technology

The science of the 21st century demands computational capability well beyond what is available today. Oak Ridge National Laboratory (ORNL) and the Department of Energy's Office of Science intend to lead the world in computational science and engineering as a tool for frontier scientific discovery to deliver new insights and significant breakthroughs with far-reaching impact for U.S. scientific leadership, industrial competitiveness, and national security.

ORNL will be home to a wealth of resources to accomplish these goals:

- **National Center for Computational Sciences (NCCS)**
- **DOE Leadership Computing Facility (LCF)**
- **Advanced Scientific Computing Research (ASCR)**
- **Regional Cyber Infrastructure for Science**
- **Classified Computing (for national security and energy assurance)**



Leadership Computing Capabilities

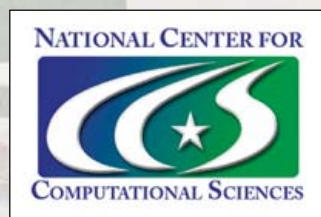
Computing power is key to scientific leadership in basic energy sciences, biological and environmental sciences, fusion energy, and high-energy and nuclear physics. The high performance computing platforms housed within the NCCS are vital to research programs and will provide solutions to many of our most pressing national challenges.

These capabilities will make ORNL the nation's most powerful open resource for capability computing, with a sustainable path that will maintain and extend national leadership for the Federal government and the scientific community at large, and will include computer science, mathematics, modeling, simulation, knowledge extraction and discovery, and numerical methods.

Join the Oak Ridge National Laboratory

The continued success of the nation requires world-class leadership in terascale and petascale computing. ORNL is in search of world-class scientific and technical staff and currently has opportunities in the following areas:

- **System Programming**
 - kernel
 - networking
 - parallel file system
 - hierarchical storage
- **Early State Research (new computing technologies and architectures)**
- **Design, Development, Maintenance, and Operation of IT Infrastructure**
- **Compiling, Debugging, and Next-Generation Tools Development**
- **Project Management and Controls**
- **System Administration and Operation**
- **HPC User Support and Outreach**
- **Scientific Applications R&D**
- **Future Computing Architectures**
- **Performance Analysis**



To learn about and apply for these and other related opportunities, please visit <http://computing.ornl.gov/Employment>. For inquiries or more information, send e-mail to CCSD_Staffing@ornl.gov.

ORNL, a multiprogram research facility managed by UT-Battelle, LLC, for the U.S. Department of Energy, is an equal opportunity employer committed to building and maintaining a diverse work force.

More of Less

If you're looking to put together combinations of small low-cost computing systems, it's hard to overlook the goods coming from e-Way, an American-run Taipei-based company. e-Way specializes in hardware that sells—in single units or in quantities—at prices you could cover with the average ATM withdrawal. They have a variety of tiny CPU boards and boxes ranging from \$90–\$189 US, plus screens and other components at prices as low as \$8 US. At the Consumer Electronics Show, e-Way President Steve Freiburger even ran a slideshow off an extremely compact desktop Linux. The distro and 50 apps—including the slideshow software—all fit inside a 128Mb CompactFlash memory card. Check them out at ewayco.com.—DOC SEARLS

The Market Aperture Opens

Cinema-grade 35mm film cameras always have been brutally expensive, and the same goes for their digital counterparts. One reason is just that they're complex and expensive to produce. Another is that the manufacturers always have maintained a distance between castes of cus-



tomers. Professional gear not only has features and abilities far beyond those of "consumer" gear, but it is far more expensive as well.

This is starting to change with video gear. The RED digital camera (red.com) is the first professional video camera that makes 35mm-grade cinematography available in digital form at prices independent producers can afford. It shoots with a 4,520 x 2,540 resolution (2,540 progressive) at 60 frames per second RAW, using a 12-megapixel Mysterium CMOS sensor. It's flexible, format-agnostic and costs

\$17,500. For something this good, that's cheap.

Linux has become the platform of choice for much of Hollywood's cinema production. Does the availability of cameras like RED's beg for more Linux in the rest of the cine production world? We'll see.—DOC SEARLS

They Said It

Make no little plans; they have no magic to stir men's blood and probably will themselves not be realized. Make big plans; aim high in hope and work, remembering that a noble, logical diagram once recorded will not die.

—Daniel Burnham

The best way to get money isn't to find some VCs to beg, borrow, or steal from; the best way to get money is to make something people will pay for.

—Giles Bowkett, gilesbowkett.blogspot.com/2006/12/tale-of-two-startups.html

Most embedded device vendors don't release any source. Then someone from the community nags the vendor's legal department for six months and eventually gets a partial source tree that doesn't compile.

My impression is that companies treat legal compliance not as a hard requirement but as a risk management exercise; if (cost of GPL lawsuit) * (probability of GPL lawsuit) < (cost of VxWorks license) * (number of units), use Linux.

—Wes Felter, www.linuxjournal.com/comment/reply/1000164/209612

Rosenberg's Law: "Software is easy to make, except when you want it to do something new."
Rosenberg's Corollary: "The only software that's worth making is software that does something new."

—Scott Rosenberg, from *Dreaming in Code*, Crown Publishing, 2007

Open sourcing is the most important decision we've made in seven years of Second Life development. While it is clearly a bold step for us to proactively decide to open source our code, it is entirely in keeping with the community-creation approach of Second Life.

—Cory Ondrejka, CTO of Linden Lab

If there isn't enough food in the fridge, do you say "the store must be down?"

—Greg Elin, at a conference

1. Millions of Second Life Residents as of January 1, 2007: **2.287**
2. Thousands of US dollars spent per day in Second Life as of January 1, 2007: **803.79**
3. Days into 2007 when Linden Lab opened the source code for the Second Life Viewer: **8**
4. Billions of US dollars of consumer electronics sales in 2006: **145.7**
5. Percentage of surveyed German residents who read blogs: **15**
6. Percentage of surveyed German "influencers" who read blogs: **27**
7. Percentage of surveyed US residents who read blogs: **27**
8. Percentage of surveyed US "influencers" who read blogs: **34**
9. Percentage of surveyed Japanese residents who read blogs: **74**
10. Percentage of surveyed Japanese "influencers" who read blogs: **91**
11. Percentage of all blogs that are in English: **39**
12. Percentage of all blogs that are in Japanese: **33**
13. Percentage of US women living without a spouse in 1950: **35**
14. Percentage of US women living without a spouse in 2000: **49**
15. Percentage of US women living without a spouse in 2005: **51**
16. Years since Jabber source code was first released: **8**
17. Range in millions of users of open-source XMPP (Jabber) technologies: **40–50**
18. Number of dual-processor PCs running Linux at Tradebit AG: **10**
19. Terabytes of data served by Tradebit AG: **20**
20. Millions of number of downloads per day from Tradebit: **1**

Sources: 1, 2: Tristan Louis | 3: Linden Lab
4–10: Edelman | 11, 12: Technorati
13–15: *New York Times* | 16, 17: XMPP.org
18–20: Tradebit AG

—Doc Searls

Do you really want to do anything on your own?

Or just get it off the shelf!



ElinOS

INDUSTRIAL GRADE LINUX

ElinOS Industrial Grade Linux – Off the shelf Embedded Linux.

Embedded Linux, with its feature richness and royalty free usage, has gained a massive growth in popularity over the last years. But quality assurance, reproducibility, and version stability are challenges project managers always have to face. This consequently leads to the question "Make or Buy?" ElinOS Industrial Grade Linux gives the answer.

ElinOS Industrial Grade Linux at a Glance

Versatile Embedded Linux

- Industrial Grade
- Available for PowerPC, x86, MIPS, ARM, XScale
- Kernels 2.6.15 and 2.4.31

Application Development

- Integrated development environment (IDE)
- Quickstart configuration editor
- Out-of-the-box experience

Service and Support

- Peer-to-peer development and lifecycle support

Visit us at:

**Embedded
Systems** CONFERENCE
SILICON VALLEY

Make USB/MIDI work, turn your existing soundcard into a high-quality synthesizer, and exploit the power of X.

»» How to Make Your USB/MIDI Connection Work with Your Custom-Compiled Kernel

Many of us like to compile our own kernels. In my case, I compile my own kernels for two reasons. First, stock kernels include `initrd` images that tend to discover my SCSI devices in the wrong order. You can fix the `initrd` to discover devices properly (I explained how to do that with Ubuntu/Kubuntu in a previous installment of Tech Tips). But, I prefer to avoid the problem by compiling the drivers into the kernel rather than loading them as modules. Second, I just like to run the latest stable kernel available.

I've been playing with synthesizers with a USB/MIDI connection. Much to my dismay, I couldn't seem to make the USB/MIDI connection work with my own compiled kernel. I couldn't find any information on the Web that would point me to the problem, but I eventually stumbled on the answer quite by accident.

The solution turns out to be quite simple and obvious once you look through enough of the kernel configuration options. One problem is that the kernel drivers are organized in such a way that it wasn't obvious (at least it wasn't obvious to me) which drivers to include to make this work. I picked all the MIDI sequencer drivers, so why wasn't it working? The USB driver you need actually resides within the tree for sound drivers, not USB drivers. Select the following from "make menuconfig" (or whichever method you prefer for kernel configuration): Device Drivers→Sound→Advanced Linux Sound Architecture→USB Devices→USB Audio/MIDI driver.

Although, as I said, I generally compile my drivers directly into the kernel, I recommend that you compile this particular one as a module, instead. There's no point in having the module loaded during those times when you're not using the synthesizer keyboard.

Select the above driver as a module, and it creates a module called `snd-usb-audio`. The module name was the source of my confusion. I found the `snd-usb-audio` module when I tried to track down what made the stock kernel work, but I dismissed this module as a possible candidate for fixing this problem due to its name. It didn't occur to me that `snd-usb-audio` had anything to do with MIDI until I stumbled across the label "USB Audio/MIDI" in the kernel configuration. The module name itself makes it seem like the module is meant for an external sound source, not an external MIDI source.

By the way, I was inspired to set this up after getting my daughter a Korg X50, a very affordable and excellent synthesizer keyboard. The latest Korg keyboards don't seem to require any special configuration in order to connect the USB/MIDI port. However, I later discovered that the Yamaha keyboards aren't quite as friendly. You have to change some MIDI settings on the Yamaha Motif ES keyboards to make the keyboard work with the computer via the USB port.

This may seem self-evident, but the trick is to follow the instructions in the Yamaha Motif ES manual for connecting the USB/MIDI to the computer. Well, duh, right?

RTFM, or more politely, read the fine manual. But when you encounter problems, it's sometimes tempting to look for tips on the Web to make the keyboard work. Be warned that you should *not* follow many of the instructions you'll find on the Web. These instructions are generally for Windows and the Mac, and they'll tell you how to configure the Yamaha keyboard to send the MIDI signals through the computer and echo them back to the keyboard. It's probably possible to set up the Linux driver and/or patch daemons (such as `jackd`) to make Linux applications work with this configuration, but that's not how Linux behaves by default. So, this is definitely a case where you should avoid the Web and RTFM instead.

—Nicholas Petreley

»» Turn Your Computer into a High-Quality Synthesizer

Maybe you can't afford even the Korg X50, but you want to try your hand at composing music or even just playing MIDI files. The problem is most soundcards that work with Linux do not come with a very impressive collection of MIDI sounds (such collections are usually referred to as soundfonts). Free soundfonts are available, but they don't sound as professional as some of the ones you can purchase. For example, SONI VOX MI sells a fantastic General MIDI (GM) soundfont, and it's available for just under \$100 US. If you want to use a keyboard to record MIDI sequences, you can purchase one of many cheap MIDI keyboards that do nothing but send MIDI signals (they have no synthesizer included). These keyboards sell for well under \$100 US depending on the quality that you'll find satisfactory.

Here's how to use the SONI VOX soundfont. First, purchase the font from the URL listed below. It has been too long since I've purchased my copy for me to recall whether the file you download is a ZIP file or a Windows executable. Even if it's a Windows EXE file, you should be able to unpack it with Wine.

Now, download and install `fluidsynth` and the `Qsynth` front end (it's as simple as an `apt-get install fluidsynth qsynth` from Debian and many Debian-based distros). You may have to load the ALSA sequencer drivers manually or specify the module in a file like `/etc/modules`. The module you want to load is `snd-seq`, and the command to load it is `modprobe snd-seq`.

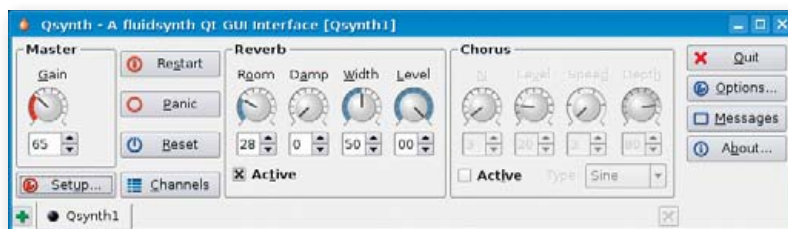


Figure 1. The Qsynth Front End to fluidsynth

In addition, just playing with X in this way makes you understand the interrelationships between X, your window manager and your applications.

Start up Qsynth, and you'll see a window like the one shown in Figure 1.

Press the Setup button. You may have to configure Qsynth with the MIDI and Audio tabs depending on the distribution and setup you use.

Now, click on the Soundfonts tab (Figure 2), and click the Open button to navigate to the SONiVOX soundfont you downloaded and installed. Click the Open button in the file picker, and you're done. Click OK on the window, and you should be ready to go.

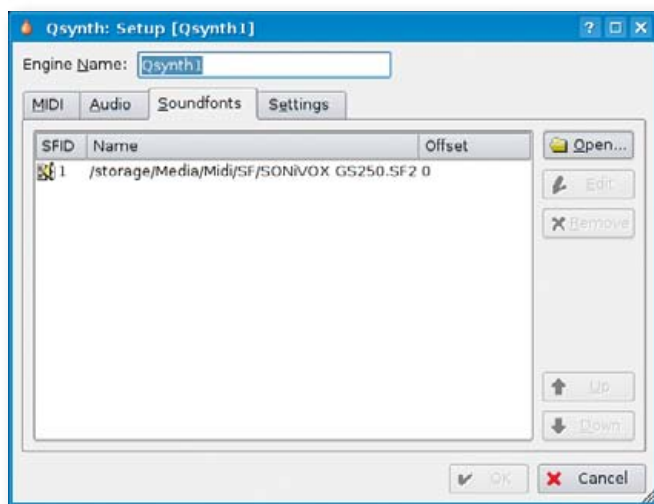


Figure 2. The Soundfont Setup Tab in Qsynth

SONiVOX 250MB GM Soundfont: www.sonivoxmi.com/ProductDetail.asp?Item=GMWavetable250Meg

—Nicholas Petreley

»» Run Multiple X Sessions Simultaneously

On most PCs, you can start more than one X session and switch between them with Ctrl-Alt-F7 and Ctrl-Alt-F8, for example. Why would you do this? Well, some X applications don't really need a full-blown window manager gobbling up your precious RAM and CPU. For example, VMware Workstation and Stellarium are two applications that I use (rarely simultaneously, by the way) that don't need anything but a display. I don't need cut and paste with Stellarium, and VMware is basically a display manager in itself. In addition, just playing with X in this way makes you understand the interrelationships between X, your window manager and your applications.

Start your X engine (aka implicit xinit). You probably just use startx, and it reads your .xinitrc file and, doing what it's told, thereby launches X on the first available console, complete with window manager. This is probably display :0.

Start your X2 engine (aka explicit xinit). From a terminal, you can launch another X server on your machine:

```
xinit /opt/vmware/workstation/bin/vmware
➤-display :1 -- :1 &
```

The first argument taken by xinit is the path for the client that will be launched. It must be an absolute path starting at /. Everything after the -- is passed to the X server. Read the xinit(1) man page a bit for more fine examples.

—Bill Longman

»» Don't Do Things the Hard Way; Run Remote Applications Using the Power of X

It's often extremely frustrating or time consuming to run an xterm on a remote host just to fork your programs from that remote machine. Why not simply run your window manager there, even though you're not on its console? The window manager is just another X application after all, isn't it?

Fire off your local X server:

```
xinit /usr/bin/xterm -- :1 &
```

This yields a vanilla X session with merely an xterm running—no window manager. Now, you need to add permissions to this window session for the remote host. You can tunnel the connection through SSH if your network is insecure, but there's a distinct performance hit. If your network is secure, you can simply do xhost +remotehost and spray directly to your X server.

For tunneled SSH:

```
ssh -fY remotehost /usr/bin/wmaker
```

For spray directly:

```
xhost +remotehost
ssh -f remotehost /usr/bin/wmaker
➤-display localmachine:1
```

The first option, if your remote SSH server supports it, uses a locally defined DISPLAY that then gets tunneled to your local side over SSH. The second option allows remotehost to send X data directly to your local display, then runs Window Maker there but displays it locally. Now, all your desktop actions are done on the remote machine, not locally.

—Bill Longman

Credits

- Nicholas Petreley is Editor in Chief of *Linux Journal*.
- Bill Longman is NIS Manager at Sharp Laboratories of America. ■

Linux Journal pays \$100 for reader-contributed tech tips we publish. Send your tips and contact information to techtips@linuxjournal.com.



REUVEN M. LERNER

Dojo Events and Ajax

The quality of your Dojo depends upon your connections.

Last month, we began looking at Dojo, one of the most popular open-source JavaScript toolkits to emerge in the last year or two. Although using a toolkit is not required if you want to include Ajax or sophisticated client-side functionality in your Web application, it certainly makes things a great deal easier. In particular, such toolkits typically know how to handle the many subtle differences in JavaScript implementations on different browsers. JavaScript is far more standardized than used to be the case, but a number of traps still exist when trying to work with multiple platforms, and using a toolkit can relieve you of having to handle them yourself.

In last month's article, we looked at Dojo's packaging system, some of its enhancements to the JavaScript language and its rich-text editor. This month, we look at some of Dojo's other capabilities that might interest modern Web developers, including support for events and Ajax.

Event Handlers

One of the cornerstones of JavaScript programming is the use of event handlers—functions that are invoked when a particular event occurs. For example, we can define a function that opens an alert box:

```
<script type="text/javascript">
  function openAlert() {
    alert("Hello! This is an alert!");
  }
</script>
```

JavaScript is far more standardized than used to be the case, but a number of traps still exist when trying to work with multiple platforms, and using a toolkit can relieve you of having to handle them yourself.

We can then tell the user's browser to invoke our openAlert function whenever someone clicks on a paragraph of text:

```
<p onclick="openAlert();">This is a paragraph.</p>
```

There are several interesting things to notice in this short example. First, we have set the onclick event handler in this case. About a half-dozen other event handlers exist from which we could choose. In many cases, we might set more than one event handler. This was particularly preva-

Listing 1. test-dojo.html

```
<html>
  <head>
    <title>Testing JavaScript with Dojo</title>

    <script type="text/javascript"
      src="/javascript/dojo.js"></script>
    <script type="text/javascript">
      dojo.require("dojo.event.*");

      function openAlert(evt) {
        alert("Hello! This is an alert from Dojo!");
      }
    </script>

  </head>

  <body>
    <p id="para">This is a paragraph.</p>
    <script type="text/javascript">
      var para = dojo.byId("para");
      dojo.event.connect(para, "onclick", openAlert);
    </script>
  </body>
</html>
```

lent in the pre-CSS days, when JavaScript event handlers would be used to change the look of an icon when the mouse was hovering over it.

Second, event handlers sometimes can be used in contexts you might not expect. For example, the above <p> tag has an onclick handler. You normally wouldn't think of clicking on a paragraph of text, but we can do that. This is the basis for some of the modern drag-and-drop events.

Third, although JavaScript does make it pretty easy to attach handlers to particular events, some messiness still is involved. We cannot define multiple event handlers easily or disconnect handlers that have been defined.

Dojo Events

By this point, you might be wondering what JavaScript event handlers have to do with using Dojo for Ajax and modern Web applications. The answer is that much of Dojo's functionality, across all of its many packages, depends on the event system. If you want to use Dojo's Ajax package, for example, you need to connect it using Dojo events. This might seem restrictive at first glance;

however, Dojo events are surprisingly easy to understand.

As a simple example, let's see how we might implement our onclick handler from before using Dojo events. First, we need to modify our event-handling function so that it takes one argument, the event itself:

```
<script type="text/javascript">
  function openAlert(evt) {
    alert("Hello! This is an alert from Dojo!");
  }
</script>
```

Next, we must connect the paragraph to the event. Rather than doing this directly, by setting the onconnect handler, we give our paragraph an id tag:

```
<p id="para">This is a paragraph.</p>
```

Now, we can use Dojo's `dojo.byId` function—similar in some ways to Prototype's `$$` function—to get the node itself:

```
var para = dojo.byId("para");
```

Finally, we connect our paragraph to the handler

function we created:

```
dojo.event.connect(para, "onclick", openAlert);
```

If we put it all together, we get the program shown in Listing 1, which I have called `test-doj.html`.

One thing you might notice is the three `<script>` tags in the file. The first, in the head of the document, downloads `dojo.js`, the main Dojo source file, from the server. The second, also in the head of the document, imports the Dojo package for events and defines our event-handling function, `openAlert`. The third and final piece of JavaScript, which attaches the node to the event, is in the body of the document, right after our `p` tag is defined. This is because we can set an event handler only for an object that already exists, which means after the `p` tag itself.

If you load the page into a browser window, you will see that it works just like the previous version. Given that this version is more complex, it might not seem obvious how it is better.

Advanced Dojo Events

Here, then, is one example. Suppose you want to invoke one particular object method, rather than a simple function call,



Expert Included.

Our product development team constantly juggles multiple projects with tight deadlines. As an integral part of this team, Patrick is responsible for hardware compatibility, power and performance testing, and operating system validation, just to list a few things.

He is a fan of the Rackform nServ A443, loaded with four Next-Generation AMD Opteron™ processors, because he appreciates the advantages of AMD's Direct Connect Architecture in spreading workload across multiple processors. Patrick has a lot to do, which is why he likes a server designed to do a lot.

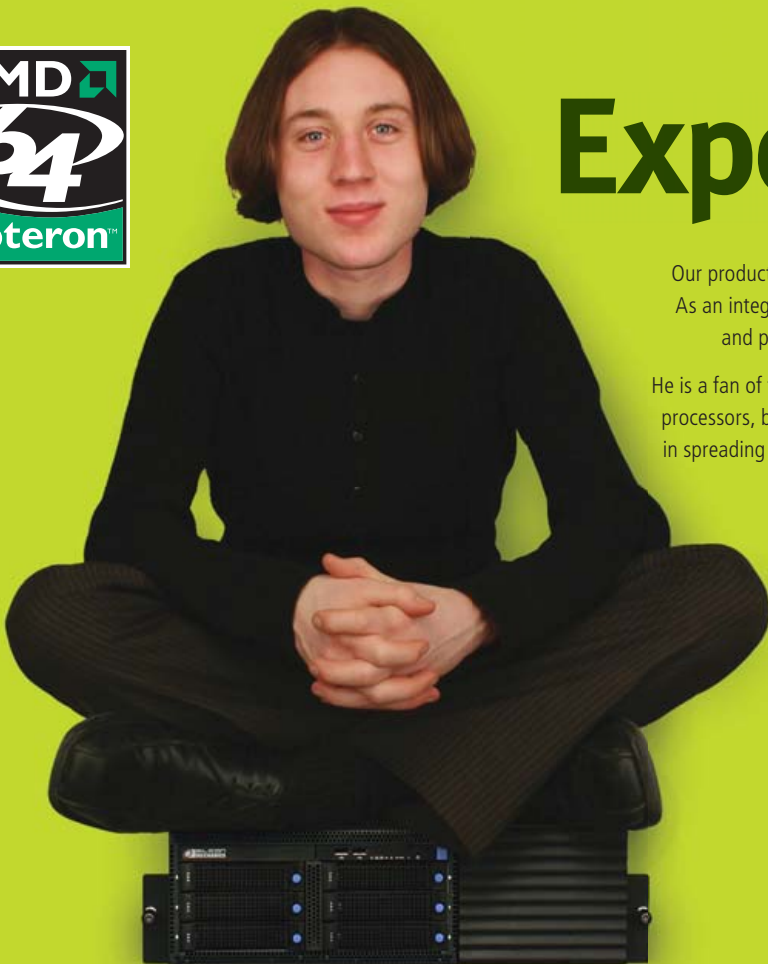
When you partner with Silicon Mechanics, you get more than an enterprise-quality server — you get an expert like Patrick.



visit us at www.siliconmechanics.com
or call us toll free at 866-352-1173

Silicon Mechanics and the Silicon Mechanics logo are registered trademarks of Silicon Mechanics, Inc. AMD, the AMD Arrow logo, AMD Opteron, and combinations thereof, are trademarks of Advanced Micro Devices, Inc.

Professional Product Developer. Do not attempt in your data center.



in an event handler. JavaScript makes it difficult to do this directly from an event handler. However, `dojo.event.connect` handles this quite simply, in its four-parameter version. As before, the first two parameters are the node and event that will trigger the handler. The third and fourth arguments are the object and function that will be invoked. For example:

```
dojo.event.connect(eventObject, "onClick",
    handlerObject, "handlerMethod");
```

Dojo also makes it possible to connect more than one handler to an event. In non-Dojo JavaScript, you could accomplish this only by making your event handler a function that then invokes other functions. Using Dojo events, you can connect any number of methods:

```
dojo.event.connect(para, "onclick", testFormContents);
dojo.event.connect(para, "onclick", submitFormContents);
```

Events are fired in the order that they are connected. So, in the above example, `testFormContents` would be invoked before `submitFormContents`.

Note that Dojo allows you to add the same event handler twice, if you want. So, be careful to invoke `dojo.event.connect` only once for each event-handler combination to avoid potentially odd and hard-to-debug problems.

There is even a topic mechanism for Dojo events, which lets you create multiple channels for event notifications

Let's say you want to provide an expert mode to your users, so they don't have to see all of the annoying alert boxes we're generating. We could create a button that, when pressed, removes the event handler from the object—ooh, but now that's getting kind of tricky, especially if we have multiple events to deal with.

The solution is to use `dojo.event.disconnect`, which does what you might expect:

```
dojo.event.disconnect(para, "onclick", testFormContents);
```

`dojo.event.disconnect` requires that the parameters be completely identical to those used in `dojo.event.connect`. Once it is invoked, however, the event is disconnected.

An advanced piece of the event system is known as *advice*, a term that always has confused me, but which is common in the worlds of Lisp and aspect-oriented programming. The basic idea behind advice is that you can tell the system to invoke a function before or after another function. (If you have used Ruby on Rails, this is analogous to a filter.) This is admittedly an advanced feature, but it might help when debugging an application—rather than inserting logging statements into a problematic function

manually, you simply can add advice to the function, invoking the logger before or after the function is invoked.

There is even a topic mechanism for Dojo events, which lets you create multiple channels for event notifications. (This is similar in some ways to the syslog facility in Linux and UNIX.) Thus, a particular object might register its interest when particular events happen on another object.

Listing 2. `dojo-ajax.html`

```
<html>
<head>
<title>Testing Ajax with Dojo</title>

<script type="text/javascript"
    src="/javascript/dojo.js"></script>
<script type="text/javascript">
    dojo.require("dojo.io.*");
    dojo.require("dojo.event.*");

    function ajaxAlert(evt) {

        var ajaxArgs = {
            url: "hello.php",

            error: function(type, data, evt){
                alert("An error occurred");
            },

            load: function(type, data, evt){
                alert(data);
            },

            mimetype: "text/plain",

            formNode: document.getElementById("myForm")
        };
        dojo.io.bind(ajaxArgs);
    }
</script>
</head>

<body>
<form id="theForm">
<input type="button" id="theButton"
    value="Press here" />
</form>

<script type="text/javascript">
var theButton = dojo.byId("theButton");
dojo.event.connect(theButton, "onclick", ajaxAlert);
</script>
</body>
</html>
```

Finally, Dojo events are used to give functionality to widgets—Dojo’s name for GUI elements made up of HTML and CSS.

Ajax in Dojo

Now that we understand how to create and use Dojo events, we can look at how to perform Ajax queries using Dojo. As you may recall, Ajax (which stands for Asynchronous JavaScript and XML) is a paradigm for Web development that uses the browser’s ability to make HTTP requests behind the scenes. Combining such background HTTP requests with JavaScript, the DOM and CSS makes it possible to create more intuitive and aesthetic Web applications. We could create Ajax applications without Dojo or another toolkit, but it’s much easier and more expressive to use a toolkit, if only because it means we can avoid browser differences and incompatibilities.

Listing 2 shows `dojo-ajax.html`, a page that contains only a single button marked “Press here”. When the button is pressed, the user sees an alert box, much as in Listing 1. But, in this version of the program, the contents of the alert box have come from a server-side program, defined in this case to be the very short `hello.php` (Listing 3).

Listing 3. `hello.php`

```
<? echo "Hello from the server!"; ?>
```

The button itself is defined as we might do with any button to which we expect to attach an event, with an `id` attribute. It sits inside of a very small HTML form, named “theForm”:

```
<form id="theForm">
<input type="button" id="theButton" value="Press here" />
</form>
```

Using Dojo events, we connect the button to a function (`ajaxAlert`):

```
<script type="text/javascript">
  var theButton = dojo.byId("theButton");
  dojo.event.connect(theButton, "onClick", ajaxAlert);
</script>
```

The only remaining question is what the `ajaxAlert` function does. First, it creates a JavaScript object literal and assigns it to the local variable `ajaxArgs`. This object literal assigns several names that will help our Ajax call work: `url` is the URL of the server-side program that will respond to our Ajax call, `error` indicates which function should be invoked if an error occurs, `load` indicates what function should be invoked if the call to `url` is successful and `mimetypes` defines the MIME type we expect to

receive from the server.

One of the annoying aspects of some other JavaScript toolkits is that they require you to create your own list of name-value pairs to be submitted in the Ajax request. This is not the case with Dojo. By setting the `formNode` name in our object literal to a form node, we can rest assured that all the form elements will be passed to the server. In this particular case, that is not necessarily useful or interesting, but it certainly saves some programmer time and increases program readability.

Finally, our `ajaxArgs` object is bound, and we’re off and running. Clicking on the button means the associated Dojo event is invoked, which is `ajaxAlert`. That function, thanks to `dojo.io.bind`, then sends its arguments to the defined URL and invokes the `load` function upon successful completion. This is surprisingly straightforward and opens up many possible avenues for using Ajax in applications.

Conclusion

Dojo, which we explored over the last two installments of this column, and Prototype, which we looked at in the January 2007 issue, are both strong libraries for Web developers looking to improve the quality of their JavaScript. Each has a different style associated with it. I tend to be more of a Prototype kind of guy, but many aspects of Dojo are also quite appealing to me. In particular, Dojo’s extensive set of widgets, and the way those widgets can be connected to one another via the event system, provides a rich set of functionality that all JavaScript developers can enjoy. Even if you don’t plan to use Dojo, you should consider installing and trying it, just to understand how it works. ■

Reuven M. Lerner, a longtime Web/database consultant, is a PhD candidate in Learning Sciences at Northwestern University in Evanston, Illinois. He currently lives with his wife and three children in Skokie, Illinois. You can read his Weblog at altneuland.lerner.co.il.

Resources

The main source for information about Dojo, as well as Dojo software releases, is dojotoolkit.org. Documentation for the toolkit is still a bit sparse, but it has improved significantly in the last few months, and continued improvements seem likely, given Dojo’s growing popularity. The main URL for Dojo documentation is at dojotoolkit.org/docs, and Dojo.book (the Wiki-based Dojo documentation) is at manual.dojotoolkit.org/index.html.

Some good articles about JavaScript toolkits, including Dojo, are at www.sitepoint.com/article/javascript-library.

Finally, a noteworthy introduction to Dojo events is at www.dojotoolkit.org/docs/dojo_event_system.html.



MARCEL GAGNÉ

Security for Your Data—It's Totally Mondo!

Security means different things to different people. On your Linux system, security isn't only about keeping people out, it's also about knowing you can restore the e-mail folder you deleted accidentally.

Why can't I log in to our main server, François? You were trying to improve the security? Without discussing it with me? Yes, yes, of course, I appreciate the spirit of your intentions, but now there's no way to access the system remotely at all. In fact, I can't even log in on the main console. You changed the passwords? And encrypted the filesystems? *Mon Dieu*, François, that is certainly a little over the top. Well, just tell me the new password, and I'll put things back to the way they were. What do you mean, you can't? You've forgotten the passphrase you used when you encrypted the filesystems?

Luckily for you, *mon ami*, our guests are already here, and we have backups. Head to the wine cellar and bring back the 2001 Mas la Plana Cabernet from Spain. And, while you are down there, don't lock any doors or change any combinations.

Welcome, *mes amis*, to *Chez Marcel*, where exceptional vintages are paired with exceptional Linux and open-source software. Please, take your seats and make yourselves comfortable. My faithful waiter is in the cellar fetching the wine. Before you arrived, François demonstrated admirably why it is important to have a reliable backup system. Backups can be simple collections of files burned to a CD,

a tarred bundle stored on a remote system or a copy of your data on a separate drive. There are, in fact, thousands of ways to create a backup, and for many of us, it usually involves backing up only those files that are near and dear to our hearts. In a multiuser environment or on a large, busy system, picking up files here and there may not be enough. You need everything.

Ah, François, you have returned. Please, pour for our guests. Enjoy, *mes amis*, this is a wonderful wine with some great, dark fruit complexity and just a hint of chocolate.

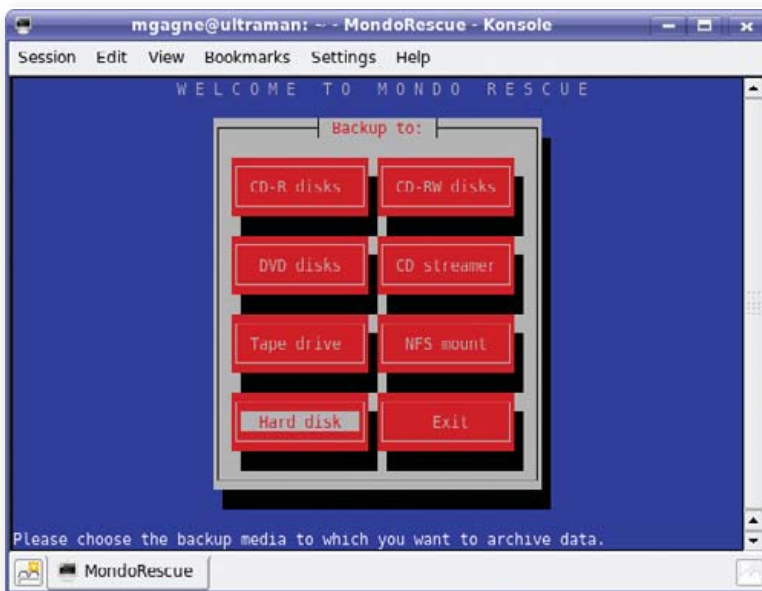
As excellent as the wine might be (and it is), the real star of tonight's menu is a powerful backup and system recovery program called Mondo Rescue. The spirit of Mondo Rescue resides in a scenario no one wants to envision, a catastrophic system failure. I'm not talking about losing your e-mail folder (although I would consider this a catastrophe as well). Mondo Rescue is concerned with "the hard disk is gone, the machine has exploded, and we need to start from scratch" kind of catastrophe. Or, as in François' case, security enhancements gone terribly wrong. Mondo Rescue works with a variety of backup media, and it can create bootable backups that let you restore a mirror image of your system prior to the disaster.

To get started, visit the Mondo Rescue Web site to pick up your copy of the software (see Resources). You'll need a few things to get started, because there isn't a single, all-inclusive Mondo Rescue package. Don't worry; it's a short list, and Mondo Rescue provides packages for an impressive number of distributions and release levels. The packages you need are *afio*, *buffer*, *mindy*, *mindy-busybox*, *syslinux* and the main package, the aptly named *mondo*. As I felt it necessary to use the word "aptly", this is where Debian and Ubuntu users can claim bragging rights, because they can install everything they need by typing `apt-get install mondo`.

Mondo Rescue has, of course, two sides: preparation for disaster and recovery from that disaster. The backup program is called *mondoarchive*, and the restore program is called *mondorestore*. Let's start with the backup program.

The *mondoarchive* program runs in interactive mode by default, with a stylish (by *ncurses* standards) and easy-to-use interface. You navigate the interface by using your keyboard and pressing the Tab key to go from one menu option to another. Start *mondoarchive* from a shell

Figure 1. Ready to back up? Select your medium of choice.



prompt. You also need to be running as root, so something like `sudo mondoarchive` or `su -c 'mondoarchive'` should work well.

The Welcome screen (Figure 1) also is the selection screen for your backup medium. You can choose from CD-R or DVD-R disks, tapes, an NFS-mounted directory, a location somewhere else on disk and more. Given the nature of a catastrophic disaster, somewhere on your local disk may not appear to be the best choice, but you also can use Mondo Rescue to generate bootable-CD or DVD ISO images from which you can boot and restore your system. Because many home users have access to a CD or DVD writer to which they can burn these images, but not necessarily a tape drive, let's use that as our example.

By the way, this isn't the same as backing up directly to a CD- or DVD-recordable drive. If you choose that option, you are asked to insert blank disks at various points in the process.

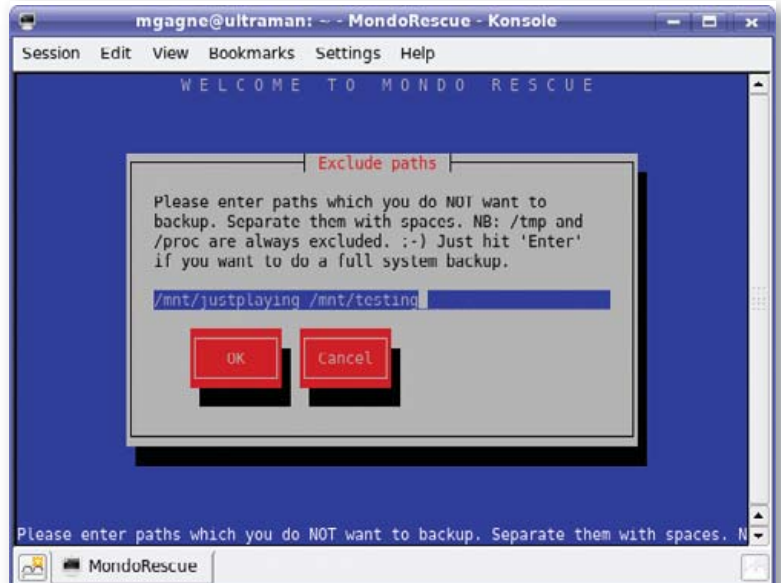
Tab to the Hard disk option, and press Enter. You'll be asked for the pathname to the disk location you want to use for your backup (Mondo Rescue will provide a suggestion). If you chose a tape-drive backup, Mondo Rescue would try to guess the location of your tape drive—normally successfully.

The next screen (Figure 2) is worth thinking about, because it seriously affects the performance of your backup. This is the compression screen. To minimize the space in which backups are stored, the `mondoarchive` program can compress files on the fly. You can elect to skip compression or select minimum, average or maximum compression. The higher the compression, the more impact on speed and performance.

Those of you following along with my example will be writing bootable-ISO image backups to disk, but what kind of images? CD-Rs can store 650MB–700MB of data (depending on the type you bought), and DVDs can store roughly 4GB. Enter the information in megabytes, press Tab to select OK, and then move on to the next screen. The ISO images are called `mondorescue-1.iso`, `mondorescue-2.iso`



Figure 2. Compression can affect the performance of your backups dramatically.



and so on. You now have the opportunity to override that naming convention by selecting a different name. If you're happy with the default, press Enter to continue.

Next, is the Backup Paths screen. By default, everything is backed up from the root (`/`), on down. Most people will be happy with this and can safely move on to the next screen. Incidentally, should you happen to have a system with NTFS partitions (such as on dual-boot systems with Windows), Mondo Rescue offers to back up those as well and informs you of their presence. You can accept these or remove them from the list of backed-up partitions.

Having mentioned that it makes sense to back up the whole system, I recognize you probably really don't want everything. On my system, I often have entire filesystems where I download ISOs of Linux distributions so I can experiment with them on virtual machines. I don't want to back these up. I also have folders filled with what can be described only as ephemeral junk—things that seemed like a good idea at the time, but that I haven't gotten around to cleaning up, and certainly don't want to back up. Simply list all the folders you want to exclude from backup, separated by spaces.

At this point, you are almost ready to roll. The `mondoarchive` program asks whether you want to verify your backup, and then it follows up with a very strange question: "Are you confident that your kernel is a sane, sensible, standard Linux kernel? Say 'no' if you are using Gentoo < 1.4 or Debian < 3.0, please." Mondo Rescue wants to make sure the kernel it uses to boot the CD (or DVD) has the smarts to boot properly. If you have any doubts, or you like to spin your own kernels, say no, and Mondo Rescue will use its own. Once you have made a choice, the `mondoarchive` program alerts you that it is ready to start. This is your last chance to change your mind.

The backup begins, also in ncurses graphical mode, starting with the creation of a catalog of filenames to back

Figure 3. You can trim your backups by excluding certain folders or filesystems.

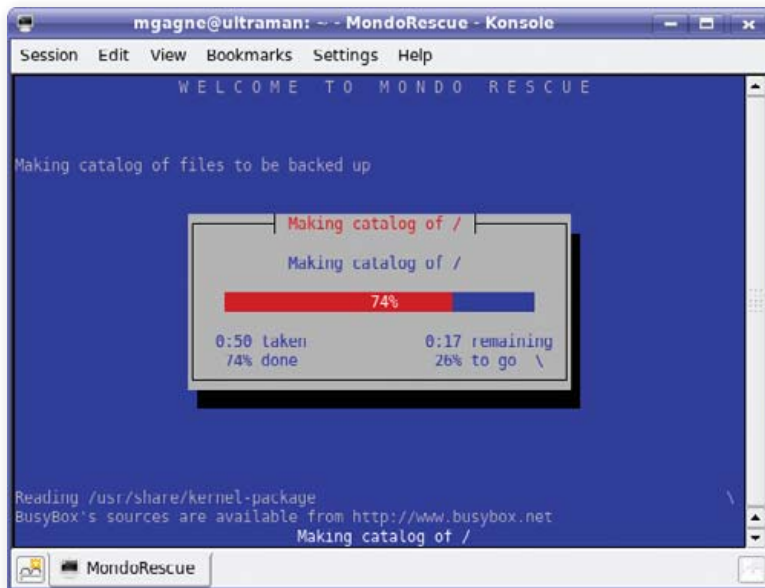


Figure 4. Mondo Rescue creates a catalog of files when starting the backup.

up (Figure 4).

What follows next is interesting only the first few times—mostly because you probably have better things to do with your time. The screen shows a report of the backup broken up into file sets, the creation of boot diskettes and so on. At this point, Mondo Rescue is ready to back up your data and displays a nice progress bar, telling you which ISO is being written, how much of it is done and how long you can expect the whole process to take (Figure 5).

Speaking of better things to do with our time, this is probably a good time for a wine refill. François, please make sure our guests' glasses are topped up.

This is all well and good, but sitting in front of a terminal session running a backup isn't what most people want to do most of the time. Consequently, all of this can be done from the command line, which is exactly what you want if you are going to run the program from a cron job. For example, take a look at the following command:

```
mondoarchive -Oid /mnt/bigdrive -l GRUB -F -V -3 -N
```

That command says to create a mondoarchive backup (-O), to create ISO images (-i), to use a location on disk (-d), that the bootloader is GRUB (-l), to skip the creation of boot diskettes (-F), to verify the backup (-V), to use moderate compression (-3) and to ignore NFS-mounted partitions (-N). I'm going to concentrate on the interactive mode of the backup here, but I invite you to examine the various options by typing `man mondoarchive` at a command prompt.

Eventually, you will have a complete backup and, in this case, one or more ISO images that you can burn to a CD or DVD. The first disk in the set is the one from which you'll want to boot. In a few seconds, you'll see a menu like the one shown in Figure 6 (currently running in a QEMU virtual machine).

You have several options when it comes to restoring



Figure 5. The backup is underway, with an on-screen progress report.

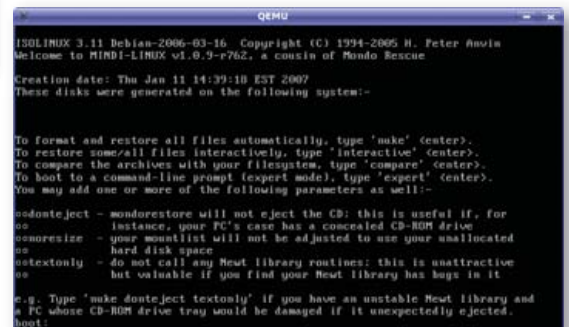


Figure 6. The Mondo Rescue Boot Menu



Figure 7. The Top-Level Menu for the mondorestore Program

your system (nuke, interactive and expert), including *not* restoring your system (compare). If you choose the nuke option, your system is restored as it was, and any filesystems currently on your computer are destroyed and re-created from the backup. Use this option with extreme care. You also might want to restore one or more files and folders. For this, use interactive mode. Finally, expert mode drops you to a command prompt. You also can simply wait a few seconds, and the restore disk boots normally and then takes you to a graphical (ncurses) interface for the mondorestore program (Figure 7).



One

PGI Unified Binary™

Now, PGI® compilers can generate a single PGI Unified Binary executable fully optimized for both Intel EM64T and AMD64 processors, delivering all the benefits of a single x64 platform while enabling you to leverage the latest innovations from both Intel and AMD. PGI Fortran, C, and C++ compilers deliver world-class performance and a uniform development environment across Linux and Windows as part of an integrated suite of multi-core capable software development tools. Visit www.pgroup.com to see why the leading independent software vendors in structural analysis, computational chemistry, computational fluid dynamics and automotive crash testing choose PGI compilers and tools to build and optimize their 64-bit applications.



The Portland Group™
www.pgroup.com ++ 01 (503) 682-2806

Your four choices, although worded differently, are the same as those you saw earlier at boot time. If you choose Interactively, you'll be prompted for the source of your backups. Before we go any further, it's worth noting that the idea behind Mondo Rescue is to provide a means of disaster recovery when everything is gone, which is why backups are created to be bootable (tapes, CDs and so forth). This is fantastic if major disaster strikes, but what if it's a minor disaster, such as accidentally deleting your boss' e-mail folder? You certainly don't want to take down a running production system, even if the only important information in his e-mail folder are stats from a football pool. Luckily, you can restore a file or folder to a live system, interactively. This is how you do it.

From the command line, type `mondorestore`. An ncurses-based display appears asking for your boot disk, CD or floppy. Simply press Enter, and you'll find yourself at the file catalog.

It may take a few seconds for the program to extract the file catalog, but soon you'll be presented with a list of files and folders starting from the root directory. Using the arrow keys, you can navigate up and down through the list. Along the bottom of this screen are text buttons labeled Less, More, Toggle, RegEx, Cancel and OK (Figure 8). To expand a folder or directory, cursor to the right, go to the More button, and press Enter. To select a file or folder for restoring, cursor right again to the Toggle button, and press Enter. An asterisk appears to the left of the filename you've selected. Press Enter again to deselect it. To continue searching through the file list, cursor left past the Less button, and you can scroll up and down through the list again.

Before you ask, the reason I didn't mention the RegEx button is that this is still a feature under development, and it really doesn't do anything at this time.

Once you have selected everything you want to restore, cursor over to the OK button, and press Enter. An alert pops up asking whether you are happy with your selection. Press Yes to continue with the restore. On the next screen, select a restore path. If you want to restore in place (and overwrite any current files), accept the default, which is the root direc-

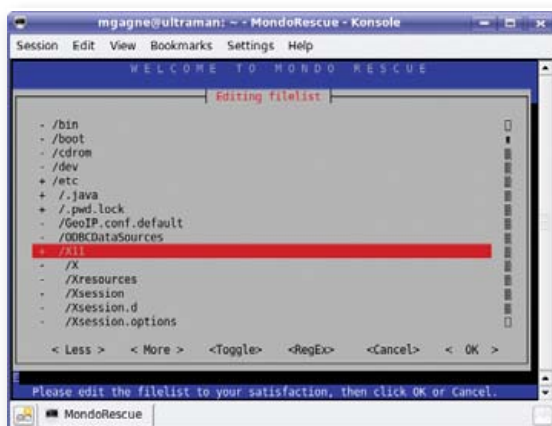


Figure 8. Choosing the Files or Folders to Restore

tory. Often, you'll want to restore a file into an alternate location and move it back when you are satisfied with its content. If that is the case, enter an alternate path, and press Enter. The next screen (Figure 9), boasts "Restoring from archives" and provides a nice report of the restore process.

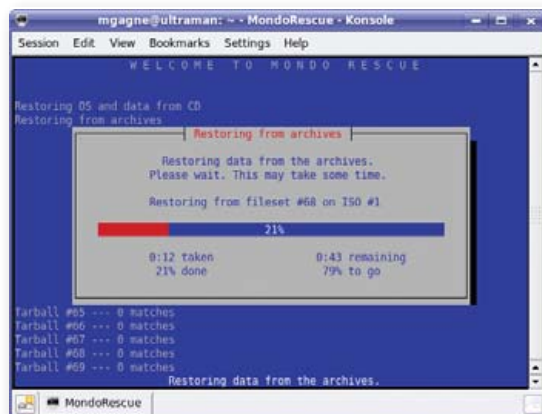


Figure 9. Hurrah! The lost files are being restored.

The dialog displays the tarball in which it is currently searching, on which disc, a percentage of completion and an estimated time remaining before all your files are restored. That's it. Your all-important files (and, they are *all* important when lost) have been restored.

Once again, *mes amis*, the clock indicates that it is indeed closing time. I trust you are feeling satisfied and relaxed from the wine. While François refills your glasses a final time, I should point out that development on Mondo Rescue is ongoing, and there is a helpful and enthusiastic user base, ready to help with any issues you might encounter. Take a moment to visit the support page and join the mailing list on the Mondo Rescue site, and you'll not only be more relaxed, you also will sleep soundly knowing your data can be restored. Please raise your glasses, *mes amis*, and let us all drink to one another's health. *A votre santé! Bon appétit!* ■

Marcel Gagné is an award-winning writer living in Waterloo, Ontario. He is the author of the all-new *Moving to Free Software*, his sixth book from Addison-Wesley. He also makes regular television appearances as Call for Help's Linux guy. Marcel is also a pilot, a past Top-40 disc jockey, writes science fiction and fantasy, and folds a mean Origami T-Rex. He can be reached via e-mail at mggagne@salmar.com. You can discover lots of other things (including great Wine links) from his Web site at www.marcelgagne.com.

Resources

Mondo Rescue: www.mondorecue.com

Marcel's Web Site: www.marcelgagne.com

The WFTL-LUG, Marcel's Online Linux User Group: www.marcelgagne.com/wftllugform.html

EmperorLinux

...where Linux & laptops converge



Portable

Since 1999, EmperorLinux has provided pre-installed Linux laptops to universities, corporations, government labs, and individual Linux enthusiasts. Our laptops range from full-featured ultra-portables to desktop replacements. All systems come with one year of Linux technical support by phone and e-mail, and full manufacturers' warranties apply.

Toucan T60/T60ws

ThinkPad T60/T60ws by Lenovo

- Up to 15.4" WSXGA+ w/ X@1680x1050
- ATI Mobility FireGL V5200
- 1833–2333 MHz Core 2 Duo
- 512 MB–4 GB RAM
- 60–120 GB hard drive
- CDRW/DVD or DVD±RW
- 5.2–6 pounds
- 10/100/1000 Mbps ethernet
- 802.11a/b/g (54Mbps) WiFi
- Starts at \$1950



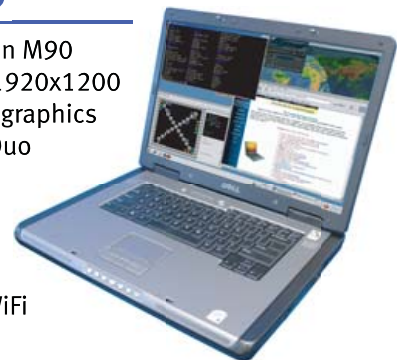
Powerful

EmperorLinux specializes in the installation of Linux on a wide range of the finest laptops made by IBM, Lenovo, Dell, Sony, and Panasonic. We customize your choice of Linux distribution to your laptop and provide support for: ethernet, wireless, X-server, ACPI power management, USB, EVDO, PCMCIA, FireWire, CD/DVD/CDRW, sound, and more.

Rhino D820/M90

Dell Latitude D820/Precision M90

- Up to 17" WUXGA w/ X@1920x1200
- NVidia Quadro FX 3500M graphics
- 1667–2333 MHz Core 2 Duo
- 512 MB–4 GB RAM
- 40–160 GB hard drive
- CDRW/DVD or DVD±RW
- 6.3–8.6 pounds
- 802.11a/b/g (54Mbps) WiFi
- ExpressCard/EVDO
- Starts at \$1455



Unique

EmperorLinux offers Linux laptops with unique features. Ruggedized Panasonic laptops are designed for harsh environments: drops, vibrations, sand, rain, and other extremes. ThinkPad tablet PCs are like other laptops, with an LCD digitizer for pen-based input both as a mouse and with pressure sensitivity for writing and drawing on-screen.

Raven X60 Tablet

ThinkPad X60 Tablet by Lenovo

- 12.1" SXGA+ w/ X@1400x1050
- 1667–1833 MHz Core Duo
- 1–4 GB RAM
- 80–120 GB hard drive
- 4 pounds
- Pen/stylus input to screen
- Dynamic screen rotation
- Handwriting recognition
- X60s laptops available
- Starts at \$2300



www.EmperorLinux.com

1-888-651-6686



DAVE TAYLOR

Displaying Image Directories in Apache

Step one toward a shell script for Web-based image management.

Most of the time when I write shell scripts, it's to solve what I consider a lightweight problem. Yes, I admit it, if you need to forecast weather, geomap 50,000 data points or create an on-line shopping cart, a shell script is probably not the optimal tool!

Nonetheless, when I encounter problems or opportunities for simplification in my daily work, the first tool out of the box is a shell script. For some of you, it might be Perl or some fancy PHP coding, but because anyone who can type commands on the Linux command line is ready to start scripting, I have to say I still believe shell scripts are a good starting point.

What's surprising is just how much you can accomplish in a short segment, and this month I share a script I cobbled together to address what might be a common problem on your Web server too—a huge “Images” directory.

Apache Directory Listings

To be perfectly candid, the directory listings that are generated by Apache and other Web servers stink. They're basically `ls -l` with no additional information, no previews, nothing. Most of the time it doesn't really matter, because most of your site is probably seamless, and people aren't exposed to the back end.

But, the directory where you might collect all the images, graphics and photos on your site is most likely a different story. Whether it's called “Images”, “Graphics”, “Photos”, “Art” or what have you, odds are that your directory is like my own: 1,400 graphics files.

A text-based listing capability is useful if the files have highly mnemonic names, but wouldn't it be far more useful to have thumbnails of all the images shown along with their names, rather than only file size and last-modified dates?

That's what this script does, and like all scripts that are actually working as CGI scripts, it has to start out by pushing the appropriate header information immediately:

```
#!/bin/sh

echo "Content-type: text/html"
echo ""
```

Now that that's out of the way, the rest of the content can be generated in a loop. In fact, the first skeletal version of the script just duplicates the file listing capability already in your Web server:

```
for name in *
do
    echo "$name <br>"
done
```

Of course, this output isn't all that interesting. At a minimum, we can change it so that the filenames are clickable:

```
echo "<a href=$name>$name</a><br>"
```

But, even that's not particularly interesting. Let's add some conditional code so that images are displayed while everything else just garners a link. Rather than testing the filename though, let's do something more interesting and use the `unsung` command file.

When just run against the contents of a typical image directory, here's the kind of output you can expect:

```
$ file *
aol-safety-menu.png:      PNG image data, 161 x 230,
 8-bit/color RGB, non-interlaced
apple-ipod-enter-code.png: PNG image data,
 268 x 202, 8-bit/color RGB, non-interlaced
archos-av700.png:        PNG image data,
 567 x 294, 8-bit/color RGB, non-interlaced
empty.jpg: empty
hentai-manga-example.gif: GIF image data,
 version 89a, 358 x 313,
index.cgi:                Bourne shell script
text executable
teamgeist.jpg:            JPEG image data,
 JFIF standard 1.02, aspect ratio, 100 x 100
```

Nice command, eh? It includes the type of the image, dimensions, depth and any other characteristics it can ascertain.

Most important, notice that “XX image data” appears consistently with these images, whether they're PNG, JPG or GIF images. By using this, we can avoid all the hassles with JPG vs. JPEG, JPG vs. jpg, Gif vs. GIF and on and on.

Now, the little loop looks like this:

```
for name in *
do
    if [ ! -z "$(file $name | grep 'image data')" ] ; then
        echo "$name <br>"
    fi
done
```

To be perfectly candid, the directory listings that are generated by Apache and other Web servers stink.

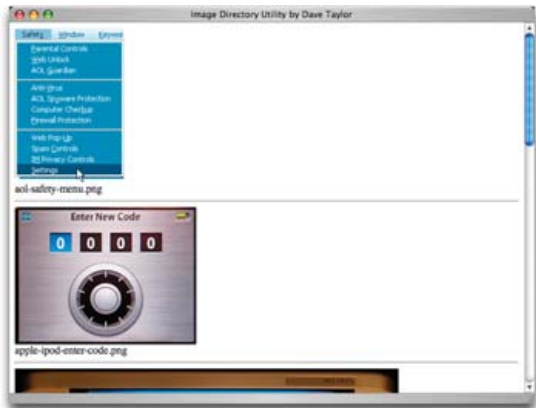


Figure 1. Script in Action

```
fi
done
```

This is enough so that the files that aren't images, even empty.jpg, which is a zero-byte file, are skipped automatically:

```
$ sh index.cgi
Content-type: text/html

aol-safety-menu.png <br>
apple-ipod-enter-code.png <br>
archos-av700.png <br>
hentai-manga-example.gif <br>
teamegeist.jpg <br>
```

Finally, we're getting somewhere, because now we can differentiate between the files that actually are images, and the files that are other sorts of data.

One last refinement before I wrap this up: instead of just showing the links as clickable, let's actually output clickable links for non-images, and make the images themselves clickable. This can be done as follows:

```
for name in *
do
  if [ ! -z "$(file $name | grep 'image data')" ]; then
    echo "<a href=$name><img "
    echo "src=$name></a><br>$name<hr>"
  else
    echo "<a href=$name>$name</a><hr>"
  fi
done
```

If the images aren't too large, this starts to look pretty nice, as you can see in Figure 1. If they are big images, however, it doesn't work quite as well. So, next month I'll show you some refinements to this script, including how we can have more than one image appear on a line. ■

Dave Taylor is a 26-year veteran of UNIX, creator of The Elm Mail System, and most recently author of both the best-selling *Wicked Cool Shell Scripts* and *Teach Yourself Unix in 24 Hours*, among his 16 technical books. His main Web site is at www.intuitive.com.

**Hardware Systems For The
Open Source Community—Since 1989**
(Linux, FreeBSD, NetBSD, OpenBSD, Solaris, MS, etc.)

**The AMD Opteron™ processors deliver high-performance,
scalable server solutions for the most advanced applications.
Run both 32- and 64-bit applications simultaneously**

**AMD Opteron Value Server-
\$795**

1 U 14.3" Deep
AMD Opteron 140 1M Cache
1 GB DDR ECC REG PC-3200
1 of 2 40GB SATA Drive
2 X 10/100/1000 NIC
Options: CD, FD, or Second Drive, Raid
ADD Your Logo



**iSCSI Dual AMD Opteron
1U to 8U, Call for Pricing**

1TB to 30TB of iSCSI Storage
Dual AMD Opteron 246
1 GB DDR ECC REG PC-3200
Dual GigE LAN
Redundant PS, Hot-Swap Drives
RAID Options, RAID 5, 10, 50
More Customization is available



**1U SCSI Quad AMD Opteron-
(Rev.F) Starting @ \$5293.20**

1 of 4 AMD Opteron 8212 CPU
12 GB DDR2 ECC REG PC-3200
1 of 3 73GB SCSI Drive
2 GigaE, CD, FD,
Optional Remote Management Card (IPMI)
Call for more choices of AMD socket F servers.



**30TB AMD Opteron Storage
Solution- Starting @ \$26,395**

30TB SATA Storage in 8U
Includes all Raid Cards, Raid 5, 10
Dual AMD Opteron 246
2 GB DDR, ECC REG PC-3200
Dual GigE, FD, CD



Your Custom Appliance Solution

Let us know your needs, we will get you a solution



Custom Server, Storage, Cluster, etc. Solutions

Please Contact us for all type of Storage solutions,
NAS, DAS, iSCSI, Fiber RAID, SATA, SAS.

*Free shipping on selected servers and all notebooks



2354 Calle Del Mundo, Santa Clara, CA 95054

www.asacomputers.com

Email: sales@asacomputers.com

P: 1-800-REAL-PCS | FAX: 408-654-2910

Prices and availability subject to change without notice.
Not responsible for typographical errors. All brand names and logos
are trademark of their respective companies.



MICK BAUER

Linux Firewalls for Everyone

Need a personal firewall, an enterprise Internet gateway or something in between? iptables does it all!

The **Linux kernel** includes some of the most powerful and flexible firewall code in any general-purpose operating system. This code is called Netfilter, though most of us refer to it by the name of its user-space command, iptables. Netfilter/iptables allows your Linux kernel to inspect all network traffic that passes through your system, deciding what to do with that traffic based on a very rich set of criteria.

Building Linux firewalls with iptables is a big topic—entire books have been written about it (see Resources). In fact, firewall engineering is a profession unto itself (my profession, in fact). So, alas, nobody can tell you everything you need to know about building firewalls with iptables in one magazine article.

I can, however, provide an overview of the things iptables can do, some sound principles for Linux firewall design, descriptions of some handy tools for building different types of Linux firewalls and pointers to more detailed information on Linux firewalls.

Types of Linux Firewalls

Firewalling, or more precisely, packet filtering, can be used for many things. It can be used locally on individual servers and desktop systems for host-level protection from network-based attacks. It can be used at the network infrastructure level to protect entire networks from other networks, and it can be used to redirect, or even alter, network packets in various ways.

A Linux firewall can be a dedicated hardware appliance based on Linux, a PC with multiple network interfaces, or it even can be an ordinary, single-interfaced workstation or server. Many commercial firewall appliances are Linux/iptables-based. Contrary to what you might think, PC-based Linux firewalls can perform and scale quite well, if deployed on sufficiently powerful hardware.

Those are the form factors Linux firewalls take, and they serve in two different roles. Firewall appliances and multi-interface PC-based firewalls are used as what I call network firewalls. They serve as dedicated network devices, logically equivalent to IP routers that regulate traffic between different networks. (Technically, firewalls *are* routers; they're just fussier about what they route than ordinary routers.) Network firewalls also often perform Network Address Translation (NAT), typically to allow hosts with non-Internet-routable IP addresses to communicate with the Internet.

Then, there are what I call local firewalls—worksta-

tions or servers whose primary function isn't firewalling at all, but the need to protect themselves. In my opinion, *any* computer connected to the Internet, whether server or workstation, should run a local firewall policy. In the case of Linux systems, we have no excuse for not taking advantage of Linux's built-in Netfilter/iptables functionality. Furthermore, this is the easiest type of firewall script to create, as I show later in this article.

Firewall Design Principles

Before we discuss Linux firewall tools, we should cover some general firewall design principles. Most of these principles are (or should be) equally valid whether you're using iptables to protect a single host or entire networks.

First, here are some terms:

- **Packet filtering:** the practice of inspecting individual network packets, comparing against a set of rules and processing them accordingly.
- **Firewall policy:** either a specific set of iptables commands or a higher-level set of design goals that your iptables commands enforce.
- **Firewall rules or packet-filtering rules:** the individual components of a firewall policy—that is, individual iptables command iterations.

The first step in building any set of packet-filtering rules is to decide precisely what you want your firewall to do—that is, to formulate your high-level firewall policy. For example, if I'm creating a local firewall script for a workstation, my logical policy might look like this:

1. Allow outbound DNS queries, Web surfing via HTTP and HTTPS, e-mail retrieval via IMAP, outbound SSH and outbound FTP transactions from the local system to the entire outside world.
2. Allow inbound SSH connections to this system from the other workstation in my basement.
3. Block everything else.

Skipping this crucial step of defining your high-level policy is akin to writing a software application without first defining

requirements. You run the risk of wasting time on rules you don't need and of overlooking rules that you do need.

I further recommend that whatever policy you decide on, you make it *as restrictive as is feasible*. Marcus Ranum very succinctly stated the guiding principle for firewall design many years ago: "that which is not expressly permitted is forbidden". The reason for this is quite simple; just because you can't think of how an allowed but unnecessary network transaction can't be abused, doesn't mean some attacker can't abuse it nonetheless.

Every firewall policy, therefore, must logically end with a rule that blocks everything not specified earlier.

This is true not only for network/enterprise firewall policies, but also of personal/local firewall policies. A common blunder on personal firewalls is to allow all "outbound" transactions, on the assumption that all local processes are "trusted". If your system is infected with a worm, trojan or virus, however, this assumption breaks down.

In the event of such an infection, you probably don't want the malware to be able to use your system to send spam, participate in distributed denial-of-service attacks and so forth. Therefore, it's preferable to restrict not only "inbound" (externally originated) network transactions, but also outbound (internally/locally originated) transactions, even on the local firewall policies of desktop systems and servers.

Another important firewall design principle is, whenever possible, to group similar risks together. In other words, systems and networks with different levels of trust and different levels of exposure to risk should be isolated from each other by network firewalls.

The classic example of this principle in action is the DMZ or de-militarized zone, which is a network containing all of an organization's Internet-accessible systems. Figure 1 shows the relationship between such a DMZ: the "internal" network containing an organization's workgroups and other non-public-facing network resources and the Internet.

With firewalls separating the DMZ network from both the Internet and the internal (trusted) network, you can write rules that specify, in a very granular way, how hosts in these three zones can interact with each other. In formulating such rules, you should assume that, being exposed to a nearly infinite range of possible attackers (via the Internet), the hosts in your DMZ should be treated as semi-trusted at best—that is, you should assume that any host in the DMZ may be compromised at some point. Accordingly, you should allow as few transactions as possible to be initiated from the DMZ to the internal network.

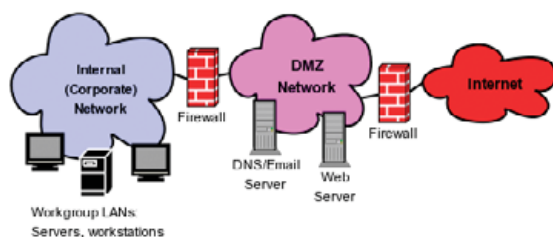


Figure 1. A DMZ Network

You also should take into consideration the threat a compromised DMZ host could pose to the outside world. If an Internet-based attacker compromises your DNS server, for example, even if the attacker's attempts to hack into your internal network are blocked by firewall rules, that attacker can still cause your organization embarrassment or even legal problems if the compromised server is able to connect arbitrarily (that is, attack) to other systems on the Internet. I can't state this often or strongly enough: firewall policies should allow only the bare minimum set of network transactions necessary for your users and systems to do their jobs. Unnecessary dataflows can *and will* be abused, sooner or later.

You probably noticed that in Figure 1, two firewalls are used. This is the classic firewall sandwich DMZ architecture, but many organizations opt instead for a more economical multi-homed-firewall DMZ architecture (Figure 2), in which a single firewall with multiple network interfaces interconnects and restricts traffic between different networks. Although the sandwich topology provides greater protection against, for example, the external firewall itself being compromised in some way (assuming the other firewall isn't subject to the exact same vulnerability), the multi-homed-firewall approach can be equally effective, so long as you write your rules carefully enough.

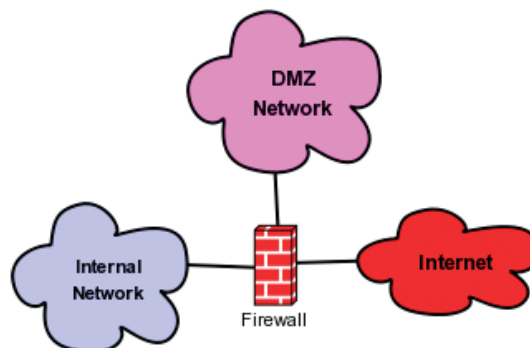


Figure 2. A DMZ and a Multi-Homed Firewall

Also, regardless of whether you use a single multi-homed firewall or pairs of firewalls, it's extremely important that each network zone (inside, outside/Internet and DMZ) be connected to a *dedicated* physical network interface on a firewall. Yes, this does make your firewall a potential single point of failure. However, if it's possible for hosts in one network zone to route packets to other network zones without traversing the firewall, your firewall will have little practical value!

The last general firewall design principle I mention for now applies only to multi-interface firewalls (that is, not to local/personal firewalls): always use anti-spoofing rules.

Consider the Internet-facing firewall in Figure 1. It has two network interfaces: inside (which faces the DMZ) and outside (which faces the Internet). Suppose that the internal network in Figure 1 uses IP addresses in the Class C network space 192.168.55.0/24, and the DMZ

uses 192.168.77.0/24.

This firewall therefore can and should drop any packets arriving on its Internet interface having source IP addresses from either of those two private IP ranges. Such packets safely can be assumed to be forged (spoofed). Attackers sometimes forge the source IP addresses of their packets, attempting to pass them through firewalls or to defeat other source-IP-based authentication mechanisms (TCPwrappers, hosts.equiv and so on).

In fact, *any* Internet-facing network interface on *any* firewall should drop packets with source IP addresses from *any* non-Internet-routable IP range, specifically those specified in RFC 1918: 10.0.0.0/8, 172.16.0.0/12 and 192.168.0.0/16. (If these numbers, which are ranges of IP addresses expressed in CIDR notation, confuse you, don't panic! Some of the iptables tools discussed later in this article assume no particular networking expertise.)

To express this important firewall design principle even more generally: you should configure your firewall to drop any packet bearing an impossible source IP address.

Those are some things all firewalls should do. Now, how do we make them do those things?

Firewall Tools for Linux

All Linux firewalls work the same way. A series of iptables commands are executed in sequence to load firewall rules into kernel memory space. All packets entering all network interfaces are then evaluated by the kernel based on these rules and handled accordingly. These rules are organized in tables (formerly, and still occasionally, called chains). Rules can be inserted, appended,

To express this important firewall design principle even more generally: you should configure your firewall to drop any packet bearing an impossible source IP address.

changed and deleted from any table at any time via the iptables command and take effect immediately.

The most direct way to create a Linux firewall policy is to write an iptables startup script from scratch and then manage it like any other startup script in `init.d`. This is how I manage my own Linux firewalls, and it works fine if you understand networking, you're comfortable with the iptables command, and you don't have many different firewalls to manage or more than a couple of different policies on any given firewall.

To learn how to roll your own iptables scripts, refer to the Resources for this article. As I said previously, I simply can't do that topic justice here. (Note that different Linux distributions handle startup scripts differently.) If you want to harness the full power of iptables, including NAT, custom chains and packet-mangling, this really is the best way to go.

Assuming you can't, or don't, want to write iptables scripts directly, here are some pointers to tools that can help.

Personal (Local) Firewalls

The first category of iptables tools I discuss here probably already exists on your system. Nowadays, nearly all Linux distributions include a firewall wizard in their installation utilities. Nearly always, this wizard is intended for creating a local firewall policy—that is, a personal firewall script, which protects only the local host.

These wizards all work the same way. They ask you which local services you want to allow external hosts to reach, if any. For example, if I'm installing Linux on an SMTP e-mail server, I would allow inbound connections only to TCP port 25 (SMTP), though possibly also to TCP port 22 (Secure Shell, which I may need for remote administration).

Based on your response, the wizard then creates a startup script containing iptables commands that allow incoming requests to the services/ports you specified, block all other inbound (externally originating) transactions and allow all outbound (locally originating) network transactions.

But wait! That third command violates Ranum's principle (deny all that is not explicitly permitted), doesn't it? Yes, it does. That's why I write my own iptables scripts even for local firewall policies. You need to decide for yourself in any given situation whether you can live with the "allow some inbound, allow all outbound" compromise, which is undeniably the simplest approach to local firewalls, or whether you're worried enough about the threat of malware mischief to write a more restrictive script, either manually or using a more sophisticated firewall tool than your Linux distribution's installer.

Note that as with other functions of Linux installers, these firewall wizards usually can be run again later, for example, in SUSE via YaST's Security and Users→Firewall module.

Two Tools for Network Firewalls

We've discussed the hard way (writing your own iptables startup script) and the easy way (letting your Linux installer generate a local firewall script). There are, however, many other tools for generating and managing sophisticated firewall scripts. Two of the most popular are Shorewall and Firewall Builder (see Resources).

Shorewall is, essentially, a script/macro environment that lets you create firewall policies in the form of text files, which are then "compiled" into iptables scripts. Shorewall's strengths are its flexibility, its ability to insulate users from needing to learn iptables syntax and its convenience in automatically generating startup scripts. If you understand networking, however, learning to use Shorewall isn't necessarily that much less time consuming than learning iptables.

For this reason, I've personally not used Shorewall very much. Friends of mine, however, who know less than me about networking but more about system administration, swear by it.

Firewall Builder, which I covered several years ago in the May and June 2003 issues of *Linux Journal*, is something else altogether. It's the firewall equivalent of an Integrated Development Environment—that is, a graphical, object-oriented interface for generating iptables scripts (among other firewall types).

Conceptually, Firewall Builder is very similar to the policy editor in Check Point firewalls. You create "objects" for the networks and hosts you want to use in rules, and then you arrange those objects and pre-defined "service" objects (HTTP, IMAP, FTP and so forth) into graphical rules statements. Firewall Builder not only generates these into iptables scripts, but it also can install them on other systems via SSH.

In my experience, the main strike against Firewall Builder is its somewhat lengthy list of dependencies, chief among them the Qt libraries for GUI development. However, many of the things Firewall Builder needs are now standard Linux packages included on typical distributions, so this is less of a problem than it used to be. See the Firewall Builder home page for detailed installation instructions.

Other graphical iptables utilities include Firestarter and Guarddog (see Resources).

Conclusions

A couple years ago, *Linux Journal* named iptables its Security Tool of the year. It really is a remarkable achievement. If you're serious about network security, you'll want to explore iptables' power in much greater detail than we've done in this article, starting with the iptables(8) man page and progressing through the how-tos available on the Netfilter home page (see Resources).

Whether you use iptables to protect your laptop or your entire enterprise network, I hope you've found this introduction useful. Be safe! ■

Mick Bauer (darth.elmo@wiremonkeys.org) is Network Security Architect for one of the US's largest banks. He is the author of the O'Reilly book *Linux Server Security*, 2nd edition (formerly called *Building Secure Servers With Linux*), an occasional presenter at information security conferences and composer of the "Network Engineering Polka".

Resources

The Netfilter home page, where you can find the most current iptables-related how-tos: www.netfilter.org

Home page for Firewall Builder, an object-oriented GUI for generating and managing rules for several different types of firewalls, including iptables: www.fwbuilder.org

The Shorewall (Shoreline Firewall) home page: www.shorewall.net

Suehring, S., and Ziegler, R. *Linux Firewalls*, 3rd edition. Upper Saddle River, NJ: Novell Press, 2005.

Home page for Firestarter, an iptables GUI: www.fs-security.com

Home page for the Guarddog iptables GUI: www.simonzone.com/software/guarddog



Linux - FreeBSD - x86 Solaris - MS etc.



Proven technology. Proven reliability.

When you can't afford to take chances with your business data or productivity, rely on a GS-1245 Server powered by the Intel® Xeon® Processors.

Quad Core Woodcrest



2 Nodes & Up to 16 Cores - in 1U

Ideal for high density clustering in standard 1U form factor. Up to 16 Cores for high CPU needs. Easy to configure failover nodes. Features:

- 1U rack-optimized chassis (1.75in.)
- Up to 2 Quad Core Intel® Xeon® Woodcrest per Node with 1333 MHz system bus
- Up to 16 Woodcrest Cores Per 1U rackspace
- Up to 32GB DDR2 667 & 533 SDRAM Fully Buffered DIMM (FB-DIMM) Per Node
- Dual-port Gigabit Ethernet Per Node
- 2 SATA Removable HDD Per Node
- 1 (x8) PCI-Express Per Node



Servers :: Storage :: Appliances

Genstor Systems, Inc.

780 Montague Express. # 604
San Jose, CA 95131

www.genstor.com

Email: sales@genstor.com

Phone: 1-877-25 SERVER or 1-408-383-0120



Intel®, Intel® Xeon®, Intel® Inside® are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.



JON "MADDOG" HALL

The Outer Banks

Software developers should know that even geeks sometimes want to be treated like Mom & Pop.

Many programs often are written that work great 99.999% of the time that people use them. Or, they work great for 95% of the people who want to use them. But, for those features or those people on the "Outer Banks", there is just not enough attention paid nor documentation written to satisfy their needs.

In sea lore, the Outer Banks always have held a bit of mystery and adventure. Usually the farthest piece of land or fishing area, they are the ones hardest to achieve and offer the most challenges, but they often have the greatest payback. A definition I like from the Internet calls the Outer Banks "ever-changing", "subject to the whims of the seas" and a "demanding environment".

Sometimes it feels like software is that way.

A business proposition I am involved with is based on using a distribution that is different from the one I had been using for the past four years, so I decided to switch to the new (for me) distribution. I have a philosophy that mandates if you cannot use your own products, you should not coerce others to use them.

I had to do a bit of due diligence in the effort of migration. I ran the new distribution as a live CD on my notebook, and all of the devices were found and configured correctly. Unlike my other distribution, I did

not have to go find the wireless network card driver, and various other aspects of it were set up more or less the way I wanted them. Initially, I was very impressed.

Also unlike my former distribution, this one had taken a philosophy of presenting a smaller number of applications to end users in its menus. The distribution's developers had done analysis and made decisions based on what they used and what they thought their customers might use. Understanding this philosophy, I made a decision to use their default mail interface, which was more integrated and Windows-like than the one I had been using for 15 years. I should say that nothing was wrong with the other program I had been using, but it was not as mainstream as the one I moved to, so therefore, it did not integrate in with the other applications as nicely. I also wanted to honor the above-mentioned philosophy of "as ye sow, so shall ye reap".

For the most part, I like the new mail interface. It does things differently in comparison to my old one, but it does have various nice features. It is well organized, responsive and supports a lot of nested folders—something I needed due to my habit of keeping all of my e-mail history on my notebook so I can work off-line at any time. (Yes, I do backups frequently.)

Fortunately for me, the new e-mail interface had the capability of migrating my old e-mail storage into the new format. I had seen this in the e-mail installation documentation, and I was very happy that my e-mail could be "converted over" to the new format easily. Unfortunately, when the time came to do this crucial step, the conversion program did not work. This brings me to the theme of this month's article.

I have relatively simple *needs* when it comes to writing an e-mail message, and I am sure this new interface will satisfy most of those needs when I become used to it. But, the thing I really needed from the very beginning was for the import mechanism actually to work. And, not only did this mechanism not work, but it also did not work in a *spectacular* way—in a way that made me wonder if the programmers had tried it out even one time before listing it proudly as a "feature". Or, perhaps they tried it out so long ago that over time (when it stopped working), they didn't notice it had stopped working.

Of course, importing old e-mail is typically something users (unless they are system administrators) do one time. After users have incorporated their e-mail, they go on and "just use it". For programmers to do regression testing of incorporating old e-mail takes time and effort in setting up a test bed or a methodology for

...I am sure this new interface will satisfy most of those needs when I become used to it. But, the thing I really needed from the very beginning was for the import mechanism actually to work.



PHOTO ©ISTOCKPHOTO.COM/MICHELLE MALVEN

testing those older systems to make sure the system still works in the future. Or, they continually have to find people willing to test the incorporation of their e-mail into the new system. Or, they have to wait until it fails for someone, and then try to get it working.

Unfortunately, this last strategy often gives the software overall a bad name. People who should be using the software never use it, because the *very first thing that should have worked* did not work. Most people would not be as stubborn as I am in getting something to work. They just stop using it.

Fortunately, I am not "Mom & Pop". I could look beyond the fact that the e-mail incorporation did not work properly and quickly formed a workaround for the problem. Now, I am using the new e-mail interface and will continue using it, and I will turn in a bug report about the incorporation problem.

I purposely have not mentioned either the old or the new interfaces that I am using in this article. The people who know me are aware of the e-mail interface I have been using for about 15 years. And, those people who see me using my computer will guess at the new one. But, enough projects and software exist that work 95% of the time to make this article applicable to many of them, and it is not fair to make examples of only these two.

The new interface still has some issues, and I miss some things from the old one. I intend on working with the developers of the integrated interface to incorporate the items that make sense in the "Mom & Pop" world and document how to do those things that do not make sense for the majority of the people, but that might be handy in the Outer Banks. I know I will see some of you in the "ever-changing and demanding environment". ■

Jon "maddog" Hall is the Executive Director of Linux International (www.li.org), a nonprofit association of end users who wish to support and promote the Linux operating system. During his career in commercial computing, which started in 1969, Mr Hall has been a programmer, systems designer, systems administrator, product manager, technical marketing manager and educator. He has worked for such companies as Western Electric Corporation, Aetna Life and Casualty, Bell Laboratories, Digital Equipment Corporation, VA Linux Systems and SGI. He is now an independent consultant in Free and Open Source Software (FOSS) Business and Technical issues.

Chip manufacturing,
warehouse automation,
and other throughput-intensive
systems require

**high
speed**
processing of
c-tree technology

FairCom database
technology makes
it possible.



FairCom

www.faircom.com/go/?speed



DOC SEARLS

Why an iPhone When We Can Make Our Own Open Phone?

Let's break up the cell-phone silos, for everybody's good.

I'm writing this in the aftermath of the 2007 Consumer Electronics Show. I attend CES every year, because it's always a treasure trove of interesting Linux stories and use cases (a few of which appear in the UpFront section of *LJ* this month), and also because it's always fun to see what's happening with a large old industry that's changing a lot more slowly than the annual hype suggests.

Many consumer electronics "revolutions" aren't. Such will likely be case this year with the arrival of Apple's new iPhone. I don't know if scheduling MacWorld and CES for the same week happened by accident or intent, but the effect was predictable: Steve Jobs' customarily charismatic and news-packed opening keynote at MacWorld upstaged all of CES—a tradeshow exceeded in size only by Europe's CeBIT.

The biggest news in Steve Jobs' speech was the iPhone. He called it "a revolutionary product...that changes everything". He said it would cause a revolution on the scale of the Macintosh in 1984 and the iPod in 2001. He even said the iPhone qualified as not one but "three revolutionary products". These were 1) "a wide-screen iPod with touch controls", 2) "a revolutionary mobile phone" and 3) "a breakthrough Internet communications device". He contrasted it with "smartphones", such as the Trio, Blackberry, Nokia E62 and Moto Q, all of which feature keyboards that "are there whether you want them or not".

The iPhone is faced with a large, sharp color screen and a patented pointing system called MultiTouch that lets you use multiple fingers to do all kinds of stuff. (Except, of course, punching phone numbers without looking at them, because all the numbers are displayed behind a layer of clear tactile camouflage.) In the Apple tradition, controls are minimal; there's a single button on the front and few others elsewhere.

At the top of techie conversation at CES was news that the phone would run on Apple's BSD-based OS X and support "desktop class" applications. That claim, and the one about iPhone being a "revolutionary Internet communications device", fueled hope that Apple would help break the cell-phone industry out of the phone-maker/carrier silos that have trapped customers inside and kept independent developers outside for the duration. In a post on the *Linux Journal* Web site, I wrote:

Knock what's closed about the iPhone all you want; it's still a computer with a mike, a screen, a speaker and a pile of other input and output openings that invite developments of many kinds. That's why I think iPhone is going to make the cell-phone market a lot bigger. It will encourage participation by developers and customers that have until now been forced to cope with far less than they've wanted from the cell-phone industry. And that includes all the legacy cell-phone players with which Apple now partners or competes.

I should have known better. In fact, I did, but ignored my inner cynic.

Back in 1997, when Steve Jobs returned to Apple after a long hiatus, one of his first moves was to kill off clones of the company's hardware. In the midst of the outcry that followed, I wrote this to Dave Winer, who published it on his own site (www.scripting.com/davenet/stories/DocSearlsonSteveJobs.html):

So Steve Jobs just shot the cloners in the head, indirectly doing the same to the growing percentage of Mac users who preferred cloned Mac systems to Apple's own. So his message to everybody was no different than it was at Day One: all I want from the rest of you is your money and your appreciation for my Art.

It was a nasty move, but bless his ass: Steve's Art has always been first class, and priced accordingly. There was nothing ordinary about it. The Mac "ecosystem" Steve talks about is one that rises from that Art, not from market demand or other more obvious forces. And that Art has no more to do with developers, customers and users than Van Gogh's has to do with Sotheby's, Christie's and art collectors.

See, Steve is an elitist and an innovator, and damn good at both. His greatest achievements are novel works of beauty and style. The Apple I and II were Works of Woz; but Lisa, Macintosh, NeXT and Pixar were all Works of Jobs. Regardless of their

market impact (which in the cases of Lisa and NeXT were disappointing), all four were remarkable artistic achievements. They were also inventions intended to mother necessity—and reasonably so. That's how all radical innovations work. (Less forward marketers, including Bill Gates, wait for necessity to mother invention, and the best of those invent and implement beautifully, even though that beauty is rarely appreciated.)

To Steve, clones are the drag of the ordinary on the innovative. All that crap about cloners not sharing the cost of R&D is just rationalization. Steve puts enormous value on the engines of innovation. Killing off the cloners just eliminates a drag on his own R&D, as well as a way to reposition Apple as something closer to what he would have made the company if he had been in charge through the intervening years.

The simple fact is that Apple always was Steve's company, even when he wasn't there. The force that allowed Apple to survive more than a decade of bad leadership, cluelessness and constant mistakes was the legacy of Steve's original Art. That legacy was not just an OS that was ten years ahead of the rest of the world, but a Cause that induced a righteousness of purpose centered around a will to innovate—to perpetuate the original artistic achievements. And in Steve's absence, Apple did some righteous innovation too. Eventually, though, the flywheels lost mass and the engine wore out.

In the end, by when too many of the innovative spirits first animated by Steve had moved on to WebTV and Microsoft, all that remained was that righteousness, and Apple looked and worked like what it was: a church wracked by petty politics and a pointless yet deeply felt spirituality.

Now Steve is back, and gradually renovating his old company. He'll do it his way, and it will once

again express his Art.

These things I can guarantee about whatever Apple makes from this point forward:

1. It will be original.
2. It will be innovative.
3. It will be exclusive.
4. It will be expensive.
5. Its aesthetics will be impeccable.
6. The influence of developers, even influential developers like you, will be minimal. The influence of customers and users will be held in even higher contempt.
7. The influence of fellow business artisans, such as Larry Ellison (and even Larry's nemesis, Bill Gates), will be significant, though secondary at best to Steve's own muse.

Ten years later, I can look back on that as one of the most prophetic pieces I've ever written.

Steven Levy of *Newsweek* (and the author of *Hackers* and many other books) reported this about his conversation with Jobs after the iPhone announcement:

But it's not like the walled garden has gone away. "You don't want your phone to be an open platform", meaning that anyone can write applications for it and potentially gum up the provider's network, says Jobs. "You need it to work when you need it to work. Cingular doesn't want to see its West Coast network go down because some application messed up."

Hmm...I have a Trio here that's full of third-party apps that don't bring down Verizon's network. Other Trios run third-party apps that don't bring down Cingular's network either. Still, whether or not Jobs is bogus on the subject of apps and networks, he clearly wants to keep the iPhone closed from outside developers.

This makes sense. In spite of Apple's support for open source and open standards

Linux Laptops

Starting at \$799



Linux Desktops

Starting at \$375



Linux Servers

Starting at \$899



**DON'T BE SQUARE!
GET CUBED!**



R³ Technologies
Making World-Wide Technologies a Life

309.34.CUBED
shopcubed.com

Still, whether or not Jobs is bogus on the subject of apps and networks, he clearly wants to keep the iPhone closed from outside developers.

(and its contributions are not trivial), the company always has played closed games with customers and developers. The original Macintosh was so closed to customers that opening it required a long-shaft torx screwdriver and a special case spreader. And, developers have rarely had much choice other than to work exclusively in Apple's development environments.

Now, in the case of the iPhone, there won't be a development environment. Apparently, I do know a number of people who think Apple is just stalling at this point, because the phone isn't due out until June, and it isn't in a position yet to produce an SDK. But, even if that's the case, the facts on the face of this thing make one thing very clear: Apple isn't going to bust any phone-maker/carrier silos. On the contrary, it's going to build a new silo of its own with Cingular—the carrier with which it signed an exclusive partnership deal. So, it's just another jail with a prettier lock.

Now, what to do?

A hint came in the form of a story in the *New York Times*, about how passengers in Japan use cell phones to speed check-in at airports. The airline ANA is using a system called skip in Japan and is working with Star Alliance partners (which include United, Lufthansa and British Airways) on extending the service, which does away with paper boarding passes. What's interesting about this isn't the system, but the fact that airlines use it for relating directly to customers, regardless of phone-maker or cell system carrier. In other words, it's not about a deal between ANA and Nokia/Verizon or Motorola/Sprint. It's a way for ANA to relate directly with customers.

This hint encourages development of apps that disintermediate phone-makers and pipe-controllers by putting customers and vendors into direct contact, for the good of everybody involved.

Cell phones are much more personal than computers. In fact, they may be the most personal technology ever created—as well as the most social. Why should the market benefits of cell phones' personal and social powers be restricted to phone-maker/carrier silo partners? In the long run, they can't. There are simply too many benefits for too many businesses—as well as customers—once these silos open up.

Can we get a sense of how many more market categories there can be, and how much more business will grow around cell phones, once the silos open up?

Yes—by looking at vertical market examples. A good one is the university-student cell-phone market. Here, a company called Rave Wireless (disclosure: I consult them) works with universities to replace their once-lucrative but now-dead wireline phone business with their own cell systems. The phones might be made by Nokia (or anybody) and the carrier might be Cingular (or anybody), but the system is independent of both. Instead, it exists for the purpose of serving relationships within the university community—between teachers and students, students and each other, students and local businesses, sports teams

and fans. Rave not only provides a raft of handy (even essential) applications for its phone users, but it also provides a platform where students (or anybody using Rave phones) can write their own applications.

Students can form "entourages" of groups around classes, fraternities, dorm floors or whatever social collections they like. Teachers can text students with schedule changes. Students can text local businesses to see, for example, which pizza parlor can set a table for nine right after a game is over. They can check bus schedules or use built-in GPS monitoring when sending an emergency message to campus police. The list of applications developed by both Rave and students is long and growing. This is made possible not only by Rave's entrepreneurial smarts, but by freedom from restrictions imposed by the customary phone-maker/carrier silo agreements as well.

Rave can drive system-opening deals with both phone-makers and carriers, because it comes to both with a large base of ready customers. It turns out that phone-makers will make a custom phone if the order is big enough. And, it also turns out that carriers will open their systems for the same reason. Both still make money—but not just with each other and their co-captive customers. Instead, they open a whole new market ecosystem that gets bigger for everybody.

I normally avoid writing about companies I consult, but the example Rave Wireless provides is too important to overlook. And, I'm not hustling them. Instead, I'm hustling something Rave's example encourages us to think about: an open-phone marketplace, populated by rapidly evolving and differentiating phone gear—with a proliferation of applications to run on it and services to support it. In the long run, that's where we're headed anyway. I'd like us to shorten the distance.

Where can we start? One place is with phones. I'm familiar with two open Linux-based phone platforms: Trolltech's Greenphone and the OpenMoko (profiled in the February 2007 issue of *Linux Journal*). There can be many more, including gear from the familiar makers.

But, let's go beyond that. Let's find whole communities that already relate and could relate much better with cell phones equipped with community- and commerce-supporting applications. These could be localities (towns, for example), professions (engineers, educators, health-care or service workers) or organizations (professional or lifestyle associations, unions, political parties). Or, hey, how about Free Software and Open Source Development communities? Why not?

We not only have strength in numbers, we have the power to produce a plethora of useful applications. (Try saying that fast.)

Again, it's going to happen anyway. Won't it be a lot more fun to *make* it happen? And, isn't that what we're about? ■

Doc Searls is Senior Editor of *Linux Journal*. He is also a Visiting Scholar at the University of California at Santa Barbara and a Fellow with the Berkman Center for Internet and Society at Harvard University.

Gemini

2U

– Ultra Dense Series



- Two fully independent systems in a 2U
- Ability to run two discrete operating systems in one box
- Up to 16 CPU cores
- Up to 12 hot-swappable SATA, SCSI, or SAS hard drives
- RAID 0, 1, 5, 6, 10, 50 available on both systems
- Opteron™ or Xeon™ multi-core processors
- Up to 64GB memory per motherboard
- One available PCI-X or PCI-E slot per motherboard
- High efficiency AC and DC power options

2 Systems in One

Built on open standards, the Gemini 2U elegantly accommodates two discrete motherboards in a 25" chassis uniquely designed for easy access from the rear.

Gemini 2U represents the realization of intoxicating power and superior environmental specifications, with considerably less power consumption, less heat and less noise. Remarkably, it all fits nicely into any standard rack. Front to back, Gemini 2U is both powerful and efficient.

At Open Source Systems we understand you need practical, customizable, and affordable solutions that are easy to manage and maintain.

For more information and to request your evaluation unit today, visit us at www.OpenSourceSystems.com, or call direct at 866.664.7867.



Patent Pending



 **Open Source
Systems™**



Digium's Asterisk-Based Solutions

By the time you read this, Digium will have three new Asterisk-based solutions for your telephony-based enjoyment. The first of these is Digium's TDM800P, an eight-port analog telephony interface card with Digium's VoiceBus technology, which is built on a single PCI bracket for universal PCI compatibility. Together with Digium's Asterisk software and a standard PC or server, users can create an inexpensive and scalable telephony solution comparable to a high-end PBX platform. The second item is the TE120P, a single span, selectable T1 (24-channel), E1 (32-channel), or J1 (24-channel) card that routes voice and data simultaneously, eliminating the need for an external router. The TE120P also delivers PBX and IVR services including voice mail, call conferencing, three-way calling and VoIP gateways. Last but not least is the software-based, G.168-compliant High Performance Echo Canceller (HPEC) for 32-bit and 64-bit Linux platforms. The HPEC provides echo cancellation for configurable tail lengths of 16ms (128 taps), 32ms (256 taps), 64ms (512 taps) and 128ms (1,024 taps), on a per-channel basis.

www.digium.com

3Com's Open Services Networking Platform

Doc Searls' and the Linux Community's call for open architectures all over is gaining nice momentum. A case in point is 3Com's Open Services Networking (OSN) Platform, the firm's strategy to base its network solutions on an open, interoperable, multivendor architecture. The idea is to "let organizations, rather than their networking vendors, select the applications that best match business requirements and implement them with timeliness and efficiency", say the 3Com-ers. Lucky for us, a key component of OSN is the Open Source Service Monitoring bundle, which leverages a number of pretested and supported open-source-based network and service management applications, such as MRTG, NTOP, TShark and Nagios. The open-source bundle is currently available for download from 3Com's Web site.

www.3com.com/osn



John Brosnan and Kyle Copeland's *Beginning TiVo Programming* (Wrox Press)

To the delight of hackers far and near, TiVo made it possible to create applications for its popular digital video recorder (DVR) product. *Beginning TiVo Programming* by John Brosnan and Kyle Copeland is targeted at programmers who want to gain a "complete understanding of all the pieces that make up a TiVo application". The book uses real-life code examples to guide the reader through the steps needed to implement an application. The team of Brosnan and Copeland designed its own application for HME, which is the code name for TiVo's powerful new open platform for applications. The latest SDK for HME can be found at tivohme.sourceforge.net.

www.wrox.com

Please send information about releases of Linux-related products to James Gray at newproducts@linuxjournal.com or New Products c/o *Linux Journal*, 1752 NW Market Street, #200, Seattle, WA 98107. Submissions are edited for length and content.



Levanta's Intrepid X Linux Management Appliance

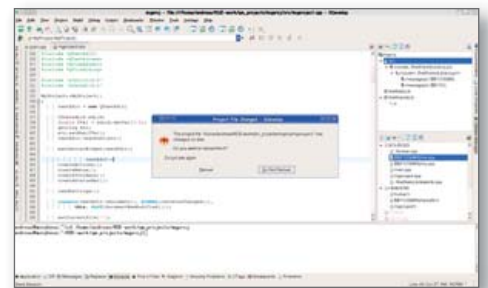
The Levanta folks recently launched a new Linux management appliance, called Intrepid X. Targeted at "large Linux departments and data centers with high scalability and mission criticality requirements", Intrepid X offers "on-demand functionality, disaster recovery, system portability, complete change control, unattended active/passive fail-over and interoperability with iSCSI or Fibre-Channel SANs for RPM-based Linux environments". Levanta says that customers who already preside over virtual storage and want to leverage their SANs to get the increased speed, portability, manageability and disaster recovery advantages found in Linux will benefit. The Intrepid X can be paired with different types of SAN storage, including products from EMC, HDS, IBM and others. In addition, one can manage Linux running in VMware virtual machines, as well as both physical and virtual environments from one interface.

www.levanta.com

The KDevelopment Team's KDevelop

The ambitious KDevelopment Team has released version 3.4 of KDevelop, a powerful, language-independent, user-friendly integrated development environment—that's not just for programming KDE apps. Version 3.4 is the first new release in more than a year, closing more than 500 bugs and adding several new features. New features include improved Qt 4 support, new debugging capabilities, an enhanced default user interface layout, improvements for C++, and Ruby and PHP support. Official KDevelop packages are available for Kubuntu and OpenSUSE; unofficial builds also are available for other distros. One of our fellow Linux media outlets recently called KDevelop one of the top "killer apps" on the Linux platform.

www.kdevelop.org



Fluendo's Media Codecs for Open Source Systems

The mellifluously named firm Fluendo of Spain recently released proprietary codecs for Windows Media Player (audio, video, MMS streaming protocol), MPEG-2 and MPEG-4 for the Linux and Solaris desktop and server platforms. Fluendo says that "agreements with Microsoft and MPEG LA" take the solution out of the legal limbo in which other codecs are entangled. Fluendo's codecs are closely integrated with the GStreamer multimedia framework, supporting applications such as Totem, Elisa, Jokosher, Rhythmbox and Banshee. Fluendo will release further codecs during 2007; existing codecs are available for purchase from Fluendo's Web site.

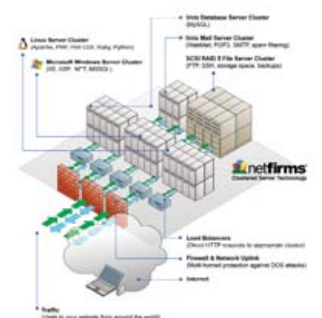
www.fluendo.com

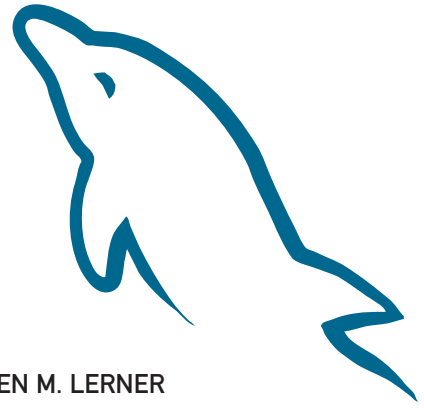


Netfirms' Business/Enterprise Web Hosting Accounts

Although many Web hosting companies let you choose whether to host your Web site on Linux or Windows, Netfirms says it's the first to unite the two platforms under the umbrella of a single account. With its Business and Enterprise lines of hosting accounts, Netfirms has Windows-based applications executed natively on a Windows Server 2003 grid and Linux-based applications natively on a Linux server grid. "The two platforms are unified through proprietary clustered technology", says Netfirms, "allowing customers to manage both transparently through a single account". The Netfirms multiplatform server technology offers Windows-based functionality via the Microsoft Hosting Partner Program, including ASP.NET, Classic ASP and Microsoft SQL Server. On the Linux side, the full LAMP stack (Linux, Apache, MySQL and PHP) is available. The two lines encompass a wide range of hosting options depending on the client's needs.

www.netfirms.com





MySQL Deserves a Double Take

What you don't know about MySQL could hurt you. REUVEN M. LERNER

In early 1995, when Windows 95 was still vaporware and Red Hat was the upstart, user-friendly alternative to Slackware, I worked for the "Pathfinder" site at media giant Time Warner. Like all media companies, Time Warner realized that the Web was going to take off, but wasn't sure just how that would happen.

So, it hired a bunch of programmers and designers, and gave us the opportunity to experiment with different types of designs and applications. It was a wonderful job, with some creative, smart and interesting people. And, during my time there, I created all sorts of applications—quizzes, mail auto-responders, games, search engines and even a personalized version of *Money* magazine's "Best Cities" rankings.

As the applications I built became increasingly sophisticated, it became obvious that the text files I often used for data storage and retrieval were neither efficient nor flexible enough for a site as large and popular as ours. Finally, someone introduced me to our newly hired database guru, who taught me about the wonders of relational

month, I discuss MySQL, including its use, features and problems. The next article will include a similar analysis of PostgreSQL, and the third article in the series will compare the two databases.

Starting with MySQL

One of MySQL's claims to fame is the ease with which people can get started using it. And, indeed, when you compare MySQL with many commercial databases, it is strikingly simple. You install it (typically with an RPM or Deb, but compiling it from source is also straightforward), and start up the database server with `safe_mysql.d`. (You also could use the plain `mysqld` command, but then you wouldn't benefit from some of the behind-the-scenes housekeeping that `safe_mysql.d` offers.)

Once you have started the server, you can create one or more databases. (I admit it is somewhat confusing that MySQL is often referred to as a database when, in fact, it is a database server, offering you the chance to create one or more databases. Each database

Recent versions of MySQL also offer the ability to create a stored procedure or function, which provides both increased speed and centralized control over commonly used functions.

databases and SQL. I was hooked, and I enjoyed working with the database server that we had installed.

When I wanted to reap the benefits of SQL on my Linux box at home, my options were, unfortunately, limited. I found a number of abandoned open-source database projects, but nothing that was as powerful as Time Warner's Sybase server or even in the same league.

So, you can imagine my delight when I discovered MySQL. No, it didn't do all the things that Sybase did, and it wasn't released under an open-source license. But, it was free of charge, it was easy to install and it had enough features to keep people like me relatively happy. Internet service providers felt similarly, and began to install it on their systems—first as a competitive advantage over their rivals, and then because everyone else was including it in the base configuration.

Fast-forward more than a decade, and MySQL is by far the best-known open-source relational database. Monty Widenius and David Axmark, whom I met back when they were the only full-time MySQL programmers, are now at the top of a large corporate pyramid. MySQL AB now distributes its products under the GNU General Public License (GPL), with a closed-source license available to those who require it. It runs, as always, on a very large number of different operating environments. And, it is still developed at a feverish pace by people around the world, who submit patches and suggestions.

This is the first of three articles on open-source databases. This

contains one or more two-dimensional tables.) To create a database, use the `mysqladmin` program:

```
mysqladmin create testdb
```

It is quite possible, depending on your configuration, that the above command worked without a hitch—particularly if you are logged in as the root user under Linux. However, your system administrator might have (wisely) decided to set a password for the MySQL root user, in which case, you need to type:

```
mysqladmin -p create testdb
```

The `-p` option tells `mysqladmin` that you want to enter a password for this account. You also can specify the root user, or any other user, with the `-u` option, as in:

```
mysqladmin -u mysqlroot -p create testdb
```

Once you have created a database, you then can connect to it with the `mysql` client program:

```
mysql -u mysqlroot -p testdb
```


Notice that I'm once again specifying a user name and that I want to enter a password. I return to the subject of permissions below; for now, we assume that this combination works.

In the client, you can issue any SQL command you want, and it will be executed immediately. For example, we can create a new table:

```
CREATE TABLE Classes (
class_name TEXT NOT NULL,
room_number INTEGER NOT NULL,
starting_date DATE NOT NULL,
ending_date DATE NOT NULL,
instructor TEXT NOT NULL
);
```

One of the problems with the above table is that it lacks a unique primary key. This makes it difficult to refer to the Classes table from another table. We could use the name assigned to the class by the university's registration system, but there is no guarantee that this will be unique. Moreover, what will we do next year, when a class of the same name is offered? For this reason (among others), it's the norm to create an "artificial" primary key, one whose purpose is to identify a row within the database uniquely.

In MySQL, we can do this most easily with the AUTO_INCREMENT keyword. For example:

```
CREATE TABLE Classes (
class_id INTEGER AUTO_INCREMENT,
```

```
class_name TEXT NOT NULL,
room_number INTEGER NOT NULL,
starting_date DATE NOT NULL,
ending_date DATE NOT NULL,
instructor TEXT NOT NULL,

PRIMARY KEY(class_id)
);
```

If we want, we can INSERT a row into Classes with an explicit integer value for class_id. The fact that class_id is defined as a primary key means that it is both indexed and guaranteed to be unique. But, if we fail to enter an explicit value for class_id, MySQL inserts a new value into the column, giving us a primary key value for the new class without having to calculate it ourselves.

The above table definition shows a few of the many data types MySQL offers. MySQL offers many traditional data types, such as NUMERIC and VARCHAR, but it also includes a number of signed and unsigned numeric types (for example, TINYINT, SMALLINT, MEDIUMINT, INT and SIGINT), a number of CLOB/BLOB types (such as, CHAR, BINARY, BLOB and TEXT), and several having to do with dates and times (DATE, DATETIME and TIMESTAMP). There are also ENUM and SET types, allowing you to work with nonstandard sets of enumerated data.

MySQL also offers a wide variety of operators, from simple string-concatenation, to date extraction, to one of my favorites, the CASE statement, which lets you place if-then logic inside of a query.

In addition, MySQL offers a system for full-text search. This means you

Hurricane Electric Internet Services... **Speed and Reliability** You Can Depend On!

**Flat Rate
Gigabit Ethernet**

1,000 Mbps of IP

\$13,000/month*

**Full 100 Mbps
Port**

Full Duplex

\$2,000/month


**Colocation Full
Cabinet**

Holds up to 42 1U
servers

\$400/month

Order Today!

email sales@he.net or call 510.580.4190

 he.net

* Available at PAIX in Palo Alto, CA; Equinix in Ashburn, VA; Equinix in Chicago, IL; Equinix in Dallas, TX; Equinix in Los Angeles, CA; Equinix in San Jose, CA; Telehouse in New York, NY; Telehouse in Los Angeles, CA; Telehouse in London, UK; NIKHEF in Amsterdam, NL; Hurricane I and Hurricane II in Fremont, CA, and Hurricane in San Jose, CA

can store text inside of TEXT columns in your tables, and then identify the column (and retrieve the text) without having to index it yourself.

If the included suite of functions doesn't suit your needs, you can always write one of your own. Recent versions of MySQL also offer the ability to create a stored procedure or function, which provides both increased speed and centralized control over commonly used functions. Stored procedures also can be invoked automatically when particular events occur, known as a trigger in database parlance. You also can write new functions in C or C++, loading them into MySQL at runtime.

Table Types

So far, MySQL sounds like a nice, flexible relational database. You might be surprised, however, to find that there is a huge amount of pent-up frustration, and even hostility, toward MySQL in the Open Source and Database communities. Just look for a recent story on Slashdot about MySQL, and you will see many comments indicating that PostgreSQL, Firebird or nearly any other option would be a better solution.

Part of this stems from a time-honored tradition of rivalry in the computer world, and particularly in the Open Source community. Over the years, we have seen fights between Emacs and vi, Perl and Python, Linux and BSD, and countless other pairings.

But, part of the animosity toward MySQL stems from several design decisions that the authors made early on. For example, documentation for an old version of MySQL said that foreign keys are really unnecessary, and that such integrity checks could (and should) be handled in the application, rather than in the database. Many experienced database people see this and don't know whether to laugh or cry. The primary reason for using

Much has been made about MySQL's fast performance over the years, with little or no tuning of the server. The truth is a bit hazier than that; although MySQL is undoubtedly a fast database, many of those tests were made using MyISAM tables, which are inherently faster because of their lack of locks and integrity checks. (As an analogy, I often say that it's faster to leave your house without locking the door, but the extra speed is usually not worth the risk.)

Scalability

Many of the features in recent versions of MySQL have been aimed at corporate customers, whose license purchases are helping drive MySQL development forward. One of the biggest bottlenecks that a database administrator can face, particularly as the data grows in size, is disk speed. Recent versions of MySQL thus provide both tablespaces (that is, allocation of disk space on a per-table basis) and partitions (that is, division of a table across several filesystems). Tablespaces are available only with InnoDB tables, but partitions are available for all storage engines. Moreover, tables can be partitioned based on column values, using a hash function to decide into which partition a particular row should be placed.

Another important aspect of MySQL has been replication and backup. These are crucial features for enterprise clients, who need their data to be available all the time and to have backups available at a moment's notice. The latest versions of MySQL have improved the replication engine and have also made it more flexible, making it possible to replicate tables even on a per-row basis.

Another feature I have been waiting to see for some time is Unicode support. Although not all string and regexp operations

Perhaps the biggest asset that MySQL has going for it is a very large, very active community of users and developers.

a database is for its reliability, not speed, and adding foreign-key checks is an easy way to increase the reliability of inserted data.

Similarly, old versions of MySQL failed to lock tables. If you wanted to be sure that no one would write to a table from which you were reading (or to which you were writing), you needed to lock the table explicitly at the application layer. Given the many years of research that had gone into row-level locking (and even more-advanced systems, such as multiversion concurrency control), this seemed to many like a step backward.

MySQL's solution to these problems has been a novel one. Rather than add these features to the existing (MyISAM) table structure, it made it possible to choose from a number of different table structures, each with its own set of trade-offs. Much as Linux system administrators can choose from a variety of filesystems, MySQL administrators and programmers can choose from several different storage engines.

This approach has some problems, of course. The biggest problem from my perspective is that MyISAM remains the default storage engine, which means that many users effectively choose to go without foreign keys and sophisticated locking due to ignorance. Many other storage engines seem to be of more limited use or for particular applications, such as MEMORY (for in-memory databases), BDB (Berkeley DB-based) tables and even FEDERATED (for tables on remote servers).

A very popular storage engine, InnoDB, has a different problem associated with it—the company that develops InnoDB was purchased by Oracle earlier this year. This may have no effect on MySQL's open-source distribution, because Oracle continues to make InnoDB available under the GPL. But, it has raised some questions regarding MySQL's commercial version, given that an essential part of the commercial-grade toolbox is now owned by a major database rival.

work with Unicode, this is a big boon to those who work with multiple languages.

Community

Perhaps the biggest asset that MySQL has going for it is a very large, very active community of users and developers. The sheer number of books, Web sites, mailing lists, help forums and code snippets for MySQL is overwhelming.

For its part, MySQL AB has been doing an admirable job of updating the documentation on a regular basis and of moving forward with new features at an impressive rate. (This demonstrates that although open-source software can often be written by volunteers, having paid professionals work on a project can speed it up immensely.) In particular, I am impressed by the on-line documentation, which includes not only numerous examples, but also intelligently placed links to related subjects.

Conclusion

MySQL has grown up quite a bit since I first began to use it more than ten years ago. Some of its quirks, such as using MyISAM tables by default, continue to rankle serious database users who would like to see transactions and foreign keys everywhere. But, especially with versions 5.0 and 5.1, MySQL is looking like a database that can advertise its depth of serious features, rather than claim its main advantage is speed. ■

Reuven M. Lerner, a longtime Web/database consultant, is a PhD candidate in Learning Sciences at Northwestern University in Evanston, Illinois. He currently lives with his wife and three children in Skokie, Illinois. You can read his Weblog at atneuland.lerner.co.il.

KEEP YOUR BUSINESS RUNNING SMOOTHLY

PROTECT YOUR SMALL BUSINESS WITH THE BUILT-IN SECURITY ENHANCEMENTS OF THE DUAL-CORE INTEL® XEON® PROCESSOR IN YOUR SERVERSDIRECT



Big Solutions for Small Business



Intel® Server Compute Blade SBXD132

Simplify your Data Center Management



- ▶ High-performance blade server with memory and I/O expansion options
- ▶ Improved performance, performance per watt, operational effectiveness, deployment flexibility, and simplified management
- ▶ Optimizing rack space, significantly reducing cabling, and lowering total cost of ownership

SDR-5015M-T+B
1U Entry Level Server

SDR-2501T
2U Application Server

SDR-3500T
3U Database Server

SDR-5500T
5U Advanced Storage Server



Cost Effective 1U Entry Level Server
 Optional Core 2 Duo processor support

Cost Effective 2U Server
Application Server

3U servers support high availability
storage and mission critical
business applications

5U Storage servers, ideal for email,
database or other high capacity demand
applications. Support up to 18TB drive

- ▶ Support Xeon® 3000 Series, Pentium® D, Pentium 4, Pentium Extreme
- ▶ Up to 8GB unbuffered ECC / non-ECC DDR2 667/533MHz SDRAM
- ▶ 2x 1" Hot-swap SATA Drive Bays
- ▶ 1x Intel® 82573L PCI-e Gigabit LAN
- ▶ 300W Power Supply

- ▶ Support Dual Intel® 64-bit Xeon® Quad-Core or Dual-Core, with 667 / 1066 / 1333 MHz FSB
- ▶ Up to 16GB DDR2 667 & 533 SDRAM Fully Buffered DIMM (FB-DIMM)
- ▶ 6 x 1" Hot-swap SATA Drive Bays
- ▶ Intel® (ESB2/Gilgal) 82563EB Dual-port Gigabit Ethernet Controller
- ▶ 600W Power Supply

- ▶ Support Quad & Dual Core Intel® 64-bit Xeon® Support, 667 / 1066 / 1333MHz FSB
- ▶ Up to 64GB DDR2 667 & 533 SDRAM Fully Buffered DIMM (FB-DIMM)
- ▶ 16 x 1" Hot-swap SATA Drive Bays
- ▶ Intel® (ESB2/Gilgal) 82563EB Dual-port Gigabit Ethernet Controller
- ▶ Redundant 800W Power Supply

- ▶ Support Quad & Dual Core Intel® 64-bit Xeon® Support, 667 / 1066 / 1333MHz FSB
- ▶ Up to 32GB DDR2 667 & 533 SDRAM Fully Buffered DIMM (FB-DIMM)
- ▶ 24 x 1" Hot-swap SATA Drive Bays
- ▶ Intel® (ESB2/Gilgal) 82563EB Dual-port Gigabit Ethernet Controller
- ▶ 950W Triple Redundant Power Supply

STARTING PRICE \$999

STARTING PRICE \$1,299

STARTING PRICE \$2,099

STARTING PRICE \$3,799

SERVERS DIRECT CAN HELP YOU CONFIGURE YOUR NEXT HIGH PERFORMANCE SERVER SYSTEM - CALL US TODAY!

Our flexible on-line products configurator allows you to source a custom solution, or call and our product experts are standing by to help you assemble systems that require a little extra. Servers Direct - your direct source for scalable, cost effective server solutions.

1.877.727.7887 | www.serversdirect.com

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, Pentium, and Pentium III Xeon are trademarks of Intel Corporation or it's subsidiaries in the United States and other countries.

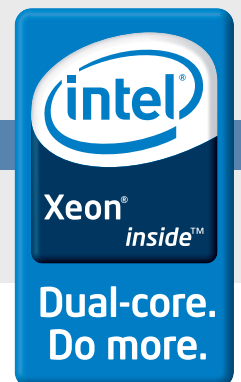




ILLUSTRATION © ISTOCKPHOTO.COM/MIEMRAH TURUDU
SECURITY LOCK ICON © ISTOCKPHOTO.COM/JAMIE CARROLL



Single Packet Authorization

Single Packet Authorization fills the gaps in port knocking. MICHAEL RASH

Countless pieces of software, protocols and complex interdependencies together form a system for which it is difficult to guarantee any particular property—particularly security. Even software specifically designed to enhance security can, at the behest of clever individuals armed with detailed knowledge, work to its detriment. Vulnerabilities have been discovered in all sorts of security software from firewalls to implementations of the Secure Shell (SSH) Protocol. For example, OpenSSH is developed by some of the most security-conscious developers in the world, and yet it occasionally contains a remotely exploitable vulnerability. This is an important fact to note because it seems to indicate that security is hard to achieve and, therefore, bolsters the case for a defense-in-depth approach. This article explores the concept of Single Packet Authorization (SPA) as a next-generation passive authentication technology beyond port knocking.

When an attacker is on the prowl in an attempt to exploit a vulnerability in server software (as opposed to client software), the first step is reconnaissance; the attacker needs to locate a target. This process has been brilliantly automated by Nmap, so it is easy to construct a list of target systems that may be ripe for compromise. If the attacker has found a zero-day vulnerability in server software that you happen to be running, you don't want to appear in this list of targets! Both port knocking and Single Packet Authorization use a packet filter configured in a default-drop stance and simultaneously provide service only to those IP addresses that can prove their identity via a passive mechanism. No TCP/IP stack access is required to authenticate remote IP addresses via this passive means. Nmap cannot even tell that a server is running when protected in this way, and it does not matter even if the attacker has a zero-day exploit.

This article is the first of a two-part series on Single Packet Authorization, and it lays the theoretical foundation for Single Packet Authorization and why it is a next-generation passive authorization technology beyond port knocking. The next article will provide a hands-on look at using fwknop to provide Single Packet Authorization protection for your SSH daemon.

Introduction to Port Knocking

Port knocking is a first-generation technology that uses the port fields within TCP and UDP packet headers to communicate information. Normally, these protocols are used to encapsulate application layer data, but port knocking encodes information in sequences of packets to various ports by using the port numbers themselves as fields to transmit data. These packets are typically either monitored out of a firewall log or via a packet capture mechanism, such as libpcap. Typically, there is a port knocking client and a port knocking server. The terms client and server, in this case (and throughout the remainder of this article unless otherwise noted), refer to the software components that send and monitor packets, respectively. The client is responsible for generat-

ing the port sequences, and the server is responsible for passively collecting the sequences and reconfiguring the packet filter to allow connections to protected services upon receipt of a valid sequence.

The typical port knocking scenario is for a port knocking server to configure a packet filter to block all access to a service, such as SSH, until a specific port knock sequence is sent by a port knocking client. For example, the server could require the client to send TCP SYN packets to the following ports in order:

- > 23400
- > 1001
- > 2003
- > 65501

If the server monitors this knock sequence, the packet filter reconfigures to allow an SSH connection from the IP address that sent it. By making use of a connection tracking mechanism provided by the packet filter (such as the conntrack system in Netfilter), an SSH session can remain established after the initial rule created by the knock server is removed after a timeout. Port knock sequences can be encrypted, and there are many implementations listed at www.portknocking.org. For a graphical representation of port knocking in action, see Figure 1.

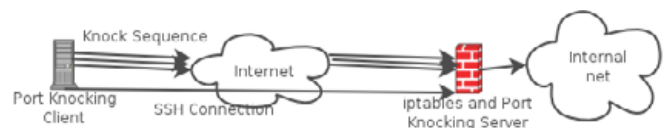


Figure 1. Port Knocking in Action

Port Knocking Limitations

Port knocking offers some real benefits for limiting access to services, but where do the limitations lurk? First, it is clear that encrypting knock sequences is important, and this in turn implies that several bytes of information must be transmitted. For symmetric crypto systems, the encrypted data will be at least as large as the block size (128 bits for the Rijndael symmetric block cipher chosen for the Advanced Encryption Standard). For asymmetric crypto systems, the encrypted data will be substantially larger.

For instance, the raw ElGamal algorithm used by GnuPG doubles the plain-text size when encrypting data. Even though GnuPG also utilizes compression (which can sometimes reduce the size of the cipher text to below the original size of the plain text), the typically large key size of GnuPG keys implies that the cipher text for even the smallest messages will be in the hundreds of bytes.

This has important implications for port knocking. Each packet within a port knock sequence can send only two bytes of information due to the 16-bit-wide port fields in the TCP and UDP headers. (This assumes that other fields within packet headers are not also used to

This far outstrips the data transmission rate possible with port knocking, and having easy access to this amount of packet data opens up a huge range of possibilities.

transmit data. However, even if other fields are used, this still cannot result in nearly as much data transmission as using packet payloads.) Hence, for a block cipher, an encrypted sequence must contain at least $B/(2*8)$ packets, where B is the block size in bits. This by itself would not be so bad when considering the general speed and reliability of today's networks, but the real issue is out-of-order delivery.

Decrypting garbled data results in garbled data, and because there is no notion of a "connection" (in the TCP sense) between the port knock client and server, the server has no ability to re-order out-of-order packets.

Packets may take different routing paths, some of which may be slow. Hence, the client must resort to an artificial mechanism to try to reduce the potential for out-of-order delivery: time. By introducing a time delay between each packet in a knock sequence, say on the order of a half second, packet order usually can be maintained by the time the packets reach the server. Now, for a block size of 128 bits, the corresponding port knock sequence is $128/(2*8) = 8$ packets. By factoring in the half-second delay, it takes four seconds just to transmit the sequence. For a much larger message, such as those that would be generated by an asymmetric cipher, this data transmission rate is simply not practical.

Having a limited ability to transmit data introduces another limitation in port knocking schemes. It is difficult to guard against a replay attack effectively. Anyone who can monitor a knock sequence as it is sent from the client to the server is free to replay the sequence against the server in an effort to gain the same access. This is an especially important issue if the sequence is sent through a NAT device, and the source IP that is allowed through the packet filter at the server side is the external NAT address. For example, if the port knock client is on an RFC 1918 subnet, say 10.10.1.0/24, and the port knock server is on a remote network that is accessible only over the open Internet, the server must allow access to the NAT IP address. Anyone on the same subnet who can replay the sequence will be granted the same level of access. Also, anyone on the same subnet has the same level of access once a rule is instantiated to accept connections from the NAT address as long as the rule exists (no sequence replay is required in this case, and this remains true for SPA as well).

There have been variations made on traditional port knocking to try to provide a solution for the replay problem, such as making time a significant factor, using S/Key-style hash function iteration and even simply changing the encryption key after each use. However, each of these methods requires some state to be maintained by both the port knock client and server and does not scale very well when multiple users become involved.

An additional port knocking limitation is that it is extremely easy for a malicious third party to bust a knock sequence just by spoofing an additional packet into the port sequence as it is sent over the wire by the client. The attacker would simply set the source address on the packet to be the same as that of the real client and choose the same port number as the last packet sent by the client. This extra packet would break the knock sequence, so the server would not allow the legitimate client any additional access. Although the chances that people would actually do this are relatively small (they still need to be able to monitor packets emanating from the client), the main issue is that such

an attack is so trivially easy to perform. A single packet is all that is required, and the attacker doesn't even need to be inline to the original packet data path.

Finally, knock sequences are easily detectable as port scans by any intrusion detection system

(IDS) that is able to monitor traffic between the client and server. This is particularly true for encrypted knock sequences, which tend to be longer than simple shared sequences. To an IDS, port knocking looks just like a series of probes to various ports from a single IP address within a relatively short period of time, and this fits the definition of a port scan quite nicely.

Single Packet Authorization

The end result of the above discussion is that port knocking provides some real benefits that enhance security, but some serious limitations also need to be addressed. Single Packet Authorization is a relatively new protocol that retains all of the benefits of port knocking, but fixes the limitations discussed above. The first publicly available SPA implementation was released in May 2005 as a piece of software called fwknop (www.cipherdyne.org/fwknop). fwknop was originally created in 2004 as the first port knocking implementation to combine passive OS fingerprinting and port knocking (this made it possible to do things like "accept knock sequences only from Linux-2.4 systems"), but the SPA method is now the most popular (and default) authentication method offered by fwknop. Note that fwknop provides both authentication and authorization services, but a full discussion of the difference between the two is beyond the scope of this article.

Single Packet Authorization mandates a similar architecture to port knocking. Both have client and server components, the server maintains control of a default-drop packet filter, and the server monitors packets passively. However, this is where the architectural similarities between port knocking and SPA diverge.

Single Packet Authorization moves the data transmission to where it belongs—in the application layer. This implies that instead of being able to send only two bytes of data per packet, as in the case of port knocking, SPA is able to send up to the minimum MTU worth of data (1,500 bytes on Ethernet networks) between the client and the server in each packet. This far outstrips the data transmission rate possible with port knocking, and having easy access to this amount of packet data opens up a huge range of possibilities. The remainder of this article discusses Single Packet Authorization as implemented by fwknop.

fwknop defines the following packet format at the application layer:

- > 16 bytes of random data
- > Client username
- > Client timestamp
- > fwknop version
- > Mode (access or command)
- > Access (or command string)
- > MD5 sum

Many of the fields in the SPA packet format have a variable length, but are separated by a : character (fields are base64-encoded, so embedded colons cannot break this syntax). Once the fwknop client builds the packet format above, the entire packet is encrypted using one of two encryption algorithms: the Rijndael symmetric block cipher with a 128-bit shared key or the asymmetric ElGamal algorithm with up to a 2,048-bit public/private key pair generated by GnuPG. By default, the fwknop client sends SPA packets over UDP port 62201,

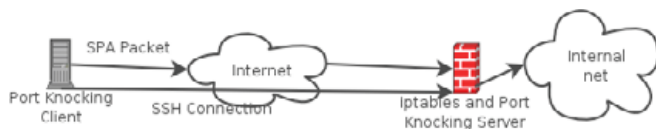


Figure 2. SPA in Action

but this easily can be changed from the command line; see the --Server-port argument. (fwknop offers many configuration options—see Resources for a link to the documentation and man pages.) For a graphical representation of SPA in action, see Figure 2.

So, what are all the fields for? First, the 16 bytes of random data allows one of the highest priority limitations in port knocking to be solved—the replay problem. Every SPA packet is prepended with 16 bytes of random data before being encrypted, and then upon a successful decrypt by the fwknop server, the MD5 sum of the entire packet is cached. The random data allows every SPA packet to be different (even when the same access directive is sent), so the MD5 sum of every packet also has a high probability of being different. If the MD5 sum of any new packet matches the sum of a previous packet, the fwknop server takes no action and writes a warning message to syslog. Hence, any SPA packet that is intercepted by a third party cannot be replayed on the network in an effort to get access through the default-drop packet filter.

The client username and timestamp are placed within the packet by fwknop and the username is used to maintain different authorization levels for remote users by the fwknop server. fwknop can be installed on a multiuser system, and each user can be authorized to connect to different services by a remote fwknop server. The fwknop version field is used to maintain backward compatibility. Fields can be added or deleted in new releases of fwknop, but by using the version number, the fwknop server can remain compatible with the manner in which older clients build SPA packets. The mode field tells the fwknop server whether the client wants to access a service or execute a command (with the specific access control directive or command in the next field). For example, to gain access to TCP port 22, the Access field would contain the string <IP>, tcp/22 where <IP> is whatever IP address the client chose to put in the packet. Finally, the MD5 sum field contains the MD5 sum of the unencrypted packet before the client transmits it. This is used by the server to verify message integrity after decryption.

We already have seen how the increased amount of data that can be transmitted via an SPA packet has solved the replay problem and

Resources

Krzywinski, M. 2003. "Port Knocking: Network Authentication Across Closed Ports". *SysAdmin Magazine* 12: 12–17.

ElGamal Encryption: en.wikipedia.org/wiki/ElGamal_encryption

There is only one other SPA implementation that I am aware of at the time of this writing, available at www.unspecific.com/spa.

Another implementation called Tumbler (tumbler.sourceforge.net) employs a single packet, but it uses a hashed payload instead of an encrypted payload, and this results in a significantly different architecture.

fwknop documentation and man pages:
www.cipherdyne.org/fwknop/docs

the extremely low data transmission rate in port knocking schemes. We have two remaining limitations in port knocking that need to be addressed. First, the single packet nature of the SPA protocol means that a malicious third party cannot break the authentication scheme just by spoofing a packet to the same port over which a monitored SPA packet is sent. Finally, because the SPA protocol requires only a single packet, it does not appear to any intermediate IDS like a port scan. All that any IDS can see is an unintelligible blob of data seemingly spuriously sent to some IP address.

Conclusion

Single Packet Authorization provides similar security benefits to port knocking in terms of protecting services with a packet filter that is configured in a default-drop stance. Anyone scanning for a target service that is protected in this way will be unable to detect such a service is listening, and this makes even the exploitation of zero-day vulnerabilities much more difficult. SPA offers elegant solutions to many limitations in port knocking implementations. These allow SPA to solve the replay problem, achieve a data transmission rate that makes the use of asymmetric encryption possible, thwart simple spoofing attacks and remain under the radar of intrusion detection systems that are monitoring networks for port scans.

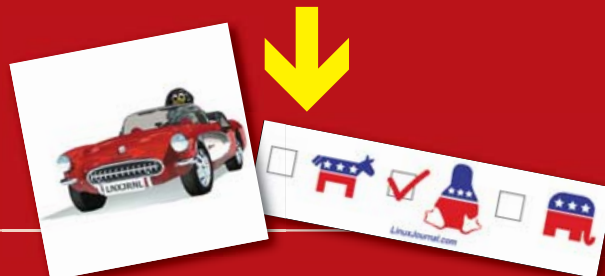
See next month's *LJ* for Part II to this article, which will show exactly how to use SPA. ■

Michael Rash holds a Masters' Degree in applied mathematics with a concentration in computer security from the University of Maryland. Michael is the founder of cipherdyne.org, a Web site dedicated to open-source security software for Linux systems, and he works as Security Architect on the Dragon Intrusion Detection System for Enterasys Networks. He is the author of the upcoming book *Linux Firewalls: Attack Detection and Response*, published by No Starch Press.

FREE STUFF YOU ASK?

OK!

Send us a postage-paid, self-addressed envelope to the below address and we'll return a handful of Linux and Linux Journal stickers to you free of charge.



LINUX JOURNAL
Attn: Sticker promo
PO BOX 980985
Houston, TX 77098



eCryptfs: a Stacked Cryptographic Filesystem

A new cryptographic filesystem in the Linux kernel uses stacking technology.

MIKE HALCROW

The media has been delivering a seemingly endless stream of reports of lost or stolen laptops, backup tapes, hard drives and servers from government and corporate facilities. These devices often contain medical, financial and other sensitive data. When the storage devices fall into the wrong hands, attackers can access the data directly, completely bypassing the access control mechanisms in place in the organization's network. Reports indicate that millions of people already have been affected by such compromises. As a result, customers and citizens are at an increasing risk of identify fraud and loss of privacy.

Although the cryptographic technology to protect data confidentiality has existed for decades, many organizations have failed to integrate this technology into their processes for handling sensitive data. In cases where cryptography is included in that process, it is frequently obtrusive, costly and complicated. Organizations sometimes neglect to establish data encryption policies, and employees often ignore such policies once they are in place.

In cases where employees attempt to utilize cryptography, they often use it ineffectively. For instance, they often select weak keys, and it is easy to save or transfer data inadvertently in unencrypted form through insecure media (such as Web e-mail or a USB Flash drive). Security strategies that depend on individual applications performing their own encryption often fail when the user copies and pastes sensitive information to other applications that do not have cryptographic capability.

Data encryption needs to be made ubiquitous, transparent, flexible, easily deployable, integrated into the data handling process and, of course, secure enough to counter sophisticated attacks. These properties need to be in effect regardless of the particular applications accessing the data. To make encryption services application-agnostic, the operating system kernel itself should provide a system-wide

data encryption service for sensitive information written to secondary storage.

Popular Cryptographic Filesystem Solutions

Several options exist for filesystem encryption under Linux, all with various advantages and disadvantages. Device mapper crypt (dm-crypt) ships with the Linux kernel and provides block device layer encryption. Loop-AES and TrueCrypt, which must be obtained separately from the official Linux kernel, also provide encryption at the block device layer. With block device layer encryption, the user creates the filesystem on the block device, and the encryption layer transparently encrypts the data before writing it to the actual lower block device.

The main advantage of block device layer encryption is that it is simple in concept and implementation. Another advantage of block device layer encryption is that attackers learn nothing about the filesystem unless they have the key; for instance, attackers will not even know the type of filesystem or the directory structure. Sparse files can be securely and efficiently supported in filesystems on encrypted block devices.

Block device encryption can have disadvantages that stem from the lack of integration with the filesystem itself:

- › A fixed region of storage must be pre-allocated for the entire filesystem. Resizing the partition later is often an inconvenient process.
- › It can be difficult to change encryption keys or ciphers.
- › There is no flexibility for the block device encryption mechanism to encrypt different files with different keys or ciphers.
- › Applications such as incremental backup utilities need access to the unencrypted data.
- › All content in the filesystem incurs the overhead of encryption and

decryption, including data that does not require secrecy.

- Files must be re-encrypted with a user-space application before they are transmitted through another medium.

EncFS is a user-space cryptographic filesystem that operates via FUSE. User-space filesystems are easier to implement than kernel-native filesystems, and they have the advantage of being able to utilize user-space libraries easily. This makes it simple to implement feature-rich filesystems with less time and effort on the part of the developer. Unlike block device encryption solutions, EncFS operates as an actual filesystem. EncFS encrypts and decrypts individual files. Disadvantages of user-space filesystems based on FUSE include performance overhead from frequent kernel/user-space context switches and a current lack of support for shared writable memory mappings.

eCryptfs

eCryptfs is a kernel-native stacked cryptographic filesystem for Linux. Stacked filesystems layer on top of existing mounted filesystems that are referred to as lower filesystems. eCryptfs is a stacked filesystem that encrypts and decrypts the files as they are written to or read from the lower filesystem.

Applications in user space make filesystem system calls that go through the kernel Virtual Filesystem (VFS). Both eCryptfs and the lower filesystem (for example, ext3, JFS, NFS and so on) are registered in the kernel VFS. The operations under the eCryptfs mountpoint first go to eCryptfs. eCryptfs retrieves key material from the user session key ring and uses the kernel cryptographic API to perform encryption and decryption of file contents. eCryptfs may make key management requests with the user-space eCryptfs daemon (ecryptfsd). eCryptfs reads and writes encrypted content stored in files in the lower filesystem (Figure 1).

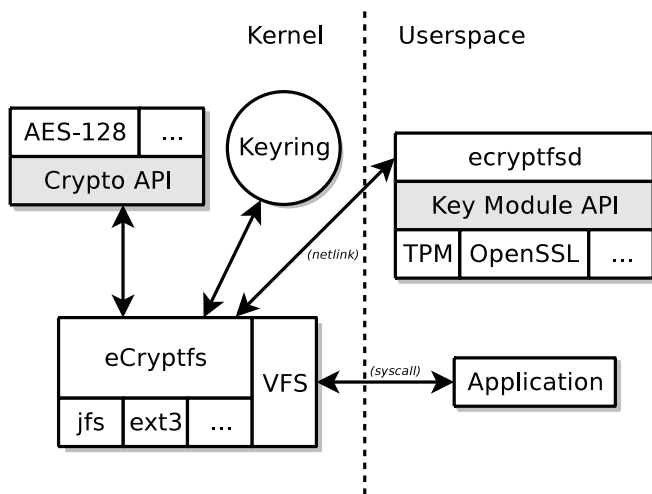


Figure 1. Application file operations go through eCryptfs.

Application file operations go through eCryptfs, which communicates with the kernel crypto API, the kernel key ring and the user-space eCryptfs daemon to perform encryption and decryption. eCryptfs manipulates files in lower filesystems, such as JFS or ext3.

eCryptfs aims to provide the flexibility of a Pretty Good Privacy (PGP) application as a transparent kernel service. For that reason, the OpenPGP (RFC 2440) specification inspires the basic key handling techniques in eCryptfs. This includes the common procedure of using a hierarchy of keys when performing cryptographic operations (Figure 2).

eCryptfs encrypts and decrypts individual data extents in each file

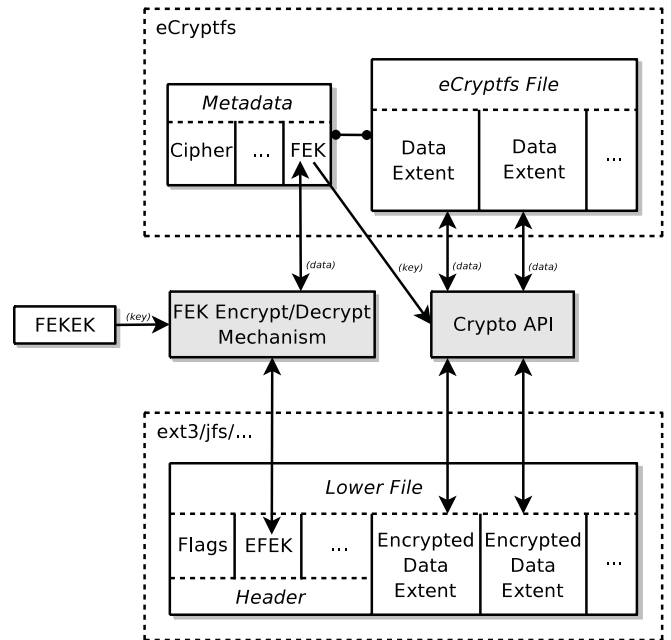


Figure 2. eCryptfs encrypts and decrypts individual data extents.

using a unique randomly generated File Encryption Key (FEK). The FEK is encrypted with the File Encryption Key Encryption Key (FEKEK), and the resulting Encrypted File Encryption Key (EFEK) is stored in the header of each lower file.

The cryptographic metadata is in the header region of the encrypted lower file. Users can transmit the lower file as is to other users, and the recipients can access the decrypted contents of the file through eCryptfs, so long as they have the proper key. This provides a high degree of flexibility in how the files can be handled while maintaining strong security.

Using eCryptfs

eCryptfs requires a kernel component and a user-space component. The kernel component ships in the current mainline Linux kernel. See Listing 1 for the minimum kernel options necessary to enable eCryptfs. By default, eCryptfs uses the AES cipher. eCryptfs can use other ciphers available in the kernel if you build them.

Listing 1. Kernel Options Needed for eCryptfs

```
Code maturity level options --->
[*] Prompt for development and/or
    incomplete code/drivers

Security options --->
<M> Enable access key retention support

Cryptographic options --->
<M> MD5 digest algorithm
<M> AES cipher algorithms

File systems --->
Miscellaneous filesystems --->
<M> eCrypt filesystem layer support (EXPERIMENTAL)
```

Data encryption needs to be made ubiquitous, transparent, flexible, easily deployable, integrated into the data handling process and, of course, secure enough to counter sophisticated attacks.

Newer versions of the Linux kernel contain more feature-rich versions of eCryptfs. For instance, Linux kernel version 2.6.19 is the first official kernel version that contains eCryptfs, and only passphrase mode of operation is available in that kernel. At the time of this writing, the development kernel branch version 2.6.20-mm contains public key support, so that feature may be now available in more recent mainline kernel versions. You can determine the features available in your kernel by loading the `ecryptfs` module and viewing the contents of `fs/ecryptfs/version_str` under your `sysfs` mountpoint.

Popular Linux distributions carry the eCryptfs user-space packages; follow the software package installation procedure for your distribution to install the `ecryptfs-utils` package. If the eCryptfs user-space tools are not yet available from your distribution, you can download, build and install the source tarball. You can obtain the user-space components from the eCryptfs SourceForge site (ecryptfs.sourceforge.net).

If eCryptfs is built as a kernel module, you need to load the module:

```
# modprobe ecryptfs
```

At this point, you can begin using eCryptfs with whatever filesystem you are currently using. To mount eCryptfs, specify the lower directory for the encrypted files and the eCryptfs mountpoint for the decrypted view of the files:

```
# mount -t ecryptfs /secret /secret
```

The first path is the lower directory, and the second path is the eCryptfs mountpoint. Note that the lower directory and the mountpoint have the same path in this example. These paths can be different, but I recommend doing a layover mount in order to help ensure that only eCryptfs has access to the files in the lower filesystem. This command transforms the given path from the lower directory into the eCryptfs mountpoint for the duration of the mount.

When performing a mount, the eCryptfs mount helper first attempts to read in options from the `.ecryptsrc` file in the current user home directory, and then it reads options provided via the command line. The mount helper interactively prompts for any mandatory options that are not specified in the `.ecryptsrc` file or the command line. For instance, you may be asked to choose a passphrase and a cipher.

Once the mount has completed successfully, files written to the `/secret` mountpoint will be encrypted transparently and written to the `/secret` directory in the lower filesystem. Encrypted files that exist in the lower `/secret` directory and that are able to be decrypted with the key specified at the time of the mount will be accessible in their unencrypted form when read from the `/secret` eCryptfs mountpoint.

When you unmount eCryptfs and look in `/secret`, you will see the encrypted lower files. You may first notice that the lower files are larger than the files viewed under the eCryptfs mountpoint. The exact size

of the lower files depends on the page size of your host and on the amount of data written. In general, the minimum lower file size is either 12KB or your host page size plus 4KB, whichever is larger. This helps ensure page alignment between the eCryptfs file and the lower file, which

helps performance. The lower file then grows in 4KB increments as data spills into new 4KB data extents.

The extra space at the front of each lower file contains cryptographic metadata about the file, such as attribute flags and an encrypted file encryption key. Having this information in the file contents makes it convenient to transfer or back up the files while preserving all the information necessary to access the files later. However, the headers can take up a disproportionately large amount of space if there are many small files. Newer releases of eCryptfs can store the data in the extended attribute region instead, reducing the size of the lower encrypted files; refer to the eCryptfs on-line documentation at ecryptfs.sourceforge.net for more information on using this feature.

If your kernel has public key support, you can utilize one of the eCryptfs key modules to manage your key. You can check for support in the version of eCryptfs in your kernel by viewing the contents of `fs/ecryptfs/version_str` under your `sysfs` mountpoint. If there is support, you will see `pubkey` listed as one of the supported features.

Key modules can be selected and parameterized via mount options. If you want to use the OpenSSL key module, you first need to generate a public/private key pair to use in eCryptfs. To generate a key pair, do the following:

- › Run `ecryptfs-manager`.
- › Select menu option 3.
- › Select the `openssl` key module.

You also need to run the eCryptfs daemon in order to manage kernel-user-space communications; the daemon can be started simply by running the executable:

```
# ecryptfsd
```

Note that running the daemon is not necessary if you are using only the passphrase mode of operation. Then, assuming you created your key in `/usb-drive/mykey.pem`, you would mount with the following options:

```
# mount -t ecryptfs \
-o key=openssl:keyfile=/usb-drive/mykey.pem \
/secret /secret
```

Given these options, the eCryptfs mount helper prompts you for a passphrase that protects the private key contained in the key file.

You can mount the same lower directory with many different combinations of keys and ciphers (known as a mount context), and that particular context will apply to any new files created under the mountpoint. For current versions of eCryptfs, files created under any given mount context will be accessible only when the mount is performed with that same context.

Notes on Security

As with any filesystem, you should make regular backups of your data when using eCryptfs. This is done easily and securely by unmounting



One Stop I.T. Solutions Provider

Building Beyond Expectations



ACMA - Your turnkey computing solution provider!

- Design Prototyping
- Production
- Fulfillment
- Branding
- Testing
- High Temp Burn-In



Quad Core Servers

1U High Density Server - Intel® Xeon® Processor based server with only 17" in depth!

2U/5U RAID Servers - Advanced multi-core Intel® Xeon® Processor based server solutions with SATA or SAS drives and up to 18TB of hot swap SATA and 7.2TB hot swap SAS storage capacity.



Supports up to 84 Intel® server blades and 168 energy efficient Dual-Core Intel® Xeon® Processors in one rack!

Expand Your HPC Flexibility

Enjoy great dual-core and quad-core performance plus the flexibility to run 32-bit and 64-bit applications with Dual-Core Intel® Xeon® Processors and Quad-Core Intel® Xeon® Processors in your **ACMA HPC Servers.**



One Stop I.T. Solutions Provider

an ISO-9001 Certified Corporation

www.acma.com

1-800-786-6888 for more info.

If your kernel has public key support, you can utilize one of the eCryptfs key modules to manage your key.

eCryptfs and reading the lower encrypted files.

eCryptfs protects only the confidentiality of data at rest that is outside the control of the trusted host environment. You should use access control mechanisms properly, such as SELinux on the trusted host in order to regulate access to the decrypted files.

eCryptfs will, by default, preserve all of the information necessary to access the decrypted contents of the files in the contents of the lower files themselves. All that is required is the key used to create the files in the first place. You should take measures to protect this key. If applications, such as incremental backup utilities, are configured to read only the lower encrypted files, these utilities do not need to apply any further encryption to the files in order to ensure data confidentiality.

If you are using a passphrase, follow common best practices in selecting and protecting your passphrase (for instance, see www.iusmentis.com/security/passphrasefaq). I recommend using the public key mode of operation instead of passphrase mode whenever possible. When using a public key module, make a backup copy of your key file and store it in a physically secure location. Should you lose your key, nobody will be able to retrieve your data. Do not store unprotected copies of your passphrase or your public key file on the same media as your encrypted data.

You are free to choose among the symmetric encryption ciphers that are available through the Linux kernel cryptographic API. eCryptfs recommends AES-128 as the default cipher. If you have hardware acceleration available on your machine, and if it is supported by the selected cipher in the kernel cryptographic API, eCryptfs encryption and decryption operations will be hardware-accelerated automatically.

You should take measures to ensure that sensitive data is not written to secondary storage in unencrypted form. Applications that write out sensitive temporary data should be configured so that they write only under an eCryptfs mountpoint. You also should use dm-crypt to encrypt the swap space with a random key. The details are beyond the scope of this article, but commands to set it up take the following form:

```
# cryptsetup -c aes-cbc-plain -d /dev/random create \
swap /dev/SWAPDEV
# mkswap /dev/mapper/swap
# swapon -p 1 /dev/mapper/swap
```

SWAPDEV is the swap block device on your machine (refer to your `/etc/fstab` file if you are not sure which device currently is used for swap). You can create simple boot scripts to set up the encrypted swap space automatically, run `ecryptfsd` and perform eCryptfs mounts. Consult your distribution's documentation for more details on writing boot scripts and using `dm-crypt` with a random key to encrypt your swap space.

Note that current releases of eCryptfs encrypt only the file contents. Metadata about the file—for instance, the size, the name, permissions and extended attributes—are all readable by anyone with access to the lower encrypted file. Future work on eCryptfs will include encryption or obfuscation of some of this metadata.

Using block device encryption together with eCryptfs can combine

the security provided by both mechanisms while offering the flexibility of having seamless access to individual encrypted lower files, although this roughly doubles the processing overhead

of encrypting and decrypting the data. If only the contents of the files on secondary storage require confidentiality, eCryptfs by itself is, in most cases, sufficient.

Future Work

eCryptfs was designed to support a host of advanced key management and policy features. The development road map for eCryptfs includes multiple keys per file, different keys and ciphers for different files depending on the application creating the file and the location where the file is being written, integrity enforcement and more extensive interoperability with existing key infrastructures and key management devices. These features will become available as they are implemented in future versions of the Linux kernel.

FIST

The Stony Brook University (SUNY) File Systems and Storage Labs (FSL) (filesystems.org) has developed a stacked filesystem framework called FIST. eCryptfs is derived from Cryptfs, which is one of the example filesystems implemented in FIST. Unionfs is another popular stacked filesystem written by the SUNY FSL.

Conclusion

eCryptfs is a flexible kernel-native solution that cryptographically enforces data confidentiality on secondary storage devices. eCryptfs can be deployed on existing filesystems with minimal effort. The individual encrypted files can be transferred to other hosts running eCryptfs and accessed transparently using the proper key. The eCryptfs key management mechanism is highly extensible. eCryptfs is suitable to use as a strong and convenient data-confidentiality enforcement component to help secure data managed in Linux environments.

Legal Statement

This work represents the view of the author and does not necessarily represent the view of IBM.

IBM is a registered trademark of International Business Machines Corporation in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

TrueCrypt is a trademark of the TrueCrypt Foundation.

Other company, product, and service names may be trademarks or service marks of others. ■

Mike Halcrow (mhalcrow@us.ibm.com) is a Security Software Engineer at the IBM Linux Technology Center and is the lead architect and developer of eCryptfs. He is also pursuing a Master's degree in Computer Science at UT, Austin. In the past, he has maintained the openCryptoki PKCS#11 application, contributed to Common Criteria CAPP/EAL security certification efforts for Linux and authored the BSD Secure Levels Linux Security Module (LSM) that shipped in previous versions of the Linux kernel.

BLADE KILLER 2 SERVERS IN 1U



*84 nodes
in 42U*



Ideal for virtual servers

Features and benefits:

- **Higher Density than Blades** (Yes, really) — Two dual quad-core servers in a single 1U. Up to 84 nodes / 672 processor cores in a 42U rack.
- **Lower Power** — Superior power utilization. Increased power supply efficiency.
- **Redundancy** — High redundancy clusters. Independent servers instead of single points of failure found in blade solutions.
- **Standardization** — Industry standard architecture. No proprietary blade backplane/architecture.
- **Cooling** — Individually cooled 1U chassis vs. several blades in a single large chassis.
- **Complete Customization** — Every node is considerably more customizable than any blade.
- **Hot Swappable** — Swap an entire dual server without interrupting network flow or data array access.
- **Lower Cost** — Substantially lower cost than comparable blade solutions.

	IBM BLADECENTER® H	ABERDEEN STIRLING 122
	9U Blade Solution with 14 Blades	7U Server Solution with 14 Nodes
Description	9U IBM BladeCenter H chassis with 14 IBM BladeCenter HS21 blade servers	7 1U Aberdeen Stirling 122 Servers, each with two server nodes
Processors per blade / node	Two Quad-Core Intel® Xeon® processors E5310 1.60 GHz with 8 MB cache and 1066 MHz FSB	Two Quad-Core Intel® Xeon® processors E5310 1.60 GHz with 8 MB cache and 1066 MHz FSB
Memory per blade / node	2 GB ECC DDR2 667 MHz memory (max 32 GB / 4 DIMM)	2 GB ECC DDR2 667 MHz memory (max 32 GB / 8 DIMM)
Hard drives per blade / node	Two 36 GB 10,000 rpm internal 2.5" SAS drives (max 146 GB without optional SIO blade)	Two 36.7 GB 10,000 rpm hot-swap 3.5" SATA drives (max 1.5 TB)
Maximum blades / processors per 42U rack	56 blades / 112 processors (448 cores) in 4 x 9U chassis (6U empty)	84 server nodes / 168 processors (672 cores) in 42 x 1U chassis (0U empty)
Warranty	3 year on-site limited warranty for parts and labor	5 year limited warranty for parts and labor
Total price for 14 blade / node solution	\$70,161	\$36,495
Price per blade / node	\$5,012	\$2,607



Intel, Intel Logo, Intel Inside, Intel Inside Logo, Pentium, Xeon, and Xeon Inside are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. Other trademarks are of their respective owners. Prices and specifications for IBM products obtained from www.ibm.com on January 30, 2007. For terms and conditions, please see www.aberdeeninc.com/abpoly/abterms.htm. lj018



Multi-Category Security in SELinux in Fedora Core 5

How to set up and use SELinux Multi-Category Security. RUSSELL COKER

The release of Fedora Core 5 added several new features to SELinux, one of which is Multi-Category Security (MCS). The purpose of MCS is to protect data confidentiality, which means it will prevent secret data from being exposed, but it is not designed to prevent the system from being cracked. The SELinux functionality that you may be familiar with from previous Fedora releases (known as the domain-type model) is still used for protecting system integrity. MCS is an extra feature for preventing accidental or deliberate leaks of secret data.

Earlier releases of SELinux used only the domain-type model for access control. In the domain-type model, every process has a domain, and every object that a process may access (files, directories and so on) has a type. The system maintains a set of rules to specify which types each domain may access and what type of access that should be.

Although domain-type can be used to implement all controls of system integrity and data confidentiality (and has been used for this in the past), it makes for a cleaner design if the goals of integrity and confidentiality are separated. MCS is designed to protect data confidentiality, thus allowing the domain-type part of the policy to be focused on protecting system integrity. MCS is based on some of the design features of Multi-Level Security (MLS). MLS is designed for military use and is not suitable for most users, so I don't cover it in this article.

In the past, MLS has had little support, because it's difficult to use and expensive. MCS is a default feature in Fedora Core 5 and above, so it will have good support by application developers and system administrators. It is expected that all applications written for Fedora Core 5 and above will have support for MCS, and that, in most cases, the MCS support also will allow those applications to support MLS. This means organizations that need the MLS features will have a

better choice of applications than they would on a proprietary UNIX system.

MCS adds a sensitivity label (which I refer to as an MCS label for the rest of this article) to each security context. The security context is the complete SELinux label for a process or a resource that a process may access. To access a file, a process must have an MCS label that dominates the MCS label of the file to be accessed. The MCS label is composed of a set of categories. A process may have an MCS label with two levels, referred to as high and low levels; the high level has a super-set of the categories of the low level.

A file might have the MCS label `s0:c0.c10`. The `s0` means nothing in the MCS policy; that field is used by the MLS policy, and the same kernel code is used for both MLS and MCS, so the format can't be changed. The part that matters is `c0.c10`, which means the set of categories from `c0` to `c10` inclusive (the `.` character indicates a range of categories). In Fedora Core 5, there are 256 categories numbered from `c0` to `c255`. In Fedora Core 6, there will be 1,024 categories numbered from `c0` to `c1023`.

A process might have the MCS label `s0-s0:c0.c100`, which means that the low level of the label (or range) is `s0` (no categories), and the high level is `s0:c0.c100`, which means all categories from `c0` to `c100`. Disjoint sets of categories also are permitted. The label `s0:c3,c5` means the categories `c3` and `c5` are in the label. The MCS range `+s0:c3,c5-s0:c0.c10,c20.c30` means that the low level has categories `c3` and `c5`, while the high level contains categories `c0` to `c10` inclusive and `c20` to `c30` inclusive.

Categories may be named, and it is expected that most users will name all the categories that are used. In Fedora Core 5, you have to edit the file `/etc/selinux/targeted/setrans.conf` to change the human-readable names for the MCS labels. Below is a section of the default `setrans.conf` file:

```
# s0:c0=CompanyConfidential
# s0:c1=PatientRecord
# s0:c2=Unclassified
# s0:c3=TopSecret
# s0:c1.c3=CompanyConfidentialRedHat
s0=
s0-s0:c0.c255=SystemLow-SystemHigh
s0:c0.c255=SystemHigh
```

And, the following is an example of how to use semanage to change the human-readable translations of MCS labels in Fedora Core 6:

```
# semanage translation -a -T ProjectA s0:c0
# semanage translation -l
Level          Translation
s0
s0-s0:c0.c1023 SystemLow-SystemHigh
s0:c0          ProjectA
s0:c0.c1023   SystemHigh
```

Figure 1 shows the access that processes are granted to files for all combinations of the categories HR and Financial.

MCS is designed with ease of use as a major concern. At the current time, its design is to control only file access. It also controls ptrace (the system interface used for strace, ltrace and debuggers) to prevent an unprivileged process from using a debugger to capture secret data from a more-privileged process.

The fact that MCS controls only file access does permit information leaks through filenames, and cooperating processes may use TCP, UDP, UNIX domain sockets or named pipes to transfer data. It was designed this way intentionally, because restricting all forms of inter-process communication will break many programs and make the entire system more difficult to use. The MLS policy (which is available in Fedora Core 5 and above but not enabled by default) restricts such communication methods, which is one of the reasons why it is regarded as being too difficult for most people to use.

When designing MCS, we decided not to try to prevent two cooperating users from inappropriately sharing data. We also decided not to prevent a user from reading a file with secret data and then writing that data to a file with a less secret label. Again, the MLS policy restricts these operations, but it is too difficult for most people to use.

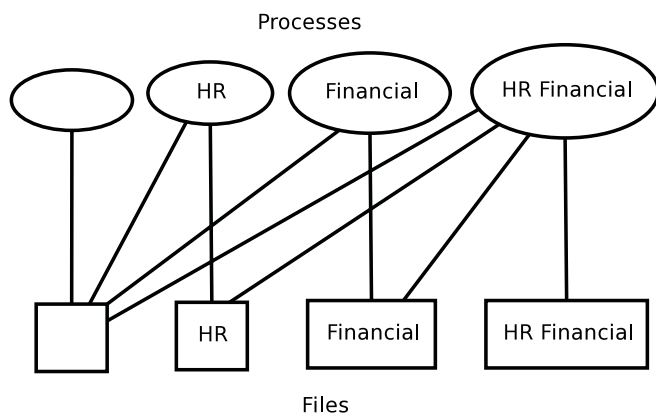


Figure 1. Sample Categorical Security Grouping

MCS and MLS are designed to protect confidentiality of data; they rely on the domain-type model to protect system integrity. FC5 comes with three policies. The default is targeted, which offers much the same integrity protections as it did in FC4 but has the addition of MCS to protect data confidentiality. The next option is the strict policy, which, again, is much the same as it was in FC4 but with the addition of MCS. Finally, there is a new policy in FC5 called mls; as the name suggests, this includes the MLS system to protect data confidentiality. The domain-type part of the mls policy is based on the strict policy (although not all the daemons from the strict policy are supported—only those from the evaluation list for LSPP certification).

It is possible to compile an SELinux policy without support for MLS or MCS features, but so far, no one has chosen to do so. It would save very little memory and would be worth considering only for the smallest embedded machines. It also is possible to use MLS instead of MCS with the targeted policy, but no one has done so, because it would not provide much benefit. System integrity is a precondition for data confidentiality. Therefore, there is no benefit in combining a strong system of confidentiality protection, such as MLS, with anything less than the best protection of system integrity. The strict policy confines all processes and significantly limits most of them. This is the level of integrity protection that is needed to take full advantage of MLS.

In MCS, a process has a range. The high level of the range determines the access granted to files, and the low level determines the default level of files that are created (ranges apply only to processes in MCS).

For an MCS level to dominate another, it must have a set of categories that is a super-set of the MCS level that is being dominated. It is possible to have two sensitivity levels for which neither dominates the other (for example, disjoint sets of categories). This is referred to as incomparable levels, and both read and write access will be denied.

To use MCS, first you need to assign sensitivity levels to users. In previous releases of SELinux, it was necessary to edit the policy source and recompile the policy to set the security context that is assigned to users when they log in, which was inconvenient and error-prone. One of the new SELinux features in FC5 is the semanage policy management tool. This supports changing the security contexts of users (and adding and removing users) without compiling the policy.

The default configuration of Fedora has the targeted policy that runs all user login sessions in the unconfined_t domain (no access restrictions in the domain-type model), so MCS provides the only SELinux access controls for users. However, a default install of Fedora Core 5 needs to have updates applied before MCS will work correctly. The development of MCS was not complete until after the release of Fedora Core 5.

The first thing you must do when configuring an SELinux system to use MCS is create SELinux identities and login records to map them to UNIX accounts.

Listing 1 is an example of using semanage to add the SELinux identity rjc with a low level of s0:c1 (which means every file the user creates will have category c1 by default) and a high level of SystemHigh, which maps to s0:c0.c1023 (the range of all categories from c0 to c1023 inclusive—the highest level of MCS access) in Fedora Core 6 and to s0:c0.c255 in Fedora Core 5. The -L parameter specifies the default level. When using MCS, you always should make the default level the low end of the range to avoid confusion. The -L option is separate from the range

MCS is designed to protect data confidentiality, thus allowing the domain-type part of the policy to be focused on protecting system integrity.

so this won't be a problem. After creating a file, it is possible to change the label to a different level with the `chcon -l` command. Below is an example of how to use it:

Listing 1. Example of Using semanage

```
# semanage user -a -P user -R user_r -L s0:c1 -r s0:c1-SystemHigh rjc
# semanage user -l
```

SELinux User	Labeling Prefix	MLS/MCS Level	MLS/MCS Range	SELinux Roles
rjc	user	s0:c1	s0:c1-SystemHigh	user_r
root	user	s0	SystemLow-SystemHigh	system_r
sysadm_r				
+user_r				
system_u	user	s0	SystemLow-SystemHigh	system_r
user_u	user	s0	SystemLow-SystemHigh	system_r
sysadm_r				
+user_r				

```
$ touch foo
$ ls -lZ foo
-rw-r--r-- rjc rjc rjc:object_r:tmp_t foo
$ chcon -l s0:c0 foo
$ ls -lZ foo
-rw-r--r-- rjc rjc rjc:object_r:tmp_t:ProjectA foo
```

Note that the level `s0:c0` was translated to `ProjectA`; that is the translation I created previously.

It is possible to run a process with a different range. The following is an example of the use of the `id -Z` command to display the SELinux context (including the MCS range at the end) as well as the use of the `runcon -l` command to run an instance of `bash` in a different range:

```
$ id -Z
rjc:system_r:unconfined_t:SystemLow-SystemHigh
$ runcon -l s0-s0:c10.c20 bash
$ id -Z
rjc:system_r:unconfined_t:s0-s0:c10.c20
$ runcon -l s0-s0:c9.c20 bash
execvp: Permission denied
```

to support the needs of the MLS policy.

Fedora Core 5 uses the low end of the range for a process to specify the default context of files. The range parameter is specified by the `-r` switch. When using Fedora Core 5, often the only significant part of the range is the high end, which specifies the access. You should use the parameters `-P user -R system_r` when creating a user with the targeted policy (which is the default policy for a Fedora Core 5 install). The strict policy is much like the targeted policy in terms of MCS. Most of this article applies to the strict policy, although the `-P` and `-R` options to the `semanage` command will need different parameters.

After adding an identity, you must add a login entry to assign a UNIX account to the identity. The login configuration also allows you to specify the MCS range, because you may have many UNIX accounts with the same SELinux identity that have different MCS ranges assigned to them when they log in. You must use the `-s` parameter to specify the name of an SELinux identity that already exists. If you do not use the `-r` option to specify the range, it defaults to using no categories for the login entry in question (which may not be valid, depending on the low level of the range for the identity).

Below is an example of adding a login entry:

```
# semanage login -a -s rjc -r s0:c1-SystemHigh rjc
# semanage login -l
```

Login Name	SELinux User	MLS/MCS Range
__default__	user_u	s0
rjc	rjc	s0:c1-SystemHigh
root	root	SystemLow-SystemHigh

Note that the range for a login entry must be a subset of the range for an SELinux user identity. This means the low end of the login range must not be lower than the low end of the user identity range, and the high end of the login range must not be higher than the high end of the user identity range. In most cases, you will create a login entry with the same range as the user identity,

Summary

MLS was implemented in a flexible manner via the policy language. This allowed us to develop the MCS policy afterward using the same language features and also permits the development of other category- and level-based confidentiality controls without changing kernel code. One example of this is my new development, Mandatory MCS (MMCS).

A Mandatory Access Control (MAC) system is a system where the access control is determined by the system administrator and enforced by the operating system. Users are not permitted to override this access control by granting excess access to their own data files. In UNIX permissions, it is possible to create a mode 777 file in the `/tmp` directory that grants full access to all users. With MMCS, I wanted to prevent such access being granted. In the MMCS policy, it is not permitted to write to a file with a level below the low level of the process. This means that by setting the low levels for a user, the administrator can determine the minimum access needed to read files created by that user.

MCS and MLS policies have several significant differences. In MLS, the access is based on the low level of the range (the effective clearance) with the high level of the range used mostly to determine the access via the `newrole` program. In MCS, the access for both reading and writing is based on the high level of the range with the low level used only for restricting write access. Another difference is that MCS is designed to protect only the contents of files, while MLS restricts all methods of data transfer. Another major difference is that in MCS, a process may launch a child with a different range with minimal restrictions. ■

Russell Coker has worked on Security-Enhanced Linux (SELinux) since 2001. He is an independent consultant specializing in SELinux and ISP administration.

Polywell's Ultimate Linux Systems

More Choices, Better Service, Great Value



Tower Systems

4-way Opteron, OpenGL QFX PolyStation 2055A-2210

- 2xAMD Opteron™ Dual-Core 2210 Processors
- 2GB Dual Channel 667MHz DDR2 ECC
- NVIDIA Quadro™ FX 1300 OpenGL Graphics
- 2x250GB SATA Drives, On-board RAID-5
- Sony 16X DVD-RW, Logitech K/B, Mouse
- 2x Gigabit LAN, 2 x PCIe 16x Slots
- Linux / BSD / Windows OS Supported

\$1999 (custom config. available)

4-way Opteron, SAS Drives PolyStation 2055SS-2216

- 2xAMD Opteron™ Dual-Core 2216 Processors
- 8GB Dual Channel 667MHz DDR2 ECC
- NVIDIA Quadro™ FX 3500 OpenGL Graphics
- 2x15K RPM 73GB SAS Drives, SAS Controller
- Sony 16X DVD-RW, Logitech K/B, Mouse
- 2x Gigabit LAN, 2 x PCIe 16x Slots
- Linux / BSD / Windows OS Supported

\$4679 (custom config. available)

4-way Opteron, 32GB RAM Tower Server 2500M-2220

- 2xAMD Opteron™ Dual-Core 2220 Processors
- 32GB Dual Channel 667MHz DDR2 ECC
- 11 1-Bay Tower with 2x700W Power Supply
- 3TB 6x500GB SATA Drives, RAID-5
- Sony 16X DVD-RW, Logitech K/B, Mouse
- 2x Gigabit LAN, 2 x PCIe 16x Slots
- Linux / BSD / Windows OS Supported

\$9499 (custom config. available)

Desktop Systems

Low-Cost Linux Appliance PC Poly 485Ax-2800SP

- AMD Sempron™ 2800+ 64-bit Processor
- 256MB 667MHz DDR2 Memory
- ATI X300 PCIe Graphics
- 80GB SATA Drive, SATA-RAID Controller
- 16X DVD-ROM Drive, Logitech K/B, Mouse
- 100Mbit LAN, 2 PCI + 1 PCIe 16x Slots
- Linux / BSD / Windows OS Supported

\$299 (for volume purchase only)

2TB Storage Appliance MiniBox 430AM2-3200SP

- AMD Athlon™ 64 3200+ 64-bit Processor
- 512MB Dual-Channel 667MHz DDR2 Memory
- NVIDIA GeForce™ 6150 Graphics
- 2TB 4x500GB SATA Drives, RAID-5
- Optional DVD or RW Drive, K/B, Mouse
- Gigabit LAN, 2 PCI + 1 PCIe 16x Slots
- Linux / BSD / Windows OS Supported

\$1239 (custom config. available)

High-End PC, 4G DDR2 Poly 430AM2-FX62

- AMD Athlon™ 64 Dual-Core FX-62 Processor
- 4GB Dual Channel 667MHz DDR2
- NVIDIA GeForce™ 7900GS Graphics
- 10K RPM 74GB + 320GB SATA Drives
- 16X DVD-RW, 20-in-1 Reader, 1394a
- Gigabit LAN, 2 PCI + 1 PCIe 16x Slots
- Linux / BSD / Windows OS Supported

\$2250 (custom config. available)



Rack Servers

Low-Cost ISP RackServer 1102EV-17 485Ax-3000SP

- AMD Sempron™ 3000+ 64-bit Processor
- 512MB 667MHz DDR2 Memory
- ATI X1000 PCIe Graphics
- 80GB SATA Drive, 4xSATA-RAID Controller
- 1U 17" Short Rack, 1 Slim-CD, 2 x 3.5" Bays
- 100Mbit LAN, 1 Optional PCIe or PCI RISER
- Linux / BSD / Windows OS Supported

\$399 (for volume purchase only)

2GB RAM, 500GB ISP Server 1102EV-17 430AM2-4200

- AMD Athlon™ 64 X2 Dual-Core 4200+ Processor
- 2GB Dual-Channel 667MHz DDR2 Memory
- NVIDIA GeForce™ 6150 Graphics
- 500GB 2x250GB SATA Drives, RAID-5
- 1U 17" Short Rack, 1 Slim-CD, 2 x 3.5" Bays
- Gigabit LAN, 1 Optional PCIe or PCI RISER
- Linux / BSD / Windows OS Supported

\$950 (custom config. available)

2GB ECC, 1TB ISP Server 1102EV-17 1000SL-1210

- AMD Opteron™ Dual-Core 1210 Processor
- 4GB Dual Channel 667MHz DDR2 ECC
- Integrated Graphics, 2 x Gigabit LAN
- 1TB 2x500GB SATA Drives, RAID-5
- 1U 17" Short Rack, Slim-CD+2x3.5" Bay
- Optional PCIe or PCI RISER Slot
- Linux / BSD / Windows OS Supported

\$1779 (custom config. available)

High-Density Multi-Processor Servers



16-way Opteron, 128GB RAM 5124AIS 8850S-8212

- 8 x AMD Opteron™ Dual-Core 8212 Processors
- 128GB Dual Channel 667MHz DDR2 ECC
- 12TB 24x500GB SATA Drives, RAID-5
- 4 x Gigabit Ethernet, 4 PCI-X, 2 PCIe Slots
- 5U 24-Bay RackCase, Redundant Power Supply
- Linux / BSD / Windows OS Supported

16-way 8212, 128G RAM, 24x500G **\$99999**
8-way 8212, 32GB RAM, 8x500GB **\$15999**

18TB 4U Storage, 32GB RAM 4024AIS 2500M-2210

- 2 x AMD Opteron™ Dual-Core 2210 Processors
- 32GB Dual Channel 667MHz DDR2 ECC
- 18TB 24x750GB SATA Drives, RAID-5
- 2 x Gigabit Ethernet, 4 PCI-X, 2 PCIe Slots
- 4U 24-Bay RackCase, Redundant Power Supply
- Linux / BSD / Windows OS Supported

18TB 24x750G, 32G, 2x2210 **\$19999**
11TB 22x500G, 2GB, 1x2210 **\$9999**

6TB 2U Storage, 16GB RAM 2012SAS 2500M-2210

- 2 x AMD Opteron™ Dual-Core 2210 Processors
- 16GB Dual Channel 667MHz DDR2 ECC
- 6TB 12x500GB SATA Drives, RAID-5 Controller
- 2 x Gigabit Ethernet, 4 PCI-X, 2 PCIe Slots
- 2U 12-Bay RackCase, 3 RISER or 7 Lowprofile
- Linux / BSD / Windows OS Supported

6TB 12x500G, 16G, 2x2210 **\$7999**
1.5TB 6x500G, 2GB, 1x2210 **\$3399**

2TB 1U Server, 32GB RAM 1104SC 2500M-2210

- 2xAMD Opteron™ Dual-Core 2210 Processors
- 32GB Dual Channel 667MHz DDR2 ECC
- 6TB 4x500GB SATA Drives, RAID-5
- 2 x Gigabit Ethernet, 1 PCI-X or PCIe RISER
- 1U RackCase, 1 Slim CD, 4 Swap HD Bays
- Linux / BSD / Windows OS Supported

2TB 4x500G, 32G, 2x2210 **\$7888**
1TB 4x250G, 2GB, 2x2210 **\$1999**

AMD Dual-Core technology enables one platform to meet the needs of multi-tasking and multi-threaded environments; provides platform longevity

Polywell OEM Services, Your Virtual Manufacturer

Prototype Development with Linux/FreeBSD Support
Small Scale to Mass Production Manufacturing
Fulfillment, Shipping and RMA Repairs



- 20 Years of Customer Satisfaction
- 5-Year Warranty, Industry's Longest
- First Class Customer Service

888.765.9686

www.Polywell.com/us/LJ



Polywell Computers, Inc 1461 San Mateo Ave. South San Francisco, CA 94080 650.583.7222 Fax: 650.583.1974

Opteron, Sempron and ATHLON are trademarks of Advanced Micro Devices, Inc., Quadro, nForce and Nvidia are trademarks of NVIDIA Corporation. All other brands, names are trademarks of their respective companies.



PacketFence

How to set up and use the powerful open-source network access control solution.

LUDOVIC MARCOTTE AND DOMINIK GEHL

With the ever-increasing number of attacks on networks—either by people accessing them anonymously or generating illegal activities from them, having great security tools is essential. Although a good firewall and tools, such as Snort and Nessus, can increase security, network administrators are looking for solutions that complete those security tools by responding automatically to a violation of network usage policy. Such tools are called network access control (NAC) solutions. Many of those tools exist—especially proprietary ones from big vendors, such as Cisco—but an open-source solution, PacketFence, deserves attention.

PacketFence is a free and open-source solution that provides network access control functionalities, including the following standard features:

- › Registration of network components (desktops, laptops, printers and so on) and, optionally, acceptance of a network usage policy upon registration before gaining complete network access.
- › Detection of network usage policy violations based on passive and active network scans on all connected nodes.
- › Isolation of offending nodes.
- › Notification (e-mail, pop-ups and so on) based on a network usage policy violation.
- › Remediation so that network components can regain their network access after a violation.

PacketFence is written in Perl and makes use of common open-source components, such as MySQL, Apache, Snort and Nessus. It

does not require a user agent to be installed on computers accessing the network. Its deployment is non-intrusive, and every interaction with users goes through a captive portal that can be accessed by every Web browser.

PacketFence currently supports ARP, DHCP/DNS and VLAN isolation techniques. Choosing the right isolation method depends on the size of your network and the networking equipment you possess. In this article, we cover ARP-based isolation, which works on any kind of networking equipment.

ARP-Based Isolation

ARP-based isolation works by poisoning the ARP cache of any equipment connected to the network. As you know, ARP is a protocol used to map IP addresses to MAC addresses. Fundamentally, four basic types of messages exist in Ethernet ARP that are interesting for PacketFence:

1. ARP request: request for the destination MAC.
2. ARP reply: reply containing the MAC.
3. RARP request: request IP from MAC.
4. RARP reply: reply containing the IP.

The problem with ARP is that when a client issues an ARP request, it simply trusts the reply that comes in and stores it into its cache. Poisoning the ARP cache is as simple as sending ARP replies to the client, even if it hasn't asked for one. The operating system likely will update the cache upon reception of such packets, or it'll use the poisoned data we send when it decides to update the cache.

Installation and Configuration

PacketFence has been developed on Red Hat Enterprise Linux 4, CentOS 4 and Fedora Core. Several people have succeeded in running it on different distributions, but to ease your first installation, it might be better to stick with one of the officially supported distributions. Because PacketFence is a NAC solution and installing it will act on your current LAN, make sure to coordinate your tests with your network administrator.

Preparation

PacketFence uses a MySQL database to store the information about the nodes connected to the network, whom they belong to and whether there are any violations of the specified network policy. So, if you don't already have a dedicated MySQL server you want to use for this purpose, install MySQL server by running `up2date -i mysql-server`.

As mentioned previously, PacketFence can use Snort and Nessus, and we describe below how you can integrate both tools with PacketFence.

Snort is an open-source network intrusion detection system that

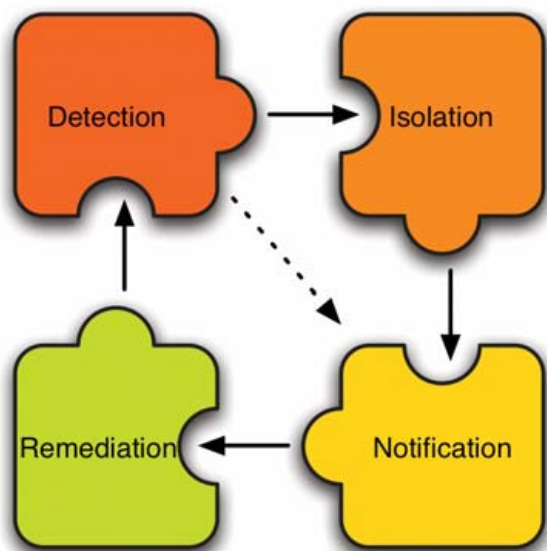


Figure 1. The Relations between PacketFence Standard Features

uses signatures to analyze the network traffic. Once a given packet matches a signature, Snort can generate an alert. Signatures not only exist for many computer viruses and spyware, but also for network traffic, such as BitTorrent, ICQ, Skype or even Hotmail access. They are available from Sourcefire, Inc., through the Snort Web site, and through Bleeding Edge Threats (see Resources). PacketFence also ships with an Oinkmaster configuration to obtain and cut down the ruleset automatically to only what is required by PacketFence. Because PacketFence support for Snort 2.6 is still under development, download Snort 2.4.5 from www.snort.org/dl/binaries/linux/old, and then install the RPM by executing:

```
rpm -ivh snort-2.4.5-1.RHEL4.i386.rpm
```

Nessus, on the other hand, is an active vulnerability scanner—meaning that it generates connections to the hosts you want to test for vulnerabilities. You have to register with Tenable Network Security, the owner of Nessus, in order to receive the available plugins. Install Nessus by downloading version 2.2.9 for Linux and executing:

```
sh nessus-installer-2.2.9.sh
```

Nessus 3 is not yet well supported, and due to the licensing issues surrounding it, stick with 2.2.9.

Once the installation finishes, start Nessus with:

```
/opt/nessus/sbin/nessusd -D
```

and test the connection with NessusClient, which is available as a separate download.

Installation

Download the PacketFence RPM from the SourceForge repository, and install it using:

```
rpm -ivh packetfence-1.6.2-1.i386.rpm
```

In `/usr/local/pf`, you will find two Perl scripts that will help you with the necessary configuration steps: `installer.pl` and `configurator.pl`. Change your current directory to `/usr/local/pf`, and execute `installer.pl`. The script, among other things, sets up the PacketFence database, installs all the necessary Perl modules (which are quite a few) and creates a user account for the Web GUI.

Configuration Steps

Now, the real configuration work starts. First, execute `configurator.pl`, and you'll be offered several choices. Choose the template configuration based on the testing mode. You'll be asked to supply several network parameters (DHCP servers, DNS servers and so on), and a basic configuration file, `/usr/local/pf/conf/pf.conf`, will be created. This configuration file contains only the differences you apply to the default configuration parameters saved in `/usr/local/pf/conf/pf.conf.defaults`. Have a look at `conf/pf.conf.defaults` to get an idea of the available options. To help you see what's going on inside PacketFence, add the following lines to

Expert Included.

Shane's customers are always pushing the limits of technology. That's why he is a fan of the Rackform iServ R2020, an innovative, 1U, two-compute-node system designed to increase computing density while reducing cost, energy, and space requirements. With support for two Quad-Core Intel® Xeon® Processors 5300 Series per compute node, the iServ R2020 combines Intel's proven reliability with industry-leading 16-core-per-1U density. Additionally, 8 Fully Buffered DIMM sockets, 2 hot-swap SATA hard drives, and a PCI-Express slot in each compute node convince Shane that the iServ R2020 provides the density, flexibility, and cost effectiveness needed to tackle even the most demanding computing challenges.

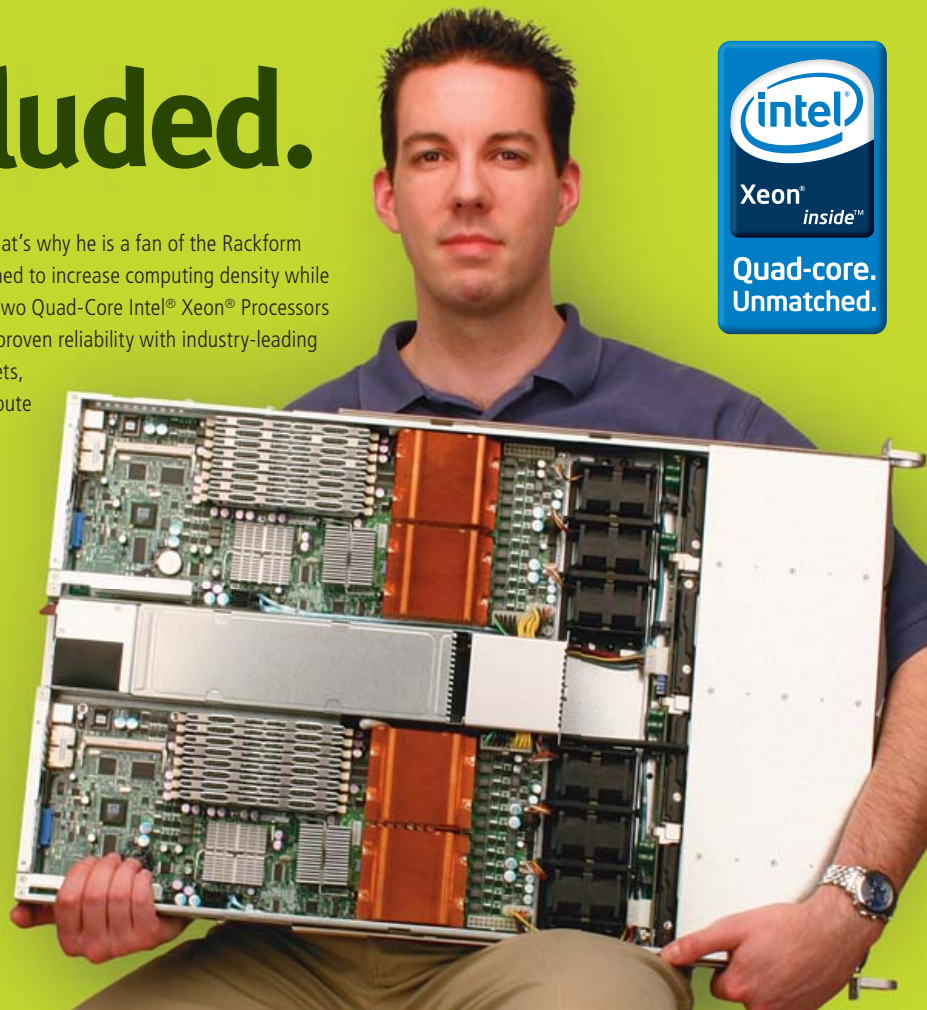
When you partner with Silicon Mechanics, you get more than a powerful Intel Solution — you get an expert like Shane.



visit us at www.siliconmechanics.com
or call us toll free at 866-352-1173

Silicon Mechanics and the Silicon Mechanics logo are registered trademarks of Silicon Mechanics, Inc.

Intel, the Intel logo, Xeon, and Xeon Inside are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.



PacketFence is written in Perl and makes use of common open-source components, such as MySQL, Apache, Snort and Nessus.

`/usr/local/pf/conf/pf.conf` to increase the logging level:

```
[logging]
verbosity=8
```

Basic Usage

Start PacketFence with `service packetfence start`. Have a look at `/var/log/messages`, and you should see that PacketFence started creating an inventory of all nodes on your network, as in the following example:

```
Jan  3 16:05:28 pf pf: update_hashes(5): UPDATE
New node 00:07:e9:05:4c:f2 (192.168.0.1)
Jan  3 16:05:28 pf pf: update_hashes(5): UPDATE
New node 00:90:27:6a:71:ea (192.168.0.2)
```

`/usr/local/pf/bin/pfcmd` is the PacketFence command-line interface. Executing it without any further parameters shows a help screen with the available options. In order to show all nodes in the database, execute:

```
#!/usr/local/pf/bin/pfcmd node view
mac|pid|detect_date|regdate|lastskip|status|
user_agent|computername|last_arp|last_dhcp|switch|
port|vlan|dhcp_fingerprint
00:07:e9:05:4c:f2|1|2007-01-03 16:05:28|||unreg|||
2007-01-03 16:10:11|||
00:90:27:6a:71:ea|1|2007-01-03 16:05:28|||unreg|||
2007-01-03 16:10:12|||
```

Manually registering a node can be done with:

```
/usr/local/pf/bin/pfcmd
➔node edit 00:07:e9:05:4c:f2 status="reg",pid=1
```

You also can use `pfcmd` to access the documentation for every configuration parameter. To see the documentation for the `logo` parameter from the `[general]` section:

```
# /usr/local/pf/bin/pfcmd config help general.logo
GENERAL.LOGO
Default: /common/packetfence.gif
Logo displayed on Web pages.
```

Have a look at the available reports using:

```
/usr/local/pf/bin/pfcmd help report
```

The `os` and `osclass` reports use PacketFence's DHCP fingerprinting feature, which tries to determine the operating system of every DHCP request (including the ones made by printers, VoIP phones, switches and so on).

Running:

```
/usr/local/pf/bin/pfcmd report os
```

shows the number and percentage of nodes on your network for every detected operating system. Note that the DHCP fingerprinting feature easily can be used

to disallow access to your network by computers running specific operating systems.

PacketFence also features an administrative Web GUI, which, by default, is available on the secured port 1443. Direct your browser to `https://<pf-host>:1443/`. Once you enter the login/password you defined during the installation, you can start monitoring and configuring PacketFence through the GUI.

When you start enforcing the registration of nodes with PacketFence, all nodes on the network have to be registered before they can gain complete network access. This registration requirement applies to all gear with network access, including wireless access points and printers. So, before actually activating this option in the configuration file, it is wise to preregister those types of devices manually.

For computers with Web browsers, on the other hand, the registration can be done by the user through the PacketFence captive portal. The portal can verify login/password information through a `htaccess` file, Radius or LDAP, which we use in our example. In order to do this, you need to adapt the provided template `/usr/local/pf/conf/templates/ldap.conf` to fit your LDAP structure.

Because all your users will be redirected to the registration screen, it also is wise at this point to change the default PacketFence logo, which is shown on the Web pages, to your own company logo. This can be done by adding `logo=/common/mylogo.gif` to the `[default]` section in `/usr/local/pf/conf/pf.conf` and copying the file `mylogo.gif` into the directory `/usr/local/pf/html/common/`.

To activate the registration, incorporate the following parameters into `/usr/local/pf/conf/pf.conf`:

```
[trapping]
testing=disabled
detection=disabled
```

```
[registration]
aup=disabled
auth=ldap
```

Now, restart PacketFence with `service packetfence restart`. You should see in `/var/log/messages` that PacketFence is trapping unregistered nodes by ARP-spoofing your network's gateway. From the client side, opening a Web browser and accessing any outside Web site should lead to a redirection to the PacketFence captive portal, which allows you to register the computer. You also can determine whether a client has been ARP-spoofed by executing `arp -n -a` (under Linux) on the client and checking which MAC is saved in the ARP cache for your network's gateway.

Incorporating Snort and Nessus

You can take it a step further by adding Snort alerts to your PacketFence installation. Let's assume that using BitTorrent clients is prohibited in your environment and you want to configure PacketFence to enforce this policy. Edit `/usr/local/pf/conf/violations.conf` so that the section containing BitTorrent reads as follows:

JavaOne™

Sun's 2007 Worldwide Java Developer Conference™

May 8-11, 2007

The Moscone Center
San Francisco, CA

JavaOne™ Pavilion: May 8-10, 2007



IT'S AN EXCITING TIME FOR THE ENTIRE JAVA™ TECHNOLOGY COMMUNITY.

With the evolution of the Java platform, new opportunities are emerging for developers, thought leaders, and entrepreneurs. That's why for 2007 the Conference is featuring an expanded program that embraces technologies outside the core Java platform while keeping Java™ technology at the center. You'll get the same in-depth content we have always focused on, plus more topics and issues relevant to today's evolving market.

LEARN MORE ABOUT:*

- > Java Technology
- > Scripting
- > Open Source and Community Development
- > Integration and Services-Oriented Development
- > Web 2.0
- > Compatibility and Interoperability
- > Business Management
- > And More



ORACLE®



NAVTEQ

INTERSYSTEMS

PARASOFT
The make software work.

PLATINUM COSPONSORS

GOLD COSPONSORS

SILVER COSPONSORS

SAVE \$200**
REGISTER BY
APRIL 4, 2007

Please use priority code: J7PA1LJ

*Content subject to change.
**Offer not available on site.

Register Today @ java.sun.com/javaone



```
[2000334]
desc=P2P (BitTorrent)
priority=8
url=/content/index.php?template=p2p
disable=N
max_enable=1
actions=trap,email,log
trigger=Detect::2000334,Detect::2000357,
➔Detect::2000369
```

This tells PacketFence that:

- The BitTorrent violation can be generated by the Snort alerts 2000334, 2000357 and 2000369 (the trigger parameter).
- The system has to act upon this violation by isolating the user, sending an e-mail alert to the administrator and logging the violation to `/var/log/messages` (the actions parameter).
- The user is allowed to re-enable network access once (the `max_enable` parameter).

Finally, you need to activate Snort from inside PacketFence by incorporating the following into `conf/pf.conf`:

```
[trapping]
detection=enabled
```

```
[interface eth0]
type=internal, managed,monitor
```

and restarting the PacketFence service. Note that if you are using switches, you have to redirect a copy of your network traffic to `eth0` (the PacketFence monitor—that is, the interface Snort listens to for packets).

Generating a violation in PacketFence is now as simple as launching your favorite BitTorrent client, such as Azureus. Do so, and open a torrent file to start a download. Once a couple of packets are exchanged on the network, Snort should catch some and match them with the 2000334, 2000357 or 2000369 rules. Those rules, which come from the Bleeding Edge Threats rulesets, correspond to BitTorrent peer sync, traffic and announce, respectively. Once Snort logs such unusual activity, PacketFence reacts by creating a violation.

As an administrator, you can see the list of violations with:

```
/usr/local/pf/bin/pfcmd violation view all
```

As a user, on your computer, launch your favorite Web browser and try to open any outside Web site. You'll be redirected to the

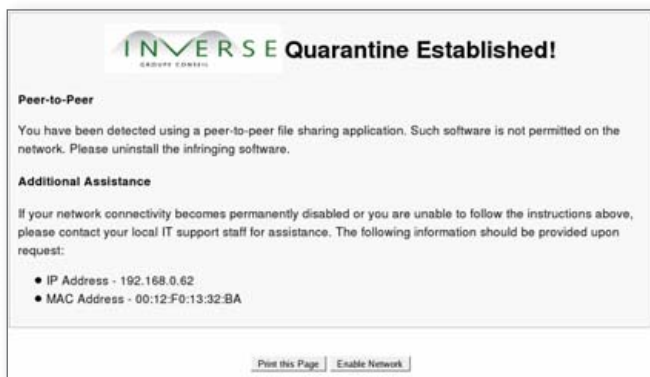


Figure 2. PacketFence Peer-to-Peer Template

PacketFence system automatically, which will display the peer-to-peer template, as shown in Figure 2.

Stopping the BitTorrent client lets you regain network access. Now, try to use BitTorrent a second time. As before, an alert is generated, and your browser is redirected to the PacketFence portal. But this time, however, you won't get another chance to re-enable your access. In real life, offending users would have to call their network administrator to re-enable network access.

To activate the Nessus scanning of hosts trying to register with PacketFence, add the following section to `/usr/local/pf/conf/pf.conf`:

```
[scan]
ssl=enabled
pass=<nessus_passwd>
user=<nessus_user>
port=1241
host=127.0.0.1
registration=enabled
```

and restart the PacketFence service. Trying to add a computer running an unpatched version of Microsoft Windows now generates an immediate violation. You also can use the administrative Web GUI (the Scan tab) to define that complete Nessus scans should be executed every night.

Conclusion

Deploying PacketFence in your network with ARP-based isolation is simple. Although this isolation technique is easy to deploy, it doesn't necessarily scale well and could be bypassed by wise users with static ARP-cache entries.

Other isolation techniques exist in PacketFence, such as DHCP/DNS, which scales well. At Inverse, we also added VLAN-based isolation to PacketFence, which makes this solution more secure and appealing for large networks. PacketFence is an ideal solution for securing campus networks or even a part of your network (a VLAN or subnet). Finally, a high-quality NAC solution has emerged from the FOSS community, and the evolution of PacketFence is promising. ■

Ludovic Marcotte (ludovic@inverse.ca) holds a Bachelor's degree in Computer Science from the University of Montréal. He is currently a senior systems architect for Inverse, Inc., an IT consulting company located in downtown Montréal that specializes in the deployment of infrastructures based on free and open-source components.

Dominik Gehl (dgehl@inverse.ca) holds a Master's degree in Computer Science from the University of Montréal. He is currently a systems architect for Inverse, Inc., an IT consulting company located in downtown Montréal that specializes in the deployment of infrastructures based on free and open-source components.

Resources

PacketFence: www.packetfence.org

Snort: www.snort.org

Bleeding Edge Threats: www.bleedingsnort.com

Oinkmaster: oinkmaster.sourceforge.net

Nessus: www.nessus.org

Azureus: azureus.sourceforge.net

Hear Yourself Think Again!



WhisperStation™ **Cool... Fast... Silent!**

For 64-bit HPC, Gaming and Graphic Design Applications

Originally designed for a group of power hungry, demanding engineers in the automotive industry, WhisperStation™ incorporates two dual core AMD Opteron™ or Intel® EM64T™ processors, ultra-quiet fans and power supplies, plus internal sound-proofing that produce a powerful, but silent, computational platform. The WhisperStation™ comes standard with 2 GB high speed memory, an NVIDIA e-GeForce or Quadro PCI Express graphics adapter, and 20" LCD display. It can be configured to your exact hardware specification with any Linux distribution. RAID is also available. WhisperStation™ will also make a system administrator very happy, when used as a master node for a Microway cluster! Visit www.microway.com for more technical information.

Experience the "Sound of Silence".

Call our technical sales team at 508-746-7341 and design your personalized WhisperStation™ today.



Microway
Technology you can count on™

Need for Speed: PS3 Linux!

Turn your PS3 into a dual-boot game machine and Linux box. DAVE TAYLOR

Can we get the hottest video game system from Christmas 2006 and turn it into a Linux box? You bet!

If you're still thinking about video game systems as being just a wee bit more technologically advanced than an old Coleco or Atari 800, you've got quite a surprise coming the first time you crack open the proverbial hood. Although the new Nintendo Wii (pronounced "we", oddly enough) has some slick hardware, as does the Microsoft Xbox 360 device, the real winner in the technology race is the rather amazing Sony PlayStation 3 system.

Built around an IBM Cell Broadband Engine processor, the PS3 includes a high-def Blu-ray drive, four USB 2.0 ports, an NVIDIA graphics processor with 256MB of separate video RAM, support for CompactFlash, SD and memory stick devices, Ethernet, built-in 802.11b and g, Bluetooth, an HDMI port and support for all the video resolutions you can imagine, including 480i, 480p, 720p, 1080i and the holy grail, 1080p. Sounds like a computer, not a video game system, doesn't it?

The Blu-ray optical drive system boasts support for most of the older disc formats too, including CD-ROM, CDR+W, DVD, DVD-ROM, DVD-R and DVD+R. If you're not familiar with the battlefield of HD video, Blu-ray can support up to 54GB of data per disk, which is pretty amazing if you're still using a CD burner! The other HD disk format is HD-DVD, but that's not supported. On the other hand, Blu-ray already has more than a hundred movies in retail channels, so the PS3 is also quite a capable HD video player.

The Cell processor itself is a pretty amazing piece of hardware, sufficiently so that Terra Soft Solutions (Loveland, Colorado) has worked with Sony to create a PS3-based supercomputer center. Imagine, hundreds of rackmounted PS3 devices running complex weather simulations rather than WWII games. The Cell processor runs at speeds greater than 4GHz and can handle 256 billion calculations per second, with 2.5MB of memory on the chip itself, squeezed in with a processor design that uses 234 million transistors (www.ibm.com/developerworks/power/cell).

One more important spec: with a 60GB hard drive included, the PS3 will run you about \$599 US at most retail outlets, if they even have PS3 units in stock. And games? Well, there are a few dozen available at this point with an average price of about \$60 US, and the standard game rental channels (GameFly, Hollywood Video and so on) should have PS3 games available for rent by the time you read this article.

Okay, so it's a darn cool computer with some terrific capabilities hiding in a sleek black shell, but is it really just useful for playing video games or can you do something else with it—can you turn it into a Linux system?

I Admit, I Liked WebTV

Perhaps the most obvious question to ask is why bother? I mean, if you buy a PlayStation 3, you're going to be investing in the fastest next-generation gaming console on the planet. Why the heck would you want to boot in to a sterile Linux environment instead?

Well, the answer isn't because the PS3 replaces your regular Linux box if you're a geek. To me, the question of running Linux on the PS3 revolves much more around whether you can essentially add functionality to the PS3 for households that don't have a computer. What if you could run all the PS3 games, watch Blu-ray HD movies and gain full interactivity with the Internet too?

It's the 21st-century answer to the late, underappreciated WebTV device. Tech geeks never quite got the point that a lightweight device with a wireless keyboard that hooked up to a regular TV was



PLAYSTATION 3 IMAGES COURTESY OF SCEA

never intended to compete with a \$5,000 Alienware Gaming PC. But WebTV still had great utility to those people who didn't want—or couldn't figure out—a personal computer, be it a Mac or PC. Simple, simple, simple. WebTV offered a Web browser and e-mail system and not much else, and for many people, that was just fine.

Think along these lines, and Linux on the PS3 suddenly seems like something that's worth doing, because the PS3 hardware is so darn powerful and capable. The target audience isn't people who could easily run Windows Vista, Mac OS X Leopard or Linux on a separate computer, but those people who would find the lightweight solution just fine. As a result, my primary testing for this solution are those two killer apps: Firefox for Web surfing and Thunderbird for e-mail.

First, Configuration Requirements

Sony actually contracted with Terra Soft Solutions to produce a version of its Yellow Dog Linux (henceforth YDL) for the PlayStation 3, a smart move considering that Linux people were going to cobble together a solution anyway. Terra Soft initially produced YDL for the IBM-chip-based Mac PowerPC systems, offering up a quite capable Linux alternative to Mac OS X.

The PS3 doesn't include any useful input devices (other than a game controller), so you need to buy a USB keyboard and mouse or, perhaps, just dig one out of your closet like I did. You'll also want a USB Flash drive for a temporary boot drive. Fortunately, I have a 2GB SanDisk Flash drive that worked just fine. They're about \$70 US at your local computer shop. Ironically, my Flash disk came from Microsoft, with Vista promotional materials pre-installed—not anymore!

Start by partitioning the hard disk in your PS3 so you have space to install Linux. This is pretty easy. Boot up the PS3, then go to Settings→System Settings→Format Utility, choose Format Hard Disk, say yes to the questions about reformatting the entire disk, and then eventually you'll be able to choose a Custom partition. Choose the Allot 10GB to the Other OS, which still gives you 50GB in the bigger unit or 20GB in the smaller unit (the 30GB model of the PS3) for games and other PS3 stuff. We've come a long way from Pong, somehow....

Now, it's time to turn to your removable drive, whether you're going to use a Flash drive like I did or try a CompactFlash, SD card or similar. You need to create a directory ps3, and then a subdirectory therein called otheros, and download two files, one from the Sony Web site (www.playstation.com/ps3-openplatform, save it as otheros.self) and one from Terra Soft Solutions (www.terrasoftsolutions.com/support/installation/ps3/otheros.bld, save it as otheros.bld).

Armed with both of these files (total size is about 8MB, by the way, so my 2GB drive is vast overkill), eject your drive from your PC or Mac and insert it into one of the USB ports on the PS3 itself. You'll also want to burn a full YDL DVD installation disk based on the OS you can download for free from the Terra Soft Solutions site, or you can just buy an installation package that includes both an install and source DVD disk, installation guide and lots of additional goodies, including six months of support, for \$99 US.

Now, it's time to install Linux. Let's hope it doesn't mess up my fancy \$600 video game system, eh?

To install, go to Settings→System Settings→Install Other OS, and the bootloader should be found automatically and be selectable. As always with the PS3, the X button on the controller selects the specified choice and lets you proceed.

Uh oh, I hit my first snag, with the bootloader complaining "No

ASA COMPUTERS

Want your business to be more productive?

The ASA Servers powered by the Intel® Xeon™ Processor provides the quality and dependability to keep up with your growing business.

Hardware Systems For The Open Source Community—Since 1989 (Linux, FreeBSD, NetBSD, OpenBSD, Solaris, MS, etc.)

Dempsey/Woodcrest Server Starts at \$2395

- 1U Dual Core 5030 CPUs.
- 4GB FBDIMM Memory.
- Supports upto 64GB FBDIMM.
- 120 GB hotswap hard drive.
- 2xintegrated Dual 10/1000 LAN.



Dual Xeon Server starts at \$4139

- 5U Dual Xeon 2.8 Ghz 800 FSB.
- iSCSI or NAS Software options.
- 8x120 GB Sata Hard drives -Upto 18TB.
- 512 MB RAM
- Fall hard drive LED indicator.



Dual Xeon 800FSB Storage starts at \$6,699

- 8U Dual Xeon 2.8 Ghz 800FSB CPUs.
- 2TB of storage (36TB max).
- 1GB RAM
- NAS or iSCSI software options
- 2 x 10/100/1000 Gigabit LAN.



Your Custom Appliance Solution

Let us know your needs, we will get you a solution



ASA Collocation

\$75 per month for 1U Rack - 325 GB/month

ASA Collocation Special

First month of collocation free.*

Storage Solutions

IDE, SCSI, Fiber RAID solutions
NAS, DAS, iSCSI, SATA, SAS
3Ware, Promise, Adaptec,
JMR, Kingston/Storcase solutions

Clusters

Rackmount and Desktop nodes
HP, Intel, 3Com, Cisco switches
KVM or Cyclades Terminal Server
APC or Generic racks

All systems installed and tested with user's choice of Linux distribution (free). ASA Collocation—\$75 per month



2354 Calle Del Mundo,
Santa Clara, CA 95054

www.asacomputers.com

Email: sales@asacomputers.com

P: 1-800-REAL-PCS | FAX: 408-654-2910



Intel®, Intel® Xeon™, Intel Inside®, Intel® Itanium® and the Intel Inside® logo are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Prices and availability subject to change without notice. Not responsible for typographical errors.



Figure 1. YDL

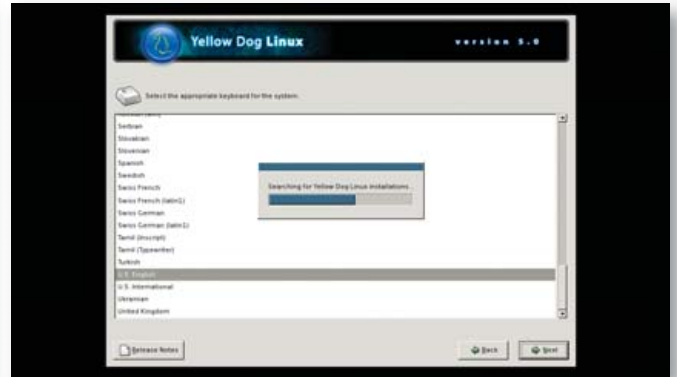


Figure 2. Installing YDL



Figure 3. YDL Partitioning Warning



Figure 4. YDL Checks for Dependencies

applicable [sic] install data was found" (yes, they didn't fix the spelling error). Because the file from Terra Soft initially unpacked with the file-name `exoboot`, I tried renaming it thusly to see what happens. Nope, somehow that meant that the PS3 didn't find any possible external bootloader. Ah, perhaps it's a Mac versus PC problem, because I downloaded and copied the files onto the USB device with my Mac. Okay, I reformatted the thumbdrive, redownloaded and re-installed the two files onto the USB drive with my trusty Windows XP device.

That was the problem—most confusing, because on the Mac I saw a download of `others.bld.gz` that unzipped to a file called `exoboot`, which I simply renamed to `others.bld`. It was corrupted somehow, because when I downloaded the two files onto the thumbdrive from the PC and then tried to install the OS, it worked like a charm!

Now, it's time to tell the PS3 that you want to boot in to the new system, rather than the default PS3 operating system. This is done by going to `Settings`→`System Settings`→`Default System` and selecting `Other OS`. Before you reboot, however, install the YDL install DVD and hook up your USB keyboard and mouse.

Flip the power switch to reboot the PlayStation 3, and after a few seconds, it'll read the install DVD and pop up with the familiar penguin and a long stream of boot messages, just as us Linux folks are used to by this point in time. You'll then get a prompt `kboot`: at which point you can simply press `Return` to boot YDL or type in `boot -game-os` to get back to the world of the PS3.

Tip: you also can reboot into the Game OS by holding down the power button for five seconds when you power on. It'll ignore the Linux partition from then on, however, until you go into the `System Settings` and choose `Default System`→`Other OS` again.

After a minute or two of streaming text, you'll get to the YDL Version 5.0 install screen in Anaconda, where you can now start clicking on `Next` until your mouse button gets tired. Actually, just a few clicks in you'll find that the system complains that "The partition table on device `sda` was unreadable" and asks if you want it to initialize the drive, erasing all data. You do want to do that, and as always, I recommend you choose automatic partitioning.

The rest of the installation is pretty typical of a Linux system, with root passwords and so on. All told, it took about an hour to install everything onto the PlayStation 3 from the YDL install DVD—perfect time to check your e-mail or grab a cup of tea!

Finally, finally, a `Reboot` button lets me restart the PS3 with its newly installed other OS. Compared to the beautiful PlayStation 3 user interface, I have to say that a Linux reboot sequence is sure ugly!

Again, as with a typical Linux install, I now see a series of first boot configuration options, including setting the date and time, specifying an initial nonroot user and, unlike many Linux installations in my experience, the YDL for PS3 installation correctly recognizes and configures the system for the PS3 soundcard.

One login later, and I'm running the X Window System with

Enlightenment as the theme and find that Firefox is already conveniently installed. Even better, the system has by default correctly found my DHCP server and configured itself so that I'm on-line and ready to go.

Surf the Net in PlayStation 3 Linux

Now, I can start to analyze whether the YDL installation is actually a configuration that addresses my earlier stated needs for a software solution that makes the PS3 a useful Internet machine, and a quick visit to linuxjournal.com confirms that, yes, it works fine, it's darn fast and eminently usable. Nice!

One of the sites I use as a test is Google's Gmail service. It's complex behind the scenes and quite powerful, so the question is always whether it works and renders properly on a new system. YDL came through like a champ, working just fine and letting me navigate through my e-mail securely through Firefox. Thunderbird is also pre-installed and ready to go, and configuring a POP3-based e-mail account is pretty straightforward for most Linux users, so there are at least two good avenues for accessing your e-mail.

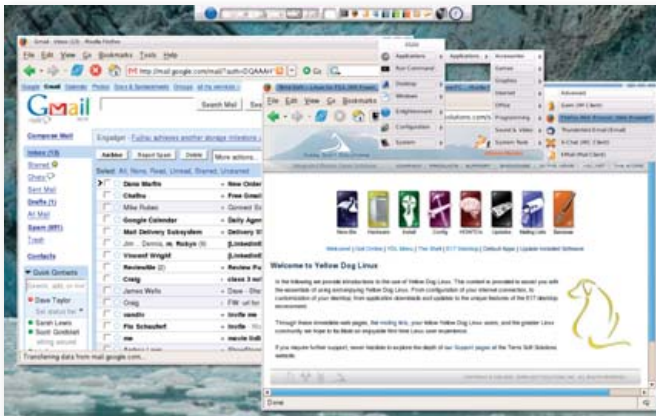


Figure 5. Testing Google's Gmail with YDL

That means, of course, that YDL does indeed meet my primary criteria for usability, letting me surf the Web and interact with my e-mail, all from the comfort of my easy chair and with a simple USB keyboard added onto my slick PlayStation 3 device.

But, Linux offers a lot more capability, and as an experiment, I launched Rhythmbox and quickly concluded that I have had my expectations of music players really screwed up by using iTunes for so many years. It's astonishing to me that I can choose "Internet radio stations" and not get a list of available stations, but instead have to figure out the URL of the station I desire so I can "tune in" to it. Unfortunately, all these years into the Linux evolution, and there are still too many apps that are rough around the edges like this.

I went to Firefox, searched for "internet radio station jazz", found one through the popular Live365 site, selected the channel, had it try to download a streaming file that caused the launch of the Helix player, only to find that it doesn't have the capability of playing back that type of content. Next stop: AccuRadio, but it wanted me to install a new plugin. Yech. New Orleans Jazz channel WWOZ offered up a URL, so I pasted that into Rhythmbox just to find it didn't work either. To heck with it! How is someone like my Mom supposed to survive so much hassle to get audio in YDL?

At the End of the Day, It's a Linux System

As I expected, it may be slick and fast running on the Sony PlayStation 3 with its powerful Cell processor system, but it's still the same Linux that we've gotten used to with no exciting new capabilities, no easier way to work with the various media on the Web, and the same rough edges I've been bothered by for over a decade now.

Unlike most Linux systems, however, YDL on PS3 at least lets you reboot and go back into the world of the PlayStation, where you can easily run photo slideshows, upload and enjoy your music library, watch DVD and Blu-ray HD video and, of course, play some of the amazing games available for the PlayStation.

Really, it's one heck of a combination, and if you know someone who would like to have access to all the power and capabilities of the Cell processor through Terra Soft Solution's YDL system, along with the fun and power of the PlayStation 3, it's really one heck of a combination. Even if you just want to hack, it's cool to have a foreign OS on the system as an option at boot time too. ■

Dave Taylor has been poking around in UNIX and Linux since the mid-1980s and has contributed various software to its evolution. He has run Linux on all sorts of strange devices now, including his tri-booting Mac laptop and Sony PlayStation 3. He also runs a busy Q&A and troubleshooting site (AskDaveTaylor.com) and has written a number of popular tech books, including *Growing Your Business with Google* and *Wicked Cool Shell Scripts*.

Data Acquisition & Control Computer

iPac 9302

- Cirrus Logic EP9302 ARM9 200 Mhz Processor
- Floating Point Math Engine
- 2 USB 2.0 Host Ports
- SD/MMC Flash Disk Slot
- 48 Digital GPIO Lines
- 1 10/100 Base-T Ethernet port
- 5 channels of 12 bit A/D & 3 PWMs
- 1 RS232 & 1 RS232/422/485 Serial Port
- Battery Backed Real Time clock/calendar
- Eclipse uClinux Development Environment





2.6 Kernel

The iPac has enough I/O for demanding applications & with a size of 3.5" x 3.8" it can fit almost anywhere. Prices start at \$150.00. Please contact us for more information.

Since 1985
OVER
22
YEARS OF
SINGLE BOARD
SOLUTIONS

EMAC, inc.

EQUIPMENT MONITOR AND CONTROL

Phone: (618) 529-4525 • Fax: (618) 457-0110 • Web: www.emacinc.com

The OpenSSH Protocol under the Hood

The nitty-gritty details as to what OpenSSH is and why it is ubiquitous.

GIRISH VENKATACHALAM

Is there a program more commonly used in day-to-day Linux computing than SSH? I doubt it. Not only is it rock-solid, secure and versatile, but it also is extremely simple to use and feature-rich. Because its algorithms and protocols are both state of the art and their implementation is open for peer review, we can rest assured on the cryptographic integrity of SSH. SSH does have weaknesses, however; although most of them stem from social engineering, and working around broken protocols, such as X11, pose a big challenge.

SSH can do wonders in only a few lines of C code—thanks to the UNIX philosophy of stringing together powerful tools in generic ways.

SSH acts as a secure channel, and it makes a remote system appear local, and a local one appear at the remote side. It can be used either for remote command execution, with or without a pty, and it can be used for multiplexing several TCP and X11 sessions. It also can be used for tunneling insecure protocols, such as POP3 or SMTP, through secure SSH tunnels. In addition, it can be used with some limitations to tunnel FTP securely.

The OpenSSH Architecture

Let's begin with the overall scheme of things.

As shown in Figure 1, OpenSSH is composed of three key layers. The bottom layer, ssh-transport, is the most critical component involved in all the crypto operations, such as key exchange, re-keying at intervals, protecting against attacks in various ways and so on.

The layer on top of that, ssh-userauth, is responsible for authenticating end users to the sshd daemon that runs at the server end. Remember that SSH authenticates both ways. The client SSH program authenticates the sshd server daemon using the ssh-transport protocol. After authentication, key exchange is completed, and a secure connection is established. Subsequent to that, user authentication takes place in the ssh-userauth layer.

ssh-userauth provides a lot of flexibility, because users can authenticate to the server in various ways—from a private key on a smart card to simple user name/password authentication. Once it goes through, the ssh-connection layer establishes a secure channel, either for executing a remote command or to obtain an interactive login shell.

The ssh-connection layer is capable of multiplexing any number of simultaneous independent secure sessions over a single ssh-userauth layer with the transport stack layer below it, as shown in Figure 1. All of SSH's magic—forwarding arbitrary TCP ports from local to remote and remote to local, acting as a SOCKS proxy, forwarding X11 connections, establishing VPN tunnels, executing remote commands with and without a pty—is done with the ssh-connection layer.

SSH has flow control built in to the protocol. Each secure channel has a separate window size allocated. Because SSH operates above a

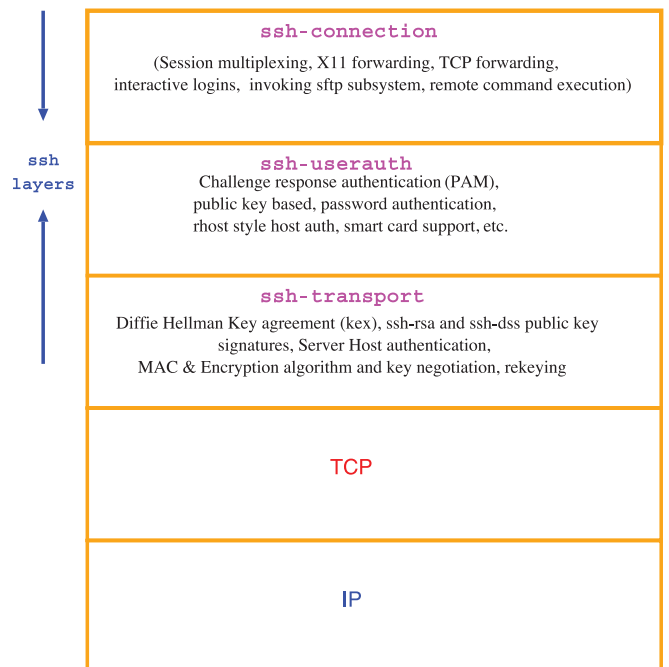


Figure 1. OpenSSH Architecture

reliable TCP layer, this does not have much of a role. At least, it is not as critical as the TCP windowing mechanism. Most of the critical channel open/close messages and other termination messages don't consume any window space.

Because all messages are encrypted and integrity-protected, nobody can interpret the messages. There is a special SSH_MSG_IGNORE message type that can be used for defeating traffic analysis attacks. These are the kinds of attacks that figure out when data is going over the wire and how much data is being transferred.

SSH, of course, comes with many other niceties for sending secure KEEPALIVE messages, redirecting stdin to /dev/null for specialized X window applications and many more.

Now, let's take a look at a sample SSH session and typical message exchanges (Figure 2).

Here is a typical unencrypted SSH packet:

```
byte      SSH_MSG_CHANNEL_REQUEST
uint32    recipient channel
string    "pty-req"
boolean   want_reply
```

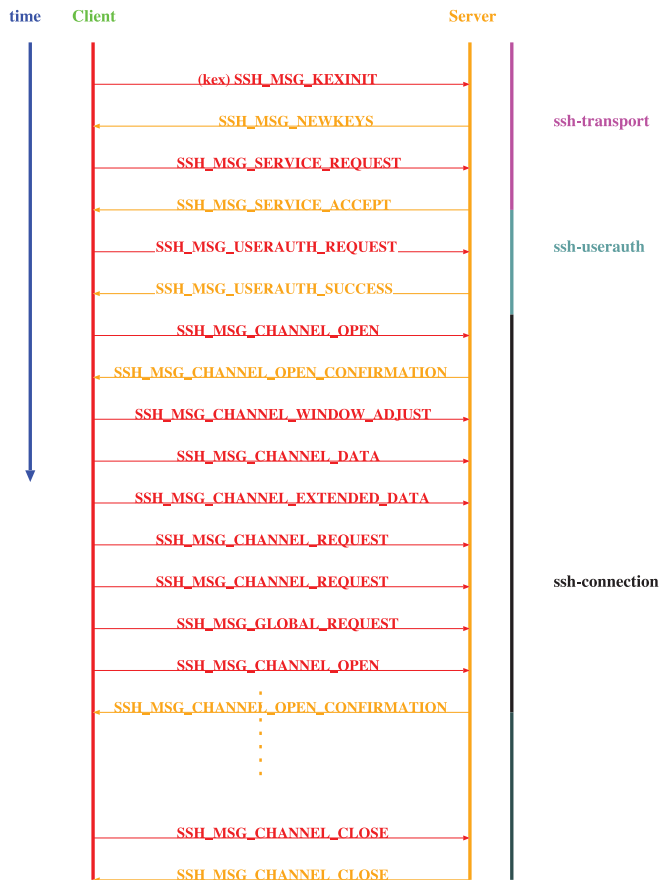


Figure 2. OpenSSH Protocol Flow Diagram

```
string TERM environment variable value (e.g., vt100)
uint32 terminal width, characters (e.g., 80)
uint32 terminal height, rows (e.g., 24)
uint32 terminal width, pixels (e.g., 640)
uint32 terminal height, pixels (e.g., 480)
string encoded terminal modes
```

Most fields are self-explanatory. The top two fields are always present in all messages. The payload packets (what the user types and the responses from the server) are all carried with the SSH_MSG_DATA message type.

Every packet has a header that describes the contents of the payload (message type) and the channel for which it is destined.

Some of the messages do not need a response from the other side, as the underlying layer is not only reliable but also tamper-resistant. But, most requests from the client have a corresponding response from the server.

Now, let's get to the gory details of the SSH key exchange protocol, because that is the most critical component that accounts for the security and popularity of SSH.

Figure 3 shows the data manipulations that are necessary to encrypt, compress and integrity-protect. Of course, we need to protect ourselves against replay attacks as well. For that, there is an implicit

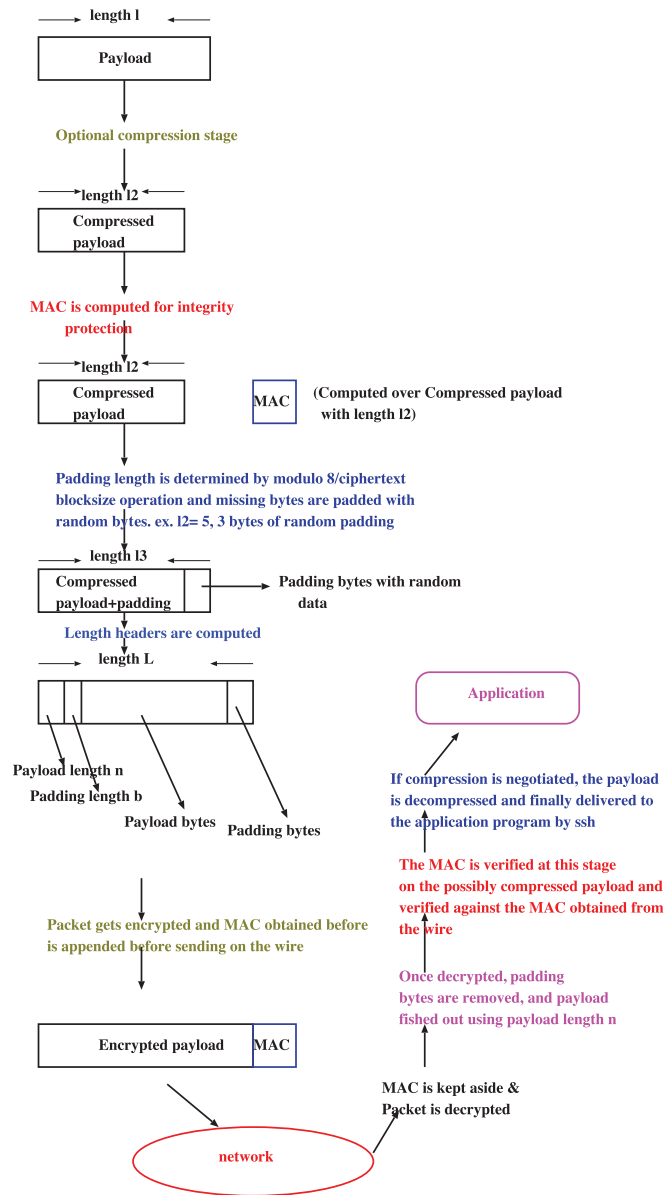


Figure 3. OpenSSH Packet Processing

sequence number for each packet, and it starts at 0 and goes to 2^{32} before wrapping around. Because the sequence number is hashed, it can be sequential, and attackers never can guess what input will lead to what hash.

The key components of OpenSSH keys are:

- Hash: H.
- Shared secret: K.
- Session ID: session_id.

SSH uses the above components to derive the following encryption

SSH can do wonders in only a few lines of C code—thanks to the UNIX philosophy of stringing together powerful tools in generic ways.

vectors and keys:

- Client to server initialization vector.
- Server to client initialization vector.
- Client to server encryption key.
- Server to client encryption key.
- Client to server MAC key.
- Server to client MAC key.

The equations used for deriving the above vectors and keys are taken from RFC 4253. In the following, the || symbol stands for concatenation, K is encoded as mpint, "A" as byte and session_id as raw data. Any letter, such as the "A" (in quotation marks) means the single character A, or ASCII 65.

- Initial IV client to server: $\text{HASH}(K \parallel H \parallel \text{"A"} \parallel \text{session_id})$.
- Initial IV server to client: $\text{HASH}(K \parallel H \parallel \text{"B"} \parallel \text{session_id})$.
- Encryption key client to server: $\text{HASH}(K \parallel H \parallel \text{"C"} \parallel \text{session_id})$.
- Encryption key server to client: $\text{HASH}(K \parallel H \parallel \text{"D"} \parallel \text{session_id})$.
- Integrity key client to server: $\text{HASH}(K \parallel H \parallel \text{"E"} \parallel \text{session_id})$.
- Integrity key server to client: $\text{HASH}(K \parallel H \parallel \text{"F"} \parallel \text{session_id})$.

Simple, right?

What is not simple, however, is figuring out the K and H parameters.

HASH is usually an SHA1 hash mechanism, but it can be something else as well.

The typical cipher algorithm used is AES or DES3 in CBC mode. The MAC is a combination of MD5 or the SHA1 hash algorithm with a secret key. There are four choices here:

- hmac-sha1
- hmac-md5
- hmac-sha1-96
- hmac-md5-96

Actually, sha1 is a little weak in today's world, because collision attacks are possible. The zeitgeist in hashing today is sha512, but with proper re-keying and other smarts built in, it should not be a problem.

Remember that hashes are of a constant length, so hmac-sha1 is

20 bytes long, hmac-md5 is 16 bytes, and the other two have a fixed length of 12 bytes each.

Okay, now for some mathematical and crypto gymnastics of the kex stage.

We know how to compute the individual encryption and MAC keys provided that we derive the basic parameters using the simple equation above. But, how do we get the parameters to begin with, in a secure, authenticated manner?

Now, we need to look at how OpenSSH uses diffie-hellman-group14 and diffie-hellman-group1 fields to derive the DH generator and DH moduli for an anonymous key agreement. However, this leaves us open to several man-in-the-middle and other active attacks. To thwart this, we use a known and trusted server public key to authenticate key exchanges. Authentication of key exchange data is nothing more than signing with a private key. And, OpenSSH typically uses ssh-dsa or ssh-rsa keys for this purpose.

In other words, a combination of DH and RSA/DSS keys are used for authentication and to derive the secret parameters K, H and session_id. session_id is simply the hash of the first key exchange. A 16-byte random cookie also is used to protect against replay and other man-in-the-middle attacks.

Here is the equation for deriving H:

$$H = \text{hash}(V_C \parallel V_S \parallel I_C \parallel I_S \parallel K_S \parallel e \parallel f \parallel K)$$

- hash is usually the SHA1 hash algorithm.
- V_C and V_S are the client and server identification strings.
- I_C and I_S are the client and server SSH_MSG_KEXINIT messages just exchanged.

Now, we are left with computing e, f and K; e and f are the DH parameters used for exponentiation:

- $e = g^x \text{ modulo } p$
- $f = g^y \text{ modulo } p$
- $K = e^y \text{ modulo } p$

Here, p is a prime number from the DH generator field. And, x and y are chosen arbitrarily by client and server. Remember that DH works using the simple mathematical principle that $a^{bc} = a^{cb} = a^{bc}$.

Now, we have everything required for computing the secret keys.

The nice thing about all of these cryptographic parameters is that they are thrown away after every session. The only reused parameter is the server RSA/DSA key, but because we add a random cookie in our calculations, it's difficult for attackers to break SSH cryptographically.

Description of Each Component

Let's take a look at the OpenSSH family before we proceed.

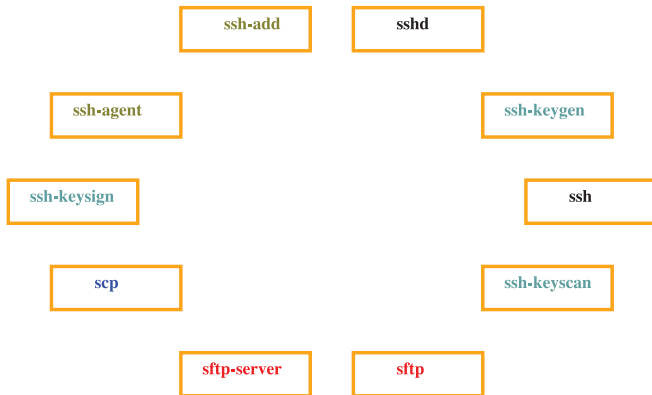


Figure 4. Stars in the OpenSSH Galaxy

As you can see in Figure 4, there are many executables and players in the grand scheme of things. However, the interplay is not a complex one. Everything I discussed above is actually implemented by SSH and sshd components (client and server, respectively). The other components are used rarely for key generation, agent forwarding and so on.

sftp-server is the subsystem for SSH. This is an FTP-like protocol, but it is highly secure and efficient, unlike the broken FTP protocol.

scp is a marvelously popular and convenient file transfer mechanism built on top of the SSH infrastructure. Because integrity protection is built in to the SSH wire protocol, file integrity is guaranteed. However, it does not have a resume feature for broken transfers, so you have to use it with rsync to get that facility.

Security Analysis and Attacks

Now, let's look at the kind of attacks and threat models SSH helps us guard against.

One of the most critical components of any cryptographic protocol is the quality of the random number generator. Because computers are deterministic devices, obtaining truly random data is a challenge. Common sources of entropy include disk access, keyboard and mouse input, process lifetimes and so forth. An incredibly large number of traditional UNIX programs have relied on the `gettimeofday(2)` system call. SSH also uses sound mechanisms to check the randomness of the pool of data.

One interesting attack specific to SSH is using control character sequences to terminate sessions and interfere with pty interactions, so we have to filter out suspicious character sequences.

The most critical and, unfortunately, the weakest point of SSH is server/host authenti-

cation. Reality and typical user negligence proves that we just say yes whenever a new host key is added to our trusted list. Efforts are underway to make this more secure and easier. If this is not ensured, different types of man-in-the-middle attacks are possible. ■

Girish Venkatachalam is a cryptographer with nearly a decade of experience working on various modern UNIX systems. He has developed IPsec from scratch on the Nucleus OS for a router and worked with the guts of Apache, OpenSSL and SSH. He can be reached at girish1729@gmail.com.

Resources

OpenSSH: www.openssh.org

SSH Protocol Architecture: www.ietf.org/rfc4251.txt

ssh-userauth: www.ietf.org/rfc4252.txt

ssh-transport: www.ietf.org/rfc4253.txt

ssh-connect: www.ietf.org/rfc4254.txt

Ultra Dense, Powerful, Reliable... Datacenter Management Simplified!

15" Deep, 2-Xeon/Opteron or P4 (w/RAID) options



Customized Solutions for... Linux, BSD, W2K

High Performance Networking Solutions

- Data Center Management
- Application Clustering
- Network and Storage Engines

Rackmount Server Products

- **1U Starting at \$499:** C3-1GHz, LAN, 256MB, 20GB IDE
- 2U with 16 Blades, Fast Deployment & more...



iron
SYSTEMS™

Iron Systems, Inc.

540 Dado Street, San Jose, CA 95131

www.ironsystems.com

CALL: 1-800-921-IRON

Starting a Linux Firewall from Scratch

The first steps in getting started with iptables. DINIL DIVAKARAN

Building a firewall is something that easily can be done using a Linux machine. This article describes the basic steps involved in developing a firewall from scratch, using tools in Linux. It is intended for newbies interested in learning about (Linux) firewalls. More important, this article is for all new administrators who would like to dirty their hands and get a firewall up and running as soon as possible, but without missing the important concepts en route. My experience in working on a Linux-based firewall at the DON (Distributed and Optical Networking) lab, in the department of Computer Science and Engineering at the Indian Institute of Technology (IIT) Madras, is the most motivating factor behind writing this article.

In this article, we examine developing a firewall that will sit on the edge, separating your private network from the rest of the world; therefore, the firewall also will act as a gateway.

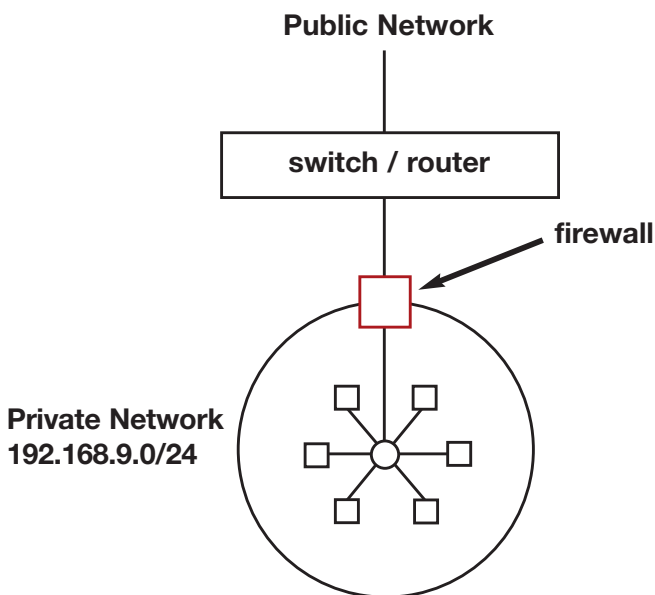


Figure 1. Firewall Diagram

First of all, why do you need such a firewall? Most important, you need to restrict access to machines in your network, a network that might consist of various servers. One of them might be a mail server, and another might be a DNS server, but only those particular services (provided by these servers) need to be accessed, not anything and everything on the network. Putting it simply, firewalls are used to protect a private network from the rest of the world—call it a public network (which is the Internet in most scenarios).

One less obvious reason for having a firewall is that it is neces-

sary to block all unwanted traffic flowing into or through your network, which might otherwise throttle the bandwidth. Such traffic should ideally stop at the gate (gateway or firewall). One good example is when there are many subnetworks, such as at a college or university campus. One of the machines in such a subnetwork could become infected with a virus and might flood or broadcast ARP packets. Similarly, some Windows PCs from outside the private network might be broadcasting netbios (netbios-ns/netbios-dgm) packets, which are meaningless to your network and, therefore, should be blocked by the firewall.

But, some of the ARP packets might be legitimate requests for machines in your network (or subnet). If you block such legitimate ARP broadcast requests, no packet (good or bad) will reach your network, as machines outside the private network will not be able to obtain the Ethernet address corresponding to the IP address of the machine in your network. To solve this problem, you should configure your firewall to act as a proxy for ARP requests—that is, your firewall should reply to the ARP requests.

Now, let's get into the implementation details. Assume your private network is 192.168.9.0/24. Your firewall, which is also a gateway, must have two interfaces: one pointing to your network (eth0) and the other connecting to the public network (eth1).

First, configure the IPs for both interfaces. This can be done using the network configuration tool or with the `ifconfig` command. Ideally, it is better to use the system network configuration tool (system-config-network in Fedora Core 2–5) or edit the configuration files (at `/etc/sysconfig/network-scripts` in FC 2–5), so that the configurations are retained even when the network is started (as part of the boot process) or restarted (manually). You also can configure the IP by appending the `ifconfig` command at the end of `/etc/rc.d/rc.local` (as this file is executed at the end of the boot process). If you do this, however, ensure that these commands are executed when the network is restarted manually.

We use `ifconfig` to be distribution-independent (for lack of a better term).

There is no hard and fast rule on the IP addresses to be used for the interfaces, but generally, the last two IP addresses in the subnet are used for such purposes. Now, assign 192.168.9.253 to eth0 and 192.168.9.254 to eth1:

```
echo "Configuring eth0"
/sbin/ifconfig eth0 192.168.9.253 up
```

```
echo "Configuring eth1"
/sbin/ifconfig eth1 192.168.9.254 up
```

The most important function of a firewall that takes the role of a

gateway is to forward packets. This is how we do it:

```
echo "Enabling IP forwarding"
echo "1" > /proc/sys/net/ipv4/ip_forward
```

Earlier, we said the firewall also should act as a proxy for ARP requests. This means the firewall will reply to the ARP requests querying for the Ethernet address of any machines in your network (192.168.9.0/24). Will the firewall send the MAC address of the machine for which the query was broadcasted (say 192.168.9.8)? No. Instead, it will send its own MAC address, and later, when it receives a packet for 192.168.9.8, it will forward the packet to 192.168.9.8 (of course, only if the rules allow the packet to pass through). Enabling proxy ARP is quite easy in new distributions:

```
echo "Enabling Proxy ARP"
echo "1" > /proc/sys/net/ipv4/conf/eth1/proxy_arp
```

Next, set up the routing entries in the firewall. The private network is reachable through eth0, although packets to the public network should go through eth1:

```
echo "Route to 192.168.9.0/24 is through eth0"
/sbin/route add -net 192.168.9.0/24 eth0
```

```
echo "The default gateway is eth1"
/sbin/route add default eth1
```

Similarly, you have to tell all machines in your network to use 192.168.9.253 as the default gateway (because you have to go through the gateway to access any machine outside your network). LAN machines can be accessed directly. Do the following on all machines (except the firewall, obviously) in your network:

```
echo "Add default route through the gateway"
/sbin/route add default gw 192.168.9.253 eth0
```

```
echo "192.168.9.0/24 is directly reachable"
/sbin/route add -net 192.168.9.0/24 eth0
```

Next comes the firewall rules—rules that protect a network. Rules are written using the iptables tool. This is a very useful tool, although a bit complex, with a detailed man page on the various options. The iptables Netfilter uses three different built-in chains: INPUT, FORWARD and OUTPUT. Packets traverse through the chains, and therefore, the rules are written for specific chains. With respect to your firewall, any packet destined to your firewall (192.168.9.253 or 192.168.9.254) goes to the INPUT chain. If the packet is meant to be forwarded (that is, it is not for

your firewall, and there is a route in your firewall to the destination), it goes through the FORWARD chain. Any packet generated by your firewall will go out from the box through the OUTPUT chain. (This brief explanation is applicable to any Linux box.)

Although you would never want the firewall to forward every packet passing through it, you might want to test whether the functionality of the gateway is working with the above configuration. To do this, make the default policy of the FORWARD chain as ACCEPT (using the -P option)—that is, any packet going through the forward chain is accepted:

```
/sbin/iptables -P FORWARD ACCEPT
```

A ping request from any machine in the network 192.168.9.0/24 (save, the firewall) to any (live) machine outside the network will now return with the ICMP echo reply packet. If the external machine is not reachable, there may be some problem with the cable or network card, or you might have misconfigured something.

Now, let's build the "wall". The easiest way of setting up a firewall is by rejecting (DROP) every kind of packet, and then writing rules to allow (ACCEPT) those packets that you want to see go through. So, let's make the default policy in each of the chains to drop packets.

PEG[®] the leading GUI for Embedded Systems

Prototype Today!
FREE EVALUATION PACKAGE

- PEG+ - Full Featured Windowing in C++
- C/PEG - Smallest Footprint in ANSI C
- Royalty Free
- Fast execution speed
- Completely ROM-able
- Delivered with Full Source Code
- Development Tools including FontCapture, PEG WindowBuilder, and ImageConvert
- Complete set of screen drivers included
- Completely customizable
- Industry leading RTOS Support
- Supports all popular target processors, video controllers and I/O devices
- Multi-lingual support – 2-byte character sets & UNICODE string encoding
- Event-driven programming model
- Application Design Services
- Knowledgeable and timely support to users around the globe
- Now includes a fully licensed version of Paint Shop Pro 9



SWELL software

GRAPHICS SOFTWARE FOR EMBEDDED SYSTEMS

WWW.SWELLSOFTWARE.COM | 810-982-5955

Before doing that, clear all the existing rules:

```
echo "Flush existing rules"
/sbin/iptables -F
```

```
echo "Set the default policy to drop packets"
/sbin/iptables -P INPUT DROP
/sbin/iptables -P OUTPUT DROP
/sbin/iptables -P FORWARD DROP
```

By now, you might have noticed that a rule basically specifies some conditions that the packet must possess. If these conditions are matched, the action specified in the rule is taken, or else the next rule in the chain is checked, and this continues until a rule is matched. If none of the rules in the chain is matched, the default action or policy (here, DROP) is taken.

Let's write our first rule—a rule to allow outgoing SSH packets from the private network:

```
echo "Allow outgoing SSH"
/sbin/iptables -A FORWARD -p TCP -i eth0 \
-s 192.168.9.0/24 -d 0/0 --dport 22 -j ACCEPT
```

This rule is self-explanatory—well, almost. The option `-A` specifies the chain to which the rule is to be appended, and `-p` specifies the

protocol (UDP, TCP, ICMP and so on). The option `-i` names the interface through which the packets will be received. Because the packets are coming from the 192.168.9.0/24 network (the `-s` specifies the source address) for outgoing SSH packets, it will come through `eth0` of the firewall. The destination port (`--dport`) is 22 for SSH traffic. The destination address is indicated with the `-d` option, and `0/0` means any address. Finally, the action for such packets that are matched is ACCEPT (specified with the `-j` option), which means allow the matched packets to go through.

Now, we have written a rule to allow SSH traffic from 192.168.9.0/24 to go anywhere. But, will this work? Will you be able to do an SSH logon from your private network to a machine in the public network? Where have we allowed packets to come from the SSH server (in the public network) back to the client (in the private network)? The following rule achieves that:

```
/sbin/iptables -A FORWARD -p TCP -i eth1 -s 0/0 \
--sport 22 -d 192.168.9.0/24 -j ACCEPT
```

This looks fine, but then we need to write such a rule for every service. Worse, the above rule does more than what is required. It allows any machine to connect to the private network using the source port 22. What we should do instead is append a rule that allows only those packets from the public network that are part of the SSH connections initiated by machines in the 192.168.9.0/24 network.

iptables maintains state information to do such connection tracking. The four states maintained are NEW, ESTABLISHED, RELATED and INVALID. We won't discuss these states in detail here. For the time being, keep in mind that state NEW indicates the packet is part of a new connection. When a response packet is seen in the reverse direction, the connection becomes ESTABLISHED. Note that this has nothing to do with the states in the TCP connection establishment process. An ICMP or UDP reply for the corresponding requests also will mark the connection as ESTABLISHED. Refer to iptables-tutorial.frozentux.net/iptables-tutorial.html#STATEMACHINE to learn exactly how the connection tracking mechanism works. Now (after removing the above rule), to forward all those packets forming part of the ESTABLISHED connection, we write the following rule:

```
echo "Allowing ESTABLISHED connections"
/sbin/iptables -A FORWARD -m state --state \
ESTABLISHED -j ACCEPT
```

This rule ensures that only packets part of an ESTABLISHED connection will be accepted; a new connection request to 192.168.9.0/24 will *not* be accepted. Ideally, to access any services (such as HTTP or FTP), we need to allow only NEW and ESTABLISHED connections to go out (NEW will allow the first packet, ESTABLISHED will allow all following packets of the same connection), and only ESTABLISHED connections to come into the private network. Similarly, if you have a DNS server in your network, which has to be permitted access (queried) from the outside, the following rule does that (assuming that 198.168.9.1 is the DNS server):

```
echo "Allowing incoming DNS requests"
/sbin/iptables -A FORWARD -p TCP -i eth1 \
```

UNIX and Linux Performance Tuning Simplified!

Understand Exactly What's Happening

SarCheck® translates pages of sar and ps output into a plain English or HTML report, complete with recommendations.

Maintain Full Control

SarCheck fully explains each of its recommendations, providing the information needed to take intelligent informed actions.

Plan for Future Growth

SarCheck's Capacity Planning feature helps you to plan for growth, before slow downs or problems occur.



APTITUDE CORPORATION

Request your free demo at www.sarcheck.com






```
-d 198.168.9.1 --dport 53 -j ACCEPT
```

Note that the interface used here is eth1, as the packets from the public network will be received at eth1. (We have not used -s 0/0, as it is added by default.) Also, keep in mind that DNS lookup will succeed only because we already have appended the rule for allowing ESTABLISHED connections to the FORWARD list (yes, UDP traffic also has an associated ESTABLISHED state).

So far, we have blocked every protocol except SSH and DNS. It is a common practice for a new system administrator to block ICMP packets. This is not a good idea, as ICMP packets are useful for many purposes, such as for learning the routes between different interconnected networks in a large LAN, to see if a machine is up, for Path MTU discovery and so on. So, assuming we are sensible administrators, let's allow ICMP packets through the firewall:

```
echo "Allowing ICMP packets"  
/sbin/iptables -A FORWARD -p ICMP -j ACCEPT
```

Earlier, we had blocked any packet to and from the firewall box (using INPUT and OUTPUT chains). For diagnostic purposes, we can allow ICMP packets through both chains—that is, allow ICMP packets to and from the firewall:

```
echo "Allowing ICMP packets to the firewall"  
/sbin/iptables -A INPUT -p ICMP -j ACCEPT
```

```
echo "Allowing ICMP packets from the firewall"  
/sbin/iptables -A OUTPUT -p ICMP -j ACCEPT
```

The ICMP packets also can be rate-limited (as a precaution against ICMP-based attacks):

```
echo "Limit ICMP requests to 5 per second"  
/sbin/iptables -A FORWARD -p icmp --icmp-type \\  
    echo-request -m limit --limit 5/s -j ACCEPT
```

We also might choose to ignore ping broadcasts—that is, ICMP packets to broadcast addresses, such as ping 192.168.9.255 (ICMP broadcast requests are used in Smurf attacks):

```
echo "Ignoring ICMP broadcast requests"  
echo 1 > /proc/sys/net/ipv4/icmp_echo_ignore_broadcasts
```

All these rules (commands) will be lost once the system is rebooted; however, iptables has options for saving and restoring these rules. But, a better approach is to save the rules in a file (say, firewall.sh), give it executable permission and append the script name to the end of /etc/rc.d/rc.local. This way, you always can edit and make modifications to the firewall script. ■

Dinil Divakaran is busy trying to learn more about himself and life. In the meantime, he likes to teach and discuss life as well as technology.

Advertiser Index

For advertising information, please contact our sales department at 1-713-344-1956 ext. 2 or ads@linuxjournal.com.
www.linuxjournal.com/advertising

Advertiser	Page #	Advertiser	Page #
ABERDEEN, LLC www.aberdeeninc.com	59	MBX www.mbx.com	9
ACMA www.acma.com	57	MICROWAY, INC. www.microway.com	C4, 69
APPRO HPC SOLUTIONS appro.com	C2	OPEN SOURCE SYSTEMS, INC. www.opensourcesystems.com	43
APTITUDE CORPORATION www.sarcheck.com	80	POGO LINUX www.pogolinux.com	11
ASA COMPUTERS www.asacomputers.com	33, 71	POLYWELL COMPUTERS, INC. www.polywell.com	63
AVOCENT CORPORATION www.avocent.com/ice	1	THE PORTLAND GROUP www.pggroup.com	29
CARLNET www.carl.net	89	RACKSPACE MANAGED HOSTING www.rackspace.com	C3
CORAID, INC. www.coraid.com	5	R CUBED TECHNOLOGIES www.rcubedtech.com	41
COYOTE POINT www.coyotepoint.com	3	SERVERS DIRECT www.serversdirect.com	49
EMAC, INC. www.emacinc.com	73	SILICON MECHANICS www.siliconmechanics.com	23, 65
EMPERORLINUX www.emperorlinux.com	31	SUN JAVA ONE CONF. java.sun.com/javaone	67
FAIRCUM www.faircum.com	39	SUPERMICRO www.supermicro.com	15
GECAD TECHNOLOGIES/AXIGEN www.axigen.com	6	SWELL SOFTWARE, INC. www.swellsoftware.com	79
GENSTOR SYSTEMS, INC. www.genstor.com	37	SYSGO AG www.sysgo.com	19
HURRICANE ELECTRIC www.he.net	47	TMP WORLDWIDE www.tmp.com	17
IRON SYSTEMS www.ironsystems.com	77	TECHNOLOGIC SYSTEMS www.embeddedx86.com	10
LINUX JOURNAL www.linuxjournal.com	53	TYAN COMPUTER USA www.tyan.com	7
LINUXWORLD CONFERENCE & EXPO www.linuxworldcanada.com	95		

Time-Zone Processing with Asterisk, Part II

Part II of our series on time-zone processing with Asterisk. MATTHEW GAST



PHOTO ©ISTOCKPHOTO.COM/PIXONAUT

Last month, I wrote about a system for handling telephone calls with Asterisk that automatically handled the call depending on the time of day at a remote location. Use of the system, however, depended on the user performing the critical task of setting up the remote location's time with a telephone call. Rather than rely on the user to initiate the telephone call manually, it would be easier if the call occurred automatically.

If the setup call occurs automatically, the user won't forget to do it. The initial SIP registration may occur at a very strange hour in the home location, but the SIP registration occurs because a user has plugged something in. Therefore, we know the user is awake and can take a short call. Asterisk provides a management interface that reports when SIP registrations occur and can be used to take action based on it. With a bit of additional processing, a script talking to the manager can initiate calls only when the SIP registration is "new".

Introduction to the Asterisk Manager

The Asterisk Manager reports events processed by Asterisk and accepts commands over the interface. The form of the interface is a text-based protocol that separates event reports and commands into clusters of lines with a key: value format. For example, the registration of extension 300 using the SIP protocol looks like this:

```
Event: PeerStatus
PeerStatus: Registered
Peer: SIP/300
```

Before gaining access to the Asterisk manager, clients must

authenticate against the list of administrative users stored in `/etc/asterisk/manager.conf`. Once logged in, a client can issue commands or query the value of variables with a set of lines that starts with a first line of `Action:`, followed by a command. Responses to commands typically start with `Response: Success`.

Because the protocol is text-based, it can be scripted in a language like Expect. A component is also available for the Perl Object Environment (POE), a framework that builds event-driven programs in Perl. The freely available component provides the base-level response parsing that would need to be written in Expect, so it is a much more extensible foundation for programs controlling the Asterisk manager.

The Program Core and Inline States

The main code for the program is simple. POE sets up a system where state handlers are called in response to program states. A state can be defined by the programmer or by an external event. The typical flow through the program is to notice a SIP registration, check to see whether it has an active time-zone registration and if not, to initiate a configuration call.

To execute code in response to an event, the POE framework uses a hash called `Callbacks`. Every entry in `Callbacks` defines a state based on the event received from the manager. When an event matches a callback, the handler defined for the state is triggered. To set up a trigger with the `Callbacks` clause, identify every line in the event and set up a hash so that the left-hand side of each line of the event is the key value for a line of the hash. As an example, consider the callback definition for the SIP registration event earlier:

```

Event: PeerStatus      register => { 'Event' => 'PeerStatus',
PeerStatus: Registered      'PeerStatus' => 'Registered', }
Peer: SIP/300

```

To link the callback to a handler, the inline_states hash has a list of states and references to the corresponding code to call. Although it is possible to inline event-handler code, for readability I have separated the code out into external procedures. Code called in response to a Callback cannot be passed arguments:

```

123456789*123456789*123456789*123456789*123456789*12
inline_states => {
  register => \&register_state_handler,
},

```

Based on the flow of the program, three events are of interest. First, SIP registration events are used to start the entire process. SIP registrations typically occur hourly, so it is important to initiate the call only when a registration is the “first” registration. To prevent duplicate telephone calls from being initiated, the program will request data from both the Asterisk internal database AstDB as well as the SIP peer information. The second and third events will handle responses to commands and database queries, respectively. A fourth event will handle initiating the telephone call after receiving data back from the queries. The code I am currently using also has a state defined for unregister events, though it is a stub for an event that I am not currently using.

The core of the program is only 35 lines, most of which defines the program event states and shows what code will be used in response to those states. Note that the state of call is defined by the program and not by a callback, so the call state can be entered only by the program itself and not in response to an event from the manager. (A full listing of the program is available on the *Linux Journal* FTP site; see Resources.)

```

POE::Component::Client::Asterisk::Manager->new(
  Alias      => 'monitor',
  RemoteHost => 'localhost',
  RemotePort => 5038,
  Username   => 'autotzcaller',
  Password   => 'secretpassword',
  CallBacks => {
    input    => ':all',
    response => {
      'Response' => 'Success',
    },
    dbresponse => {
      'Event' => 'DBGetResponse',
    },
    register => {
      'Event' => 'PeerStatus',
      'PeerStatus' => 'Registered',
    },
    unregister => {
      'Event' => 'PeerStatus',
      'PeerStatus' => 'Unregistered',
    },
  },

```

```

},
inline_states => {
  input      => \&input_state_handler,
  response   => \&response_state_handler,
  dbresponse => \&db_response_state_handler,
  register   => \&register_state_handler,
  call       => \&call_state_handler,
  unregister => \&unregister_state_handler,
},
);

POE::Kernel->run();
exit(0);

```

Two of the state handlers are only stubs. The input state handler prints out whatever it gets if a debug flag is set, and it is there for development purposes. It catches any unrecognized events that come from the manager, and it can be useful when testing that callbacks are catching the important events. The unregister state handler currently doesn't do anything, but it is there as a hook to expand if I choose in the future to take any action based on that.

With the core of the program in place, let's look at each of the states in the order they will be called through a typical program execution flow.

Registration Event Handling

The register state handler is called whenever a SIP registration event is received from a new extension. Its main purpose is to get the data required for setting up the configuration telephone call when a new extension pops up. Whether a call is made depends on the state of the extension as far as time-zone processing, so this routine requests information to determine whether the extension is registered, its IP address and other components. To get the extension, we have to take the channel name, which is prefaced with the technology and a slash (for example SIP/) and strip the leading part away.

One wrinkle of the event handler is that POE handlers run to completion. There is no way to interrupt a handler when it is running. The sub-procedure getTZChannelVars will request information on the time-zone offset and IP address, but that information will not become available until the registration handler completes and the responses return via the manager. At the end of the procedure, the registration handler uses the delay_set POE method to queue up the call state for a delay in the future so that the requests will have returned their information by that point. The delay is set by a global variable in the program. I have found that one second is more than adequate for a single-user PBX with only one outstanding extension requiring setup, but the delay is set to three seconds for safety.

Communication between state handlers is a bit different from that in a procedure-driven program. POE state handlers pass references to the POE kernel, which is used in scheduling, as well as the POE heap, which is needed to issue commands to the Asterisk Manager. POE defines constants so the heap and kernel are easily accessible to event handlers as \$_[HEAP] and \$_[KERNEL]. Any other information available is located at \$_[ARG0], which is a constant defined in such a way that it is the first argument.

Any lines in the event that defines the state will be passed as the hash \$_[ARG0] and are accessible by asking for the hash key that appears on the left-hand side of the desired line. In the registration response, it is possible to get at the peer extension by

The Asterisk Manager reports events processed by Asterisk and accepts commands over the interface.

referring to `$_[ARG0]->{Peer}`, which returns SIP/300:

```
Event: PeerStatus
PeerStatus: Registered
Peer: SIP/300
```

On SIP registration, the program needs to identify the extension, request information about it and then set up further processing of the extension data after a delay. When an event is called through the `delay_set` method, it is possible to pass an extension to the state handler, such as the extension number used here:

```
sub register_state_handler {
    my $kernel = $_[KERNEL];
    # Split peer extension off from technology
    my $peer = $_[ARG0]->{Peer};
    debug_print("\tExtension is $peer; ");
    my @exten_parts = split('/', $peer);
    my $ext = @exten_parts[1];
    debug_print("extension number is $ext\n");

    getTZChannelVars($_[HEAP], $ext);

    debug_print("Queuing call event for ");
    debug_print("$REG_CALL_DELAY seconds\n");
    $kernel->delay_set("call", $REG_CALL_DELAY, $ext);
} # register_state_handler
```

As part of the extension registration process, we collect variables about the state of the channel in the `getTZChannelVars` procedure. The POE heap, which is passed as the first argument, can be used to issue commands to the manager. For example, the `put` argument to the server can be used to issue commands. To get the SIP peer data, which includes the current IP address of the peer, the command looks like this:

```
$heap->{server}->put({'Action' => 'SIPShowPeer',
                    'Peer' => $ext });
```

To get a database variable, the action in the `put` command is `DBGet`. The time-zone data is stored as keys in the `tz` family, so it is necessary to specify both the family and assemble the correct key name, which is of the form `300-TIMESKEW` or similar:

```
$heap->{server}->put({'Action' => 'DBGet',
                    'Family' => 'tz',
                    'Key' => $ext . '-TIMESKEW'});
```

Four database requests and the SIP peer data are requested by `getTZChannelVars`. Because this function is called by an event handler, it also is not interruptible. Therefore, it sends four database query

events to the manager, but it does not process responses directly. (Complete code for the five requests within the full procedure is available on the *Linux Journal* FTP site.)

Command and Database Responses

In the gap between issuing requests and the time the call state is scheduled, responses flow in from the SIP data request and the database queries. From the SIP data request, we need to pick out the peer IP address, which appears on a line in the manager response reading `Address-IP: 192.168.1.5`. Conveniently, the POE module parses out the lines in the response, so all we need to do is look for the `Address-IP` line by getting the value of the `Address-IP` hash element in one of the arguments passed to the handler. The POE heap is accessible across events, so adding the value of the SIP peer IP address to the heap makes it accessible to other event handlers:

```
sub response_state_handler {
    my $peer_ip = $_[ARG0]->{'Address-IP'};
    if (defined($peer_ip)) {
        debug_print("SIP context found: Peer IP address");
        debug_print("is $peer_ip\n");
        $_[HEAP]->{'SIP-Peer-IP'}=$peer_ip;
    }
} # response_state_handler
```

After the SIP data response comes back, the four database queries should return responses. Responses to the queries look like this:

```
DBGetResponse: Success
Family: tz
Key: 300-TIMESKEW
Val: -8
```

The callback handler is triggered whenever there is a `DBGetResponse: Success` event from the manager, with an argument of a hash that has each of the lines in the packet. Our interest is in the key and value lines, which can be retrieved from the arguments passed to the state handler. As with the previous handler, responses are stored in the POE task heap to make it available to other handlers:

```
sub db_response_state_handler {
    my $family = $_[ARG0]->{'Family'};
    my $key = $_[ARG0]->{'Key'};
    my $value = $_[ARG0]->{'Val'};

    if (defined($family)) {
        debug_print("Key $key in DB family $family");
        debug_print("has value = $value\n");
        # Store in heap
        $_[HEAP]->{$key} = $value;
    }
} # db_response_state_handler
```

Making the Call

Every registration event triggers a “call” event to happen after a delay. The delay is used to collect information used to determine whether to initiate a call. The setup telephone call should be

triggered only if the time-zone setup has expired or the SIP device has changed its IP address and the record is no longer valid.

Because the call state handler is placed in the queue for execution by the registration handler, it does have one argument, the extension number of the call in question. The extension can be retrieved as `$_[ARG0]`. All the data we have added to the heap by processing the database responses and SIP data request is also readily available:

```
sub call_state_handler {  
  
    # Get extension out of arguments to function  
    my $exten = $_[ARG0];  
  
    my $hp = $_[HEAP];  
    # Variables we use to determine if the call is required  
    my $skew = $hp->{$exten.'-TIMESKEW'};  
    my $skew_addr = $hp->{$exten.'-TIMESKEW_ADDR'};  
    my $skew_start = $hp->{$exten.'-TIMESKEW_START'};  
    my $skew_end = $hp->{$exten.'-TIMESKEW_END'};  
    my $sip_peer_ip = $hp->{'SIP-Peer-IP'};  
    my $now = time();
```

To determine whether the call is required, the handler compares the current time with the expiration of the time-zone offset record and the IP address of the SIP device against the IP address stored in the time-zone offset record. If the IP addresses match and the offset has not expired, no call is required. Otherwise, a call is needed and made with the `makeTZSetupCall` function:

```
if ($now > $skew_end) {  
    debug_print("Make call - offset has expired.\n");  
    makeTZSetupCall($_[HEAP], $exten);  
} elsif (!$skew_addr eq $sip_peer_ip) {  
    debug_print("Make call - SIP IP addr changed\n");  
    makeTZSetupCall($_[HEAP], $exten);  
} else {  
    debug_print("No call -- record OK & same IP\n");  
}
```

As a final step, the handler needs to remove the variables placed on the heap. The heap is used only to pass variables between state handlers, and the variables are not needed once that function is complete. Each of the variables can be undefined with the `undef` function:

```
# Need to clean up heap  
undef $_[HEAP]->{$exten.'-TIMESKEW'};  
undef $_[HEAP]->{$exten.'-TIMESKEW_ADDR'};  
undef $_[HEAP]->{$exten.'-TIMESKEW_START'};  
undef $_[HEAP]->{$exten.'-TIMESKEW_END'};  
undef $_[HEAP]->{'SIP-Peer-IP'};  
  
} # call_state_handler
```

Making the setup call uses the Asterisk Manager's `Originate` command, but it is protected by one final check. I've defined a set of extensions as the remote channel list. Only extensions on the remote channel list will have time-zone setup calls made to them. Initially, the list consists

of my softphone and an analog telephone adapter, but I may need to add more in the future. Before originating the call, I ensure that the number is on a remote channel list, which is defined in the global array `REMOTE_CHANNEL_LIST`. The `Originate` command can take several arguments as well. The extension, priority and context must refer to where the setup menu is defined. In my case, these values are extension *89 (for *-T-Z), priority 1 and the context from-internal. I also can supply the caller-ID text of "Time Zone Setup" to the phone I am calling:

```
sub makeTZSetupCall {  
    my $heap = $_[0];  
    my $exten = $_[1];  
    my $callOK = 0;  
  
    # Check that extension to call is a remote channel  
    foreach $number (@REMOTE_CHANNEL_LIST) {  
        if ($number == $exten) {  
            $callOK = 1;  
        }  
    }  
  
    if ($callOK) {  
        $heap->{server}->put({  
            'Action' => 'Originate',  
            'Channel' => 'SIP/'.$exten,  
            'Context' => $TZ_CONTEXT,  
            'Exten' => $TZ_EXTEN,  
            'Priority' => $TZ_PRIORITY,  
            'Callerid' => $CALLERID,  
        });  
    }  
} # makeTZSetupCall
```

If the `Originate` command is triggered, the newly registered telephone rings, and I go through the voice menu described in last month's article. ■

Matthew Gast is the author of the leading technical book on wireless LANs, *802.11 Wireless Networks: The Definitive Guide* (O'Reilly Media). He can be reached at matthew.gast@gmail.com, but only when he is close to sea level.

Resources

Full Source Listing of Perl Script: ftp.linuxjournal.com/pub/lj/listings/issue156/9284.tgz

Information on the Asterisk Manager API: www.voip-info.org/wiki-Asterisk+manager+API

Perl POE Framework: poe.perl.org

Perl Asterisk Manager Component: search.cpan.org/~xantus/POE-Component-Client-Asterisk-Manager-0.06/Manager.pm

Use Inkscape and XSLT to Create Cross-Platform Reports and Forms

A way to create platform-independent dynamic forms and reports. CHAD FILES

I work for a health-care company developing application software. My colleagues and I are responsible for writing software to process health-care claims, manage work flow and make the company as efficient as possible. We recently decided to replace a piece of third-party software that took health-care claim data and overlaid it on standard HIPAA (Health Insurance Portability and Accountability Act) claim forms. The software would take the data and transpose it into PDF files that we stored on a large file server. Each PDF contained one claim on its proper form. We made the decision to replace the software because we needed something more agile. We wanted something that would create the claim image dynamically and not consume space on our servers.

Health-care claims are very intricate (Figure 1). Many boxes and

The image shows a standard HCFA 1500 Health Insurance Claim Form. It is a complex document with many sections and fields. Key sections include:

- Section 1:** Insured's name, address, and contact information.
- Section 2:** Insured's date of birth, sex, and marital status.
- Section 3:** Insured's policy or group number, date of birth, and employer's name.
- Section 4:** Insurance plan name and date of service.
- Section 5:** Referring physician's name, address, and specialty.
- Section 6:** Dates of service, procedure codes, and charges.
- Section 7:** Signature of patient or authorized person.
- Section 8:** Signature of provider or facility.

 The form is divided into several vertical columns and has a header with instructions like 'PLEASE DO NOT STAPLE IN THIS AREA' and 'APPROVED ONE YEAR ISSUE'. It also includes a footer with the form number 'FORM HCFA-1500 (02-99)'.

Figure 1. Health Insurance HCFA 1500 Claim Form

boilerplate text have to be drawn. The conventional way to do this with a software application is to draw a series of lines using coordinates and lengths, and then lay the static and dynamic content on top of the newly drawn lines. The process of programming an application like this is long and tedious, not to mention error-prone. We wanted something that was easier to create and maintain. Our requirements were as follows:

- We must be able to print high-quality versions of the claims.
- Claims must be accessible from a Web browser.
- The solution has to be programming language-independent. We use Python, PHP, Perl and Java. The images need to be created using any of these languages.
- We must be able to convert the claim data and form into several different file formats, specifically PDF and PNG.
- The entire solution must be platform-independent.

After reviewing the requirements, we looked at several different open- and closed-source options. None of them met all of our requirements, so we turned to creating our own solution. We tried scanning a blank claim form and using ImageMagick to put the claim data on it. This almost gave us what we wanted. The problem was that it was going to be tedious and redundant to create the solution in all of the required languages. Next, we turned to FOP (Formatting Objects Processor). This solution was closer to what we wanted. However, it would take too long create the claim forms. Plus, the solution was not really language-independent either (FOP is a Java library). We could have written wrappers for the FOP command-line interface, but we were convinced that there was still a better solution.

While exploring the FOP solution, we had the idea of using Scalable Vector Graphics (SVGs). Basically, we would take an SVG image of the claim form and make it into an XSLT (eXtensible Stylesheet Language Transformation), because the SVG format is a special XML format. Then, we would pull the claim data from our database and convert it into an XML string. Using any of our languages, we could then take the XSLT and the XML and create an SVG image of the claim. This solution met all of our requirements. It was language- and platform-independent. We could print the SVG images and embed them into Web pages. Furthermore, SVG images can be converted into different file formats easily. Another nice feature of this

solution is the small file size of the SVG images. If we wanted to archive the images, they would take a fraction of the space the old solution did. Because SVG images are text, not compressed binary, the files can be compressed and save even more space.

Creating the Master SVG

One of the things that made the SVG solution so appealing was how easy it would be to create and maintain the master SVG image of the form. To do this, we would use Inkscape. Inkscape is an SVG-authoring tool that works on Linux, Mac OS X, Windows and other UNIX-like operating systems. Other SVG-authoring tools are available, but we chose Inkscape because it is open, and it is in the package manager for most Linux distributions.

The first thing we did to create the master SVG was open Inkscape and create a new US Letter size document. To keep things organized, we created four layers in the new document: scan, overlay, boilerplate and dynamic text. Using the scan layer, we imported a scan of a claim. Doing this allowed us to line up everything on the Inkscape stage without having to measure anything. After importing the image, we locked the layer so that it could not be modified accidentally. Actually, after we were finished with each layer on the SVG, we would lock it to ensure it was not tampered with.

Next, we used the overlay layer to trace all the lines and boxes from the original claim that we imported. This step was a little tricky. When the image we scanned was originally created, the lines were not spaced evenly for one reason or another. We decided to line up things correctly on our version. Fortunately, Inkscape has tools to do this automatically. By selecting all of the objects that needed to be spaced out (Shift-left-click) and using the Align and Distribute dialog

(Object→Align and Distribute in the menu), Inkscape fixed the spacing issues. When finished, we had something that looked like Figure 2.

After drawing all of the lines, it was time to add all the boilerplate text. For this, we used the aptly named boilerplate layer. Before we got started, we decided to remove the scan layer completely, because we no longer needed it. To align the text properly, we used the Guides in Inkscape. Guides are exactly what the name suggests—guide lines that exist only inside of Inkscape for the purpose of aligning objects. To use a guide line, simply click the top or left-hand margin and drag the line into place. To get the most out of the guide lines, we enabled the Snap points to guides feature (File→Document Preferences→Guides). Doing this allowed us to place all of the text exactly in alignment. Figure 3 shows what the SVG looked like after this step.

Finally, we switched to the dynamic text layer and added placeholders where the claim data would be located. Again, we used the guides to align everything. For the text placeholders, we used a single \$ for each block of text. Then, to make life easier down the road, we renamed each of the dynamic text objects to something relevant. We did this by left-clicking on the object and going to Object→Object Properties in the menu. Figure 4 shows the final master SVG with the guide lines.

Creating the master SVG took about four full hours of work. I would venture to guess that it would have taken several days to do this programmatically.

Converting the SVG to an XSLT

Once we had the master SVG finished, it was time to convert it into an XSLT. Because SVG images are just XML files, we added all of the

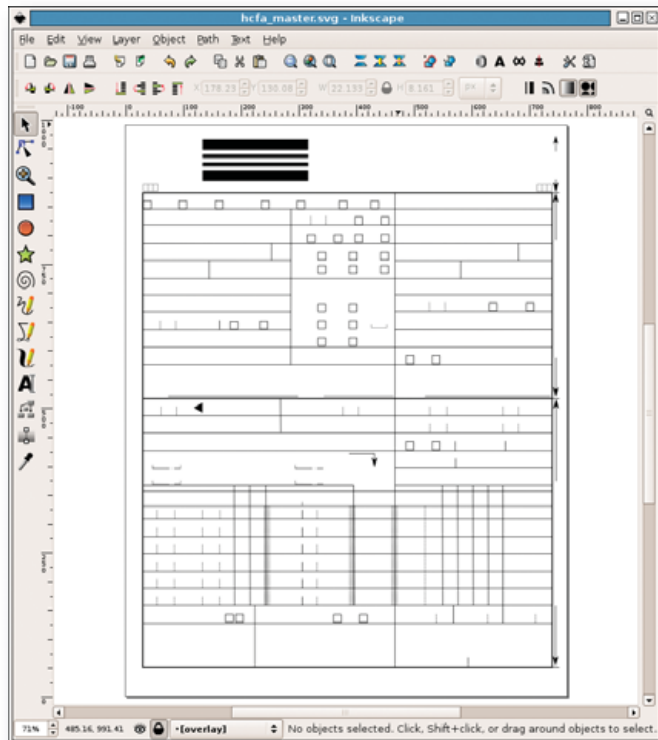


Figure 2. A Trace of the Lines and Boxes from the Claim Form

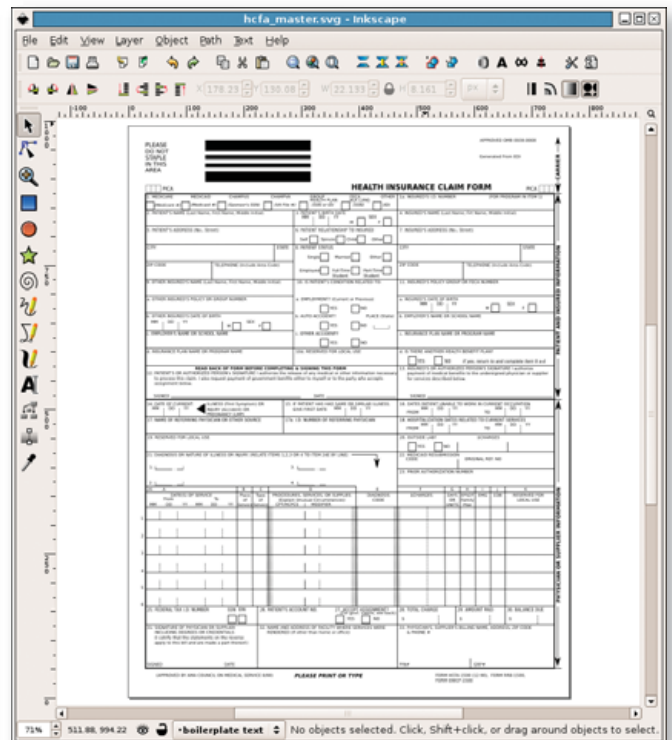


Figure 3. The Blank Claim Form Completed

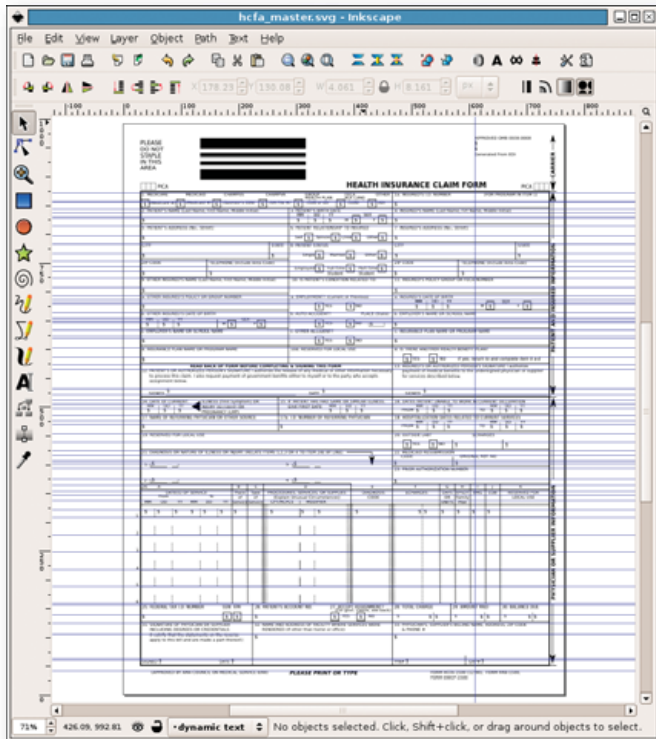


Figure 4. Final Master with Dynamic Text Layer

XSLT markup with a text editor. Converting the SVG was a rather simple matter. To make it a true XSLT, only a few lines are required in the header. Listing 1 shows a few lines of the SVG before we modified it. Listing 2 shows the same set of lines with the XSLT markup.

As you can see, there are four new lines. The first new line declares this file an XSLT. The second new line contains an XPath (XML Path Language) expression that matches the root element in our claim data XML. This line tells the XML transform engine where to start reading the XML to do the conversion. The last two new lines simply close the open xsl tags.

At this point, the XSLT can be used in conjunction with our claim data XML to produce an SVG. However, the resulting SVG would look just like the SVG did before we modified it. To make it actually show the claim data, we had to go into the XSLT and add all of the XPath expressions to populate the SVG. Because we divided the SVG objects into layers, we had to modify only the dynamic text layer. In the SVG XML, the dynamic text layer is nothing more than a series of text tags. Listing 3 shows the text tag for the Patient's City box on our claim form.

When the XSLT is applied to the claim data XML, the value of `/claim/patient/address/city` will be substituted here. We went through the entire XSLT and added the appropriate XPath expressions where they belonged. In special cases, we also added XPath conditional logic and formatting rules.

The Claim Data XML

As mentioned previously, all of our claim data was in a database—a Postgres database to be more specific. As we wanted a solution that was not language-specific, we had to devise a way to get the claim

Listing 1. A Few Lines of the SVG before Modifying

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!-- Created with Inkscape (http://www.inkscape.org/) -->
<svg
...
</svg>
```

Listing 2. The Same Set of Lines with the XSLT Markup

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<!-- Created with Inkscape (http://www.inkscape.org/) -->
<xsl:stylesheet version="1.0"
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/claim">
<svg
...
</svg>
</xsl:template>
</xsl:stylesheet>
```

Listing 3. Text Tag for the Patient's City Box on the Claim Form

```
<text
  xml:space="preserve"
  style="..."
  x="33.237278"
  y="231.77995"
  id="textPatientCity"
  sodipodi:linespacing="125.00000%"
  inkscape:label="#text7272">
  <tspan
    sodipodi:role="line"
    id="tspan7274"
    x="33.237278"
    y="231.77995"><xsl:value-of
    select="patient/address/city"/></tspan></text>
```

Listing 4. Query with ID of the Claim

```
SELECT xe2_claim('09152006A5226');
```

data out of the database and into an XML format without depending on a specific programming language. One of my fellow developers had the idea to write a series of PL/pgSQL functions to return a single XML string that contained the XML data. His solution was brilliant and fit the bill perfectly. All we needed to do to get the claim data was run one small query with the ID of the claim (Listing 4). The result was well-formatted XML that we used to make claim images.

Listing 5. A Portion of the PHP Script That Transforms the Claim XML into an SVG and Displays It in a Browser

```
// import the SVG XSLT
$xml = new XSLTProcessor();
$xml->importStyleSheet(DOMDocument::load("svg_xslt.xml"));

// load the claim data XML
// $claim is the database result from Listing 4
$doc = new DOMDocument();
$doc->loadXML($claim);

// tell the browser this is an SVG document
header("Content-Type: image/svg+xml");

// print the SVG to the browser
echo $xml->transformToXML($doc);
```

Displaying the Final SVG in a Browser

At first, the primary point of creating this solution was to display claims in our Web interface. All of our Web applications are written in PHP5 and run in an Apache/mod_php environment. To do the XSLT transformation, we used the XSL functions in PHP. This set of functions comes as an extension to PHP. The extension is a front end to the libxslt C library.

The XSLT extension makes doing the transition easy. Listing 5 shows a portion of a PHP script that transforms the claim XML into an SVG and displays it in the browser.

Listing 5 is a simplified version of our solution. In our solution, there is the possibility of having multiple pages for a single claim. To fix this, we had to do multiple transformations, one for each page. To get the multiple-page claims to display in the same browser window, we had to embed them. This can be done using the embed and object HTML tags. Note that there are several issues with browser compatibility when using these tags. To solve the compatibility issues, we wrote a script that checks the user's browser and decides which tag to use. Then, we set the target object data/embedded source to a script similar to the one in Listing 5. This allowed the Web browser to display multiple SVG images in the same window.

Other considerations must be made when using SVG images in a Web browser environment. Internet Explorer does not have native support for SVG images. The user is forced to use a third-party plugin to display the images. Adobe provides one of these for free. Mozilla

Firefox has built-in support for SVG images starting with version 1.5. However, Firefox does not support several aspects of SVG images, such as scaling and grouped objects. Fortunately for us, all of our users use an up-to-date version of Firefox.

That is all there is to it. Figure 5 shows a claim image with all of the data filled in.

Printing and Archiving the SVG Images

Once we finished the Web end of our solution, we turned our sights toward the rest of our integration. This meant we had to print the SVG images and find a way to archive them. Some clients request that we send them copies of the claims printed and/or electronically. Because all of our back-end software is written in Python, it also meant we had to do the XML transformation in a different language. To do all of the XML work, we used the 4Suite XML API.

To print the images, we again turned to Inkscape, because our PostScript printer drivers would not print the SVG images. Inkscape has a handful of command-line options that tell Inkscape to run in command-line mode, thus suppressing the graphical interface. The one we used to print is the -p option. This, combined with the lpr command, allowed us to print our images without any user interac-

DEDICATED SERVERS
Total Linux Support

Trustix suse

carinet

STARTING AT **60\$** 1GB DDR400 RAM — 160GB SATA2 HDD
 INTEL BOARDS & CPUS
 100MBPS DEDICATED CISCO PORT
 1300GB THROUGHPUT INCLUDED

CARI.NET/LJ
 888.221.5902

Figure 5. Claim Form with Sample Data

tion. Listing 6 shows how we did the same transform we did in Listing 5, except now in Python. The example also shows how we called Inkscape to print our claim images.

Earlier, I mentioned we often have multiple pages per claim. When printing, this was not an issue; we simply would send each page to the printer as a separate job. When it came to archiving, we had to do something different. As with the Web interface, we had to group the pages, this time into a file, not a Web browser. When archiving, we had to store the files in PDF format, because that is what our clients wanted. To get the images into a PDF and combine the multiple page claims, we used Inkscape and Ghostscript.

As with printing, Inkscape has an option to export a file into PostScript format. Instead of using `-p`, we use `-P` and pass Inkscape the desired output filename. After all of the pages of a claim have been written to files, we use the following Ghostscript command to put the pages into a single PDF and archive them:

```
gs -dBATCH -dNOPAUSE -q -sDEVICE=pdfwrite -sOutputFile=out.pdf
/tmp/foo1.ps
/tmp/foo2.ps
```

Maintaining the Forms

Shortly after we finished the project, we were faced with making two rounds of changes to the layout of the form. The first round of changes dealt with the positioning of text objects. The second

Listing 6. Same Transform as Shown in Listing 5. Except Using Python

```
from Ft.Xml.Xslt import Processor
from Ft.Xml import InputSource
from Ft.Xml.Domlette import NonvalidatingReader

// load the claim data XML
// claim is the database result from Listing 4
doc = NonvalidatingReader.parseString(claim,
"http://spam.com/doc.xml")

// load and process the XSLT
xsl = InputSource.DefaultFactory.fromUri("file://svg_xslt.xsl")
processor = Processor.Processor()
processor.appendStylesheet(xsl)

// do the transformation
result = processor.runNode(doc, "http://spam.com/doc.xml")

// write the SVG to a file
f = open("/tmp/"+ claim + ".svg", "w")
f.write(result)
f.close()

// print the image on the default printer
os.system("inkscape /tmp/"+ claim + ".svg -p | lpr")
```

round was far more extensive—we had to draw a series of new boxes on the form to accommodate a new identification system. Because we could not open the modified SVG in Inkscape, we had to make our changes to the master SVG and then apply them manually to the XSLT version.

At first, we thought making the changes would be hard and tedious, but it turned out that the process was simple. For the first round, we simply made the changes in the master using Inkscape, careful to keep a note of the objects we changed. Then, using a text editor, we replaced the old portions of XML with the new ones in the XSLT. Because the second batch of changes was additions only, we decided simply to make another layer in the master to which to add the boxes. When we finished adding the new boxes, we simply copied the new layer into the XSLT using a text editor.

Conclusion

From start to finish, our project took a little more than a month to design, build, test and publish. Our solution has made all of our applications more agile and effective. We also have saved terabytes' worth of storage space on our servers.

Currently, the SVG adaptation rate is rather slow. We are looking forward to seeing what other tools will be built that utilize the versatile SVG file format. ■

Chad Files is a software developer who resides in Conway, Arkansas. He is an avid hiker and longtime Linux user. He welcomes your comments at cpfiles@gmail.com.

Extract and Parse ODF Files with Python

Use Python to demystify Open Document Format files. KAMRAN HUSAIN

The **Open Document Format** (ODF) Alliance is designed for sharing information between different word processing applications. This article highlights the basic structure of ODF files, some internals of the underlying XML files and shows how to use Python to read the contents to perform a simple search for keywords. The code also can be the basis for more-advanced operations. In the spirit of openness, we use open-source software to read the ODF files, which in this case are Python and the OpenOffice.org package.

If you are running a recent version of Linux or OS X, you already should have Python and OpenOffice.org installed on your machine. If you need the latest versions, Python is available for free from www.python.org for both the Windows and Linux platforms. The OpenOffice.org package also is available for free from www.openoffice.org. Installing OpenOffice.org on an XP desktop is relatively painless. Download the packages from their respective sites and run the installer. Once installed, simply run the application from the desktop with a click on the installed icons.

Tip

Most folks do have Microsoft Office installed. If that's the case, the solution is to use a plugin for Microsoft Word (sourceforge.net/projects/odf-converter). You can install both the OpenOffice.org and Microsoft packages on the same machine without causing any conflicts.

Please read the Bugs section on the SourceForge site for any incompatibilities before you install the plugin. I used the OpenOffice.org suite to save the files for this article, because it was easier.

Once you have confirmed that you have the prerequisites, you can create an ODF file. Open up the Writer, type some text in a document and save it. You can read in a file and save it as an .odt file.

A quick look at extensions in the Save dialog reveals a lot. An ODF file can have many extensions, which provide a clue as to the type of information stored in it and the application that stored it. See Table 1.

So, what's in an ODF file? An ODF file is basically a zipped archive with several XML files. The actual files and directories in a file will vary depending on the type of information and the system on which the document was created.

The first step in picking out the names of the files in an ODF file requires unzipping the file itself. Fortunately, Python has built-in support for dealing with this endeavor with the `zipfile` module.

Table 1. ODF File Types and Their Extensions

Document Format	File Extension
OpenDocument Text	*.odt
OpenDocument Text Template	*.ott
OpenDocument Master Document	*.odm
HTML Document	*.html
HTML Document Template	*.oth
OpenDocument Spreadsheet	*.ods
OpenDocument Spreadsheet Template	*.ots
OpenDocument Drawing	*.odg
OpenDocument Drawing Template	*.otg
OpenDocument Presentation	*.odp
OpenDocument Presentation Template	*.otp
OpenDocument Formula	*.odf
OpenDocument Database	*.odb

Type `python` on the command line to run an interactive shell. Running a shell allows you to examine the contents of objects returned from the modules. Because you'll probably be doing this only once per type of data, there is really no need to write and execute a script at this time. If you want to preserve the work for future use, it's better to write a script in a text editor or use the IDLE editor that comes with Python and save the script. See Listing 1 on how to show the member functions in a class or module.

The `infolist()` command from the Python documentation returns a list the objects of a zipped archive. Run the `dir()` command on the first object from this list to get more information stored for each zipped file (Listing 2). This nice feature of looking at members in an object is called introspection.

An iteration on the list of objects displays relevant information

Listing 1. Showing the Member Functions in a Class or Module

```
Python 2.4.1 (#65, Mar 30 2005, 09:13:57)
[MSC v.1310 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()"
for more information.

>>> import zipfile
>>> myfile = zipfile.ZipFile
↳('E:/articles/odf/theArticle.odt')
>>> dir(myfile)
['NameToInfo', '_GetContents', '_RealGetContents',
'__del__', '__doc__', '__init__', '__module__',
'_filePassed', '_writecheck', 'close', 'comment',
'compression', 'debug', 'filelist', 'filename', 'fp',
'getinfo', 'infolist', 'mode', 'namelist',
'printdir', 'read', 'start_dir', 'testzip', 'write',
'writestr']
>>>
>>>
>>> listoffiles = myfile.infolist()
>>> dir(listoffiles[0])
['CRC', 'FileHeader', '__doc__', '__init__',
'__module__', 'comment', 'compress_size',
'compress_type', 'create_system', 'create_version',
'date_time', 'external_attr', 'extra',
'extract_version', 'file_offset', 'file_size',
'filename', 'flag_bits', 'header_offset',
'internal_attr', 'orig_filename', 'reserved',
'volume']
>>>
```

Listing 2. List the Files in the ODT Archive

```
import sys, zipfile
myfile = zipfile.ZipFile(sys.argv[1])
listoffiles = myfile.infolist()
for s in listoffiles: print s.orig_filename
```

about each file, such as when it was created, its extracted size, its compressed size and so on. It's better at this point to write a Python script and run it rather than work at the command line of an interactive Python shell. This way, you can preserve the script for later use. So, open up a text editor and type in the script.

The import statement allows you to use the sys module for getting a command-line argument of the file, and the zipfile module loads in the functionality for reading and unzipping files. As you saw from the Python shell, the infolist() method on the zipfile archive lists the files in it. So iterating over the items from the infolist() and then calling an orig_filename member function gives you a list of all files in the archive.

For more detailed information, try something like this:

```
print s.orig_filename, s.date_time, s.filename,
↳s.file_size, s.compress_size
```

You will receive more information about the file, quite similar to this:

```
mimetype (2006, 9, 9, 7, 50, 10) mimetype 39 39
Configurations2/statusbar/ (2006, 9, 9, 7, 50, 10)
Configurations2/statusbar/ 0 0
Configurations2/accelerator/current.xml
↳(2006, 9, 9, 7, 50, 10)
Configurations2/accelerator/current.xml 0 2
Configurations2/floater/ (2006, 9, 9, 7, 50, 10)
Configurations2/floater/ 0 0
...SNIPPED FOR BREVITY...
```

A typical ODF text file (with the .odt extension) will have some of the following files when unzipped. Here's the output:

```
mimetype
Configurations2/statusbar/
Configurations2/accelerator/current.xml
Configurations2/floater/
Configurations2/popupmenu/
Configurations2/progressbar/
Configurations2/menuubar/
Configurations2/toolbar/
Configurations2/images/Bitmaps/
layout-cache
content.xml
styles.xml
meta.xml
Thumbnails/thumbnail.png
settings.xml
META-INF/manifest.xml
```

The most important file in the archive is the content.xml file, because it contains the data for the document itself. I discuss this file here; however, for detailed information on each tag and so on, take a look at the specification in the 2,000+-page PDF file from the OASIS Web site (see Resources).

Basically, the content.xml file looks like a DHTML file with tags for all the contents. The tag I was concerned with most for my search operation was the <text:p> tag and its closing tag </text:p>, which wraps paragraphs in a document. I'll show you how to get this tag from a content file later in this article.

Other files of interest in the archive are the META-INF/manifest.xml, mimetype, meta.xml and styles.xml. Other files simply contain data for configurations for the word processors reading and displaying the contents file.

The manifest is simply an XML file with a listing of all the files in the zipped archive. The mimetype file is a single line containing the mimetype of the content file. The meta.xml contains information about the author, creation date and so on. The styles file contains all the formatting styles for displaying the file.

You can extract any of these files from the ODF file with the read() method on the zip object to get it as a very long string. Once read, you can modify, view and write the whole string to

Listing 3. Extracting Files for the ODT Archive

```
import sys, zipfile
if len(sys.argv) < 2:
    print "Usage: extract odf-filename outputfilename"
    sys.exit(0)

myfile = zipfile.ZipFile(sys.argv[1])
listoffiles = myfile.infolist()
for s in listoffiles:
    if s.orig_filename == 'META-INF/manifest.xml':
        fd = open(sys.argv[2], 'w')
        bh = myfile.read(s.orig_filename)
        fd.write(bh)
        fd.close()
```

disk as an independent file. Listing 3 shows how to extract the manifest.xml file.

For more than one file, you can use a list instead of the if clause:

```
if s.orig_filename in ['content.xml', 'styles.xml']:
```

This way, you can extract whatever files you need to look at simply by reading in their contents and either manipulating them or writing them off to disk.

The contents of an XML file are best suited for manipulation as a tree structure. Use the XML parsing capabilities in Python to get a tree of all the nodes within an XML file. Once you have the tree in a content file, you easily can get to the <text:p> nodes. You don't really have to extract the file to disk, because you also can run an XML parser on the string just as well as reading from a file.

There are two types of XML parsers, SAX and DOM. The SAX parser is faster but less memory-intensive, because it reads and parses an input file one tag at a time. You have only one tag at a time to work with and must track data yourself. In contrast, the DOM parser reads the entire file into memory and therefore provides better options for navigating and manipulating the XML nodes.

Let's look at examples of using both SAX and DOM, so you can see which one suits your purpose. The SAX example shows how to extract unique node names from an XML file. The DOM example illustrates how to read values from within specific nodes once the file has been read completely into memory.

First, let's use the SAX parser to see what nodes are available in the content.xml file. The code simply prints the name of each type node found in the file. Note that for different types of files you may get different node names (see Listing 4).

A SAX program requires a class derived from ContentHandler and overriding functions to handle the start, middle and end of each node. The tagHandler class shown in Listing 4 is used primarily to track the start of each node and ignores the contents. Use the names found in the listing as keys in a dictionary. Then, use the keys() method to list the names into a list that you also can sort(). I use this technique quite often to get a sorting of unique members quickly, because the hashing functions in Python dictio-

Listing 4. List uniq Tag Numbers

```
#
# This program will list out the uniq tag
# names in a XML document.
# Author: Kamran Husain
#
import sys
from xml.sax import parse, ContentHandler

class tagHandler(ContentHandler):
    def __init__(self, tagName = None):
        self.tag = tagName
        self.uniq = {}

    def startElement(self, name, attr):
        if self.tag == None:
            self.uniq[name] = 1;
        elif self.tag == name:
            self.uniq[name] = name
        # ignore attributes for now

    def getNames(self):
        return self.uniq.keys()

if __name__ == '__main__':
    myTagHandler = tagHandler()
    parse(sys.argv[1], myTagHandler)
    myNames = [str(x) for x in myTagHandler.getNames()]
    myNames.sort()
    for x in myNames: print x
```

naries are very fast. Here's the output from the program:

```
office:automatic-styles
office:body
office:document-content
office:font-face-decl
office:forms
office:scripts
office:text
style:font-face
style:list-level-properties
style:paragraph-properties
style:style
style:text-properties
text:a
text:line-break
text:list
text:list-item
text:list-level-style-bullet
text:list-style
text:p
text:s
text:sequence-decl
```

```
text:sequence-decls
text:span
```

I sorted the list of keys and printed only the types of tags found. You will have many tags, and the order of the listed tags is not the way they are found in the XML file. The tag you most likely will be concerned with is `<text:p>`, which has the text in each paragraph. In this example, I want to search for keywords in the text for one or more paragraphs in a document.

Although I can use the SAX version of the program to search for the text, I use the DOM libraries, because the code will be a little easier to write (and subsequently, easier to maintain), and I promised an example of this earlier.

The `xml.dom` and `xml.dom.minidom` packages in Python allow for reading in XML files as DOM trees. Both packages come with a standard Python installation. Use the `minidom` package as it has a smaller footprint and is easier to use than the `dom` package. The `minidom` package is sufficient for almost all my XML work, and I have never really needed the heavyweight `xml.dom` package. See Resources for more information.

Use the `minidom` packages to read the elements of an XML file into a tree-like structure. The nodes of the tree are objects based on the `Node` class in Python. The output from Listing 4 provides the names of nodes.

Up to this point, you have been using simple programs to list and extract data from the archive. Now, the next example runs a search operation on the file you've just read in. Look at the program in Listing 5.

The program is designed to work as a class that reads and searches for text in an ODF file. Declaring a class for the ODF reader helps in organizing the code for searching text within a node. The `showManifest()` member function simply tells me what files exist in the ODF file. In this particular program, I collect all the text as a list of paragraphs, and then I search for the keywords passed in from the command line. If the searched word matches, the paragraph is printed out.

The text found in each `<text:p>` is Unicode text. You have to convert this to normal text in order to print correctly and/or use in a widget. The `encode()` command translates the Unicode to a printable string.

Unicode provides a unique number for every character, regardless of the platform, program and language being used. The ability to work seamlessly with the same text across multiple platforms is a great feature for Unicode-enabled applications. Such features do come with a price for some legacy operations. Each Unicode character can have flags as bits set for special printing and so on, which causes a normal print statement to interpret each character as a number instead of text. In Python, the `encode()` member function of a Unicode string returns a printable version of the string. Here is an example code snippet for that:

```
def findIt(self,name):
    for s in self.text_in_paras:
        if name in s:
            print s.encode('utf-8')
```

The code in Listing 5 is not limited to an ODT file. You can

Listing 5. Reading and Parsing the ODF in Python

```
import os, sys
import zipfile
import xml.dom.minidom

class OdfReader:
    def __init__(self,filename):
        """
        Open an ODF file.
        """
        self.filename = filename
        self.m_odf = zipfile.ZipFile(filename)
        self.filelist = self.m_odf.infolist()

    def showManifest(self):
        """
        Just tell me what files exist in the ODF file.
        """
        for s in self.filelist:
            #print s.orig_filename, s.date_time,
            s.filename, s.file_size, s.compress_size
            print s.orig_filename

    def getContents(self):
        """
        Just read the paragraphs from an XML file.
        """
        ostr = self.m_odf.read('content.xml')
        doc = xml.dom.minidom.parseString(ostr)
        paras = doc.getElementsByTagName('text:p')
        print "I have ", len(paras), " paragraphs "
        self.text_in_paras = []
        for p in paras:
            for ch in p.childNodes:
                if ch.nodeType == ch.TEXT_NODE:
                    self.text_in_paras.append(ch.data)

    def findIt(self,name):
        for s in self.text_in_paras:
            if name in s:
                print s.encode('utf-8')
```

```
if __name__ == '__main__':
    """
    Pass in the name of the incoming file and the
    phrase as command line arguments. Use sys.argv[]
    """
    filename = sys.argv(0)
    phrase = sys.argv(1)
    if zipfile.is_zipfile(filename):
        myodf = OdfReader(filename) # Create object.
        myodf.showManifest()       # Tell me what files
        # we have here
        myodf.getContents()        # Get the raw
        # paragraph text.
        myodf.findIt(phrase)       # find the phrase ...
```


Security • Storage • Voice & Data
Virtualization • SOA • BI • Open Solutions • ITIL
Compliance • Network Infrastructure

modify the code presented here to work with spreadsheet files with an .ods file. Run the program in Listing 3 to get the contents.xml file out, and then run the second program (shown in Listing 4) to list the types of nodes. Below is a sample .ods file; note that this file also has paragraphs in addition to the table tags:

```
office:automatic-styles
office:body
office:document-content
office:font-face-decls
office:scripts
office:spreadsheet
style:font-face
style:style
style:table-column-properties
style:table-properties
style:table-row-properties
table:table
table:table-cell
table:table-column
table:table-row
text:p
```

Use the program in Listing 5 to extract and search text from paragraphs as before. A simple modification of changing the text:p to table:table-cell searches for text within cells instead of paragraphs.

To summarize, an ODF file is a zipped archive of several XML files. One of these files contains contents in known tags. Each type of ODF file can have different tags based on stored information. By using introspection and the XML parsing capabilities in Python, you can list the types of nodes in a file and read them into a tree structure. Once read, you can extract data only from those nodes in the tree that are of interest to you. ■

Kamran Husain has been working with software for 20 years. He can be contacted at khusain62@yahoo.com.

Resources

The OASIS Open Document Format specification and related information is available for download from www.oasis-open.org/committees/tc_home.php?wg_abbrev=office.

The documentation for tags in the content.xml file can be found at www.oasis-open.org/committees/documents.php?wg_abbrev=office.

Download Python from www.python.org.

Python in a Nutshell, Alex Martelli: O'Reilly, 2003.

Python and XML, Christopher A. Jones and Fred Drake, Jr.: O'Reilly, 2001.

XML Pocket Reference, 3rd edition, Simon St. Laurent and Michael Fitzgerald: O'Reilly, 2005.



LINUXWORLD
CONFERENCE & EXPO

it360°
CONFERENCE AND EXPO

IT360° is a powerful multi-sector experience.

Get the IT360° advantage – a single source of knowledge, realistic strategies, and tools to help you solve the critical burning issues today, paving the road for efficiency and productivity tomorrow. www.it360.ca.

Conference: April 30 – May 2, 2007

Trade Show: May 1 – May 2, 2007

**Metro Toronto Convention Centre
TORONTO, CANADA**

IT360° Conference and Expo is an ITWorld Expo event produced by ITWorld Canada the trusted name in Information Technology Resource Media.

www.it360.ca

NetworkWorld

SECURITYitWORLD

LINUXWORLD
CONFERENCE & EXPO

DATASTORAGEWORLD

PRODUCED BY:

itWorld Canada
IDG CANADA

Do Not Forget What People Fetch

The powerful iptables is intuitive enough even for lazy geeks to write their own rules.



Nick Petreley, Editor in Chief

Let's talk about protecting your network from what people can fetch. I'm going to take a twisty road to get there, so please stick with me.

I wrote a lengthy report published at www.theregister.co.uk/security/security_report_windows_vs_linux that you might want to peruse. Allow me a couple caveats. The report is old, dated October 2004. Don't e-mail me about the minor editing errors in the text that I never went back to fix. They don't affect the thrust of the report.

Anyway, I hope you read the whole report, but I reference it primarily to draw out a single point. The Summer 2004 Evans Data Linux Developers Survey states that 78% of Linux developers have never had a Linux machine compromised. The Evans survey didn't explore the nature of the very few Linux incidents that did occur. For example, some may have been victims of an Apache worm. Apache runs on both Linux and Windows, so those incidents would not be unique to Linux. More important, the same Apache worm that made the rounds many years ago could do much more damage on Windows than on Linux.

There's a reason for that. It doesn't take a rocket scientist to figure out that Windows is far more vulnerable to serious security

breaches than Linux. Vista will not improve things. Microsoft seems more interested in preventing users from bypassing DRM restrictions than preventing crackers from breaking into Windows.

You can probably protect both Windows and Linux users from outside attacks simply by putting them behind a NAT-enabled router. A Linux-based router or even a cheap OTS box will usually do. The problem is that many security breaches occur not due to incoming attacks, but because people fetch tainted Web pages, download infected software or fetch myriad other file types with embedded code that exploits all the holes in Windows that allow people to escalate privileges and compromise the entire Windows box. If everyone ran Linux, this might not be such a big problem. Not everyone runs Linux. My kids use Linux, but they also use Windows. This is not only a family issue though. You may be in charge of a network where employees run Windows.

Here's what I've done to protect my kids. First, I have taken great pains to limit their Web browsing to Firefox and their e-mail correspondence to Thunderbird. This avoids the common IE and Outlook exploits. Second, even though they access the Internet through a wireless NAT-enabled router, that router is connected to a Linux server that examines and filters what gets through before passing the traffic on to the Internet. I use tinyproxy with DansGuardian to block content. I use MIME filters to prevent potentially dangerous e-mail attachments from getting in. But, the coup de grâce is what I can do with iptables.

Before I continue, let me add a little perspective. I'd never consider myself competent enough to be a Linux kernel developer, but I have contributed a few lines of code to the kernel, and I have patched my own kernel to work around things like a stubborn ASUS motherboard that refused to shut down (the patch was too quick and dirty to be worthy of the kernel but it worked for me). I have also contributed code in various languages

to projects like Xoops, Lphoto and more. My point is that I'm not just an editor; I'm a programmer and an incurable geek.

I'm a lazy geek, however. I would much rather defer to a GUI program to generate firewall rules than write my own. But iptables has become so intuitive that even lazy geeks can write their own rules. All I had to grasp was how iptables processes traffic in terms of pre-routing, forwarding and postrouting. For example, the following rule is one of many that pre-routes various attempts to access the Web through DansGuardian. I have many such rules, and they're more specific (they specify the source IP address of various computers), but I'll list a simplified version of the obvious one:

```
iptables -t nat -A PREROUTING -i eth1 -p tcp  
-j REDIRECT --dport 80 --to-port 8080
```

Beyond a little knowledge of syntax, that's almost an English expression. It says to redirect the NAT traffic coming in through eth1 from port 80 to port 8080. If your kids or employees are clever enough to try to access the Web through an external proxy, simply redirect those ports to port 8080 too. You can use DansGuardian to block most other attempts to sneak around this protection.

Our article "Starting a Linux Firewall from Scratch" on page 78 should help you get started with iptables. There are far more comprehensive iptables tutorials on the Web, one being iptables-tutorial.frozentux.net/iptables-tutorial.html. But most of them explore the complexities of iptables that only a security expert would need. If your needs are light, like mine, don't be afraid to explore iptables and take a shot at writing your own rules. You'll find iptables much more friendly than you've ever imagined, and they're indispensable as part of your security framework. ■

Nicholas Petreley is Editor in Chief of *Linux Journal* and a former programmer, teacher, analyst and consultant who has been working with and writing about Linux for more than ten years.



Rackspace – Managed Hosting Backed by FANATICAL SUPPORT™

Fast servers, secure data centers and maximum bandwidth are all well and good. In fact, we invest a lot of money in them every year. But we believe hosting enterprise class web sites and web applications takes more than technology. It takes Fanatical Support.

Fanatical Support isn't a clever slogan, but the day to day reality our customers experience working with us. It's how we have reimagined customer service to bring unprecedented responsiveness and value to everything we do for our customers. It starts the first time you talk with us. And it never ends.

Contact us to see how Fanatical Support works for you.

1.888.571.8976 or visit www.rackspace.com

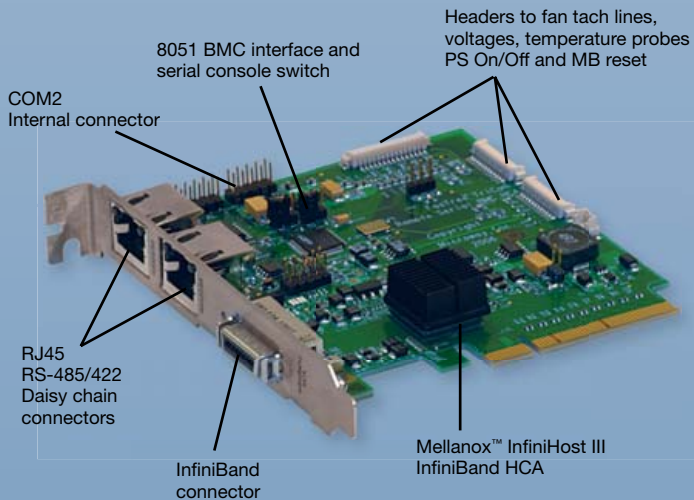


Affordable InfiniBand Solutions

4 Great Reasons to Call Microway NOW!

1 **TriCom™**

- DDR/SDR InfiniBand HCA
- "Switchless" serial console
- NodeWatch web enabled remote monitor and control



2 **FasTree™**

- DDR InfiniBand switches
- Low latency, modular design
- 24, 36 and 48 port building blocks



3 **InfiniScope™**

- Monitors ports on HCA's and switches
- Provides real time BW diagnostics
- Finds switch and cable faults
- Lane 15 interface
- Logs all IB errors



4 **ServaStor™**

- Extensible IB based storage building blocks
- Redundant and scalable
- Parallel file systems
- Open source software
- On-line capacity expansion
- RAID 0,1,1E, 3, 5, 6, 10, 50



Upgrade your current cluster, or let us design your next one using Microway InfiniBand Solutions.

To speak to an HPC expert call **508 746-7341** and ask for technical sales or email sales@microway.com
www.microway.com

Microway
Technology you can count onsm