OpenWrt | Pico Projector | D-Bus | Rockbox | Samurai | Boxee

LINUX JOURNAL™

Since 1994: The Original Magazine of the Linux Community

NOVEMBER 2010 | ISSUE 199 | www.linuxjournal.com

Security Testing with Samurai

Hacks from DEF CON

Control Your Linux System with a Smartphone

LARGE HADRON COLLIDER

Powered by Open Source

Automate and Personalize Your Desktop with **D-Bus**

Use Linux to **Find Your Phone**

Hacking a Wi-Fi-Enabled **Pico Projector**

PLUS: Intro to Rockbox

$5.99US $5.99CAN

11

0 09281 03102 4

JOIN THE MOBILE REVOLUTION WITH THE WO

WEBSITES

More and more people are browsing the web on their phones. At 1&1, you get the software you need to create websites that are optimized for mobile viewing.

- ✓ Layouts, designs and wizards enhanced for the latest smartphones, like iPhone and BlackBerry®
- ✓ Compatible across multiple platforms
- ✓ Valued at up to $479!

FREE SOFTWARE*

INCLUDED WITH 1&1 HOSTING PLANS. CHOOSE FROM:

NetObjects Fusion® 1&1 Edition is a website design application which creates sites that are optimized for mobile viewing. The 1&1 Edition was designed specifically to complement 1&1 web hosting packages and includes additional mobile templates as an extra bonus.

Adobe® Dreamweaver® CS4 is a sophisticated website design application for creating professional websites. Dreamweaver® includes the Adobe® Device Central module, enabling web designers to test their websites on mobile devices by emulating the latest smartphones.

EPA GREEN POWER PARTNER

THIS COMPANY USES CERTIFIED CLEAN ENERGY bef

RLD'S LARGEST WEB HOST. AT 1&1 INTERNET:

GO MOBILE

HURRY – OFFERS END 10/31/2010!

1&1® HOME PACKAGE

- 2 Domain Names Included (.com, .net, .org, .info or .biz.)
- 150 GB Web Space
- UNLIMITED Traffic
- 10 FTP Accounts
- 25 MySQL Databases
- Extensive Programming Language Support: Perl, Python, PHP4, PHP5, PHP6 (beta) with Zend® Framework
- NetObjects Fusion® 1&1 Edition

~~$6.99 per month~~

$3.49* per month for the first 6 months

1&1® BUSINESS PACKAGE

- 3 Domain Names Included (.com, .net, .org, .info or .biz.)
- 250 GB Web Space
- UNLIMITED Traffic
- 25 FTP Accounts
- 50 MySQL Databases
- Extensive Programming Language Support: Perl, Python, PHP4, PHP5, PHP6 (beta) with Zend® Framework
- NetObjects Fusion® 1&1 Edition or Adobe® Dreamweaver CS4

~~$9.99 per month~~

$4.99* per month for the first 6 months

1&1® DEVELOPER PACKAGE

- 5 Domain Names Included (.com, .net, .org, .info or .biz.)
- 300 GB Web Space
- UNLIMITED Traffic
- 50 FTP Accounts
- 100 MySQL Databases
- Extensive Programming Language Support: Perl, Python, PHP4, PHP5, PHP6 (beta) with Zend® Framework
- NetObjects Fusion® 1&1 Edition or Adobe® Dreamweaver CS4
- NEW: 1&1 Power Plus Performance Guarantee

~~$19.99 per month~~

$9.99* per month for the first 6 months

ALSO ON SALE:

.us domains $0.99/first year*
.com domains $7.99/first year*

Visit our website for a full list of special offers.

MEMBER OF united internet

1&1®

www.1and1.com

*Special offers valid through October 31, 2010. 12 month minimum contract term required for software offer. Setup fee and other terms and conditions may apply. Software available for download in the 1&1 Control Panel only. Domain offer valid first year only. After first year, standard pricing applies. Visit www.1and1.com for full promotional offer details. Program and pricing specifications and availability subject to change without notice. 1&1 and the 1&1 logo are trademarks of 1&1 Internet AG, all other trademarks are the property of their respective owners. © 2010 1&1 Internet, Inc. All rights reserved.

CONTENTS NOVEMBER 2010 Issue 199

FEATURES

ABERDEEN
SERVERS AND STORAGE

STORAGE WITHOUT BOUNDARIES

Imagine what you can achieve with Aberdeen.

Aberdeen's AberSAN Z-Series scalable storage platform brings the simplicity of network attached storage (NAS) to the SAN environment, by utilizing the innovative ZFS file system.

Featuring the high performance of the Intel Xeon processor 5600 series, the AberSAN Z-Series offers enterprise level benefits at entry level pricing, delivering the best bang for the buck.

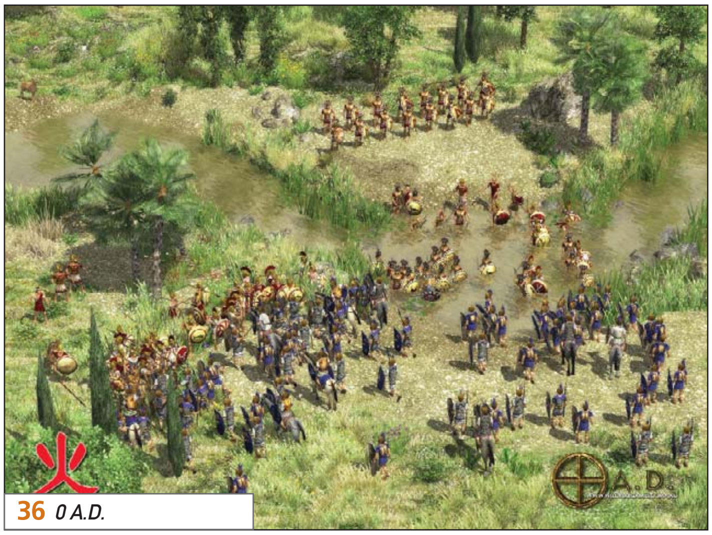
Who gives you the best bang for the buck?

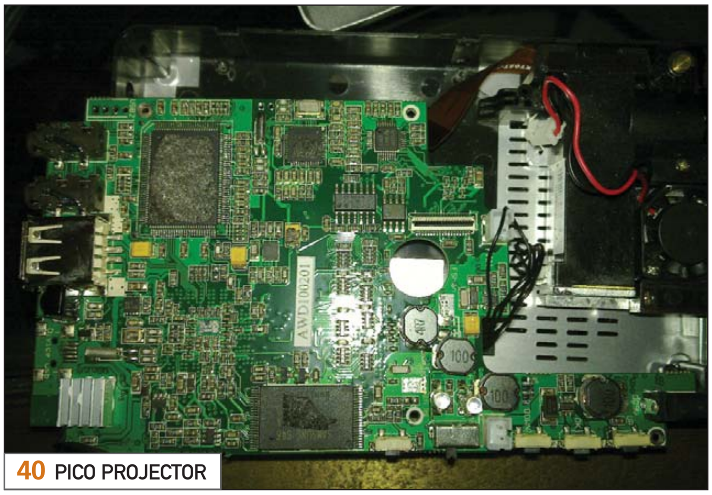
	NetApp FAS 2050	EMC NS120	Aberdeen AberSAN Z20
Deduplication	✓	✓	✓
Thin Provisioning	✓	✓	✓
VMware® Ready Certified	✓	✓	✓
Citrix® StorageLink™ Support	✓	✓	✓
Optional HA Clustering	✓	✓	✓
Optional Virtualized SAN	✓	✓	✓
Optional Synchronous Replication	✓	✓	✓
Optional Fibre Channel Target	✓	✓	✓
Unified Storage: NFS, CiFS, iSCSI	✓	✓	✓
Unlimited Snapshots	✗	✗	✓
Unlimited Array Size	✗	✗	✓
Hybrid Storage Pooling	✗	✗	✓
Integrated Search	✗	✗	✓
File System	WAFL 64-bit	UxFS 64-bit	ZFS 128-bit
RAID Level Support	4 and DP	5 and 6	5, 6 and Z
Maximum Storage Capacity	104TB	240TB	Over 1,000TB
Warranty	3 Years	3 Years	**5 Years**
Published Starting MSRP	Not Available	Not Available	**$8,995**

intel
Xeon®
inside™
Powerful.
Intelligent.

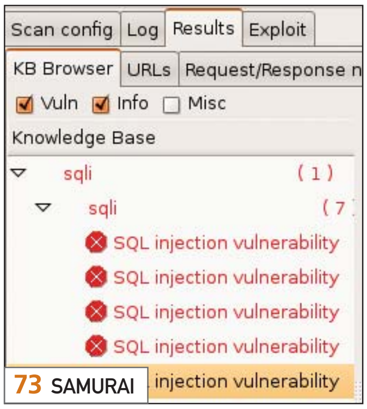
888-297-7409
www.aberdeeninc.com/lj036

CONTENTS NOVEMBER 2010 Issue 199


36 *0 A.D.*


40 PICO PROJECTOR

70 ROCKBOX



73 SAMURAI

USPS *LINUX JOURNAL* (ISSN 1075-3583) (USPS 12854) is published monthly by Belltown Media, Inc., 2121 Sage Road, Ste. 310, Houston, TX 77056 USA. Periodicals postage paid at Houston, Texas and at additional mailing offices. Cover price is $5.99 US. Subscription rate is $29.50/year in the United States, $39.50 in Canada and Mexico, $69.50 elsewhere. POSTMASTER: Please send address changes to *Linux Journal*, PO Box 16476, North Hollywood, CA 91615. Subscriptions start with the next issue. Canada Post: Publications Mail Agreement #41549519. Canada Returns to be sent to Pitney Bowes, P.O. Box 25542, London, ON N6C 6B2

They say work smarter, not harder. They must be using our processor.

intel
Sponsors of Tomorrow.™

The next generation of intelligent server processors
The Intel® Xeon® processor 5600 series automatically regulates power consumption to combine industry-leading energy efficiency with intelligent performance that adapts to your workload. Check out the new intelligent features of the Xeon 5600 at intel.com/itcenter.

intel Xeon® inside™

SEE WHAT INTELLIGENCE CAN DO

Servers from iXsystems feature the Intel® Xeon® processor 5600 series.

iX systems®

Enterprise Servers
for Open Source
www.iXsystems.com
800-820-BSDi

Intel is not responsible for and has not verified any statements or computer system product-specific claims contained herein.

The **Triton Twin-Blade Server** supports up to 120 dual processor servers with 240 Intel Xeon 5600 series processors per 42U rack, achieving an unmatched .35U per dual processor node.

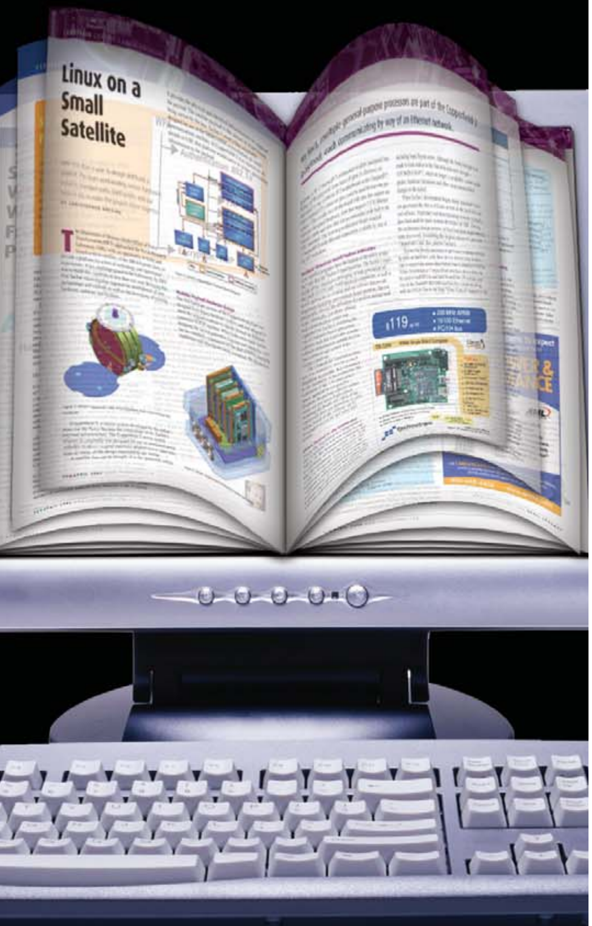
Cleverly disguised as any other 2U server, the **Gemini²** houses 4 highly efficient, extremely powerful RAID 5-capable servers.

Triton Twin-Blade

- Intel® Xeon® processor 5600 series
- 20 server compute nodes in 7U of rack space
- N+1 redundant, high-efficiency (94%) 2500W power supplies
- Remote access through SOL, KVM, KVM over IP technologies

Gemini²

- Intel® Xeon® processor 5600 series
- 4 nodes in a 2U form factor
- Up to 192GB DDR3 SDRAM per node
- 3 x 3.5" SAS/SATA drives per node



LINUX JOURNAL

Executive Editor	Jill Franklin jill@linuxjournal.com
Senior Editor	Doc Searls doc@linuxjournal.com
Associate Editor	Shawn Powers shawn@linuxjournal.com
Associate Editor	Mitch Frazier mitch@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
Security Editor	Mick Bauer mick@visi.com
Hack Editor	Kyle Rankin lj@greenfly.net
Virtual Editor	Bill Childers bill.childers@linuxjournal.com

Contributing Editors
Ibrahim Haddad • Robert Love • Zack Brown • Dave Phillips • Marco Fioretti • Ludovic Marcotte
Paul Barry • Paul McKenney • Dave Taylor • Dirk Elmendorf • Justin Ryan

Proofreader	Geri Gale

Publisher	Carlie Fairchild publisher@linuxjournal.com
General Manager	Rebecca Cassity rebecca@linuxjournal.com
Senior Sales Manager	Joseph Krack joseph@linuxjournal.com
Associate Publisher	Mark Irgang mark@linuxjournal.com
Webmistress	Katherine Druckman webmistress@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 Panel
Brad Abram Baillio • Nick Baronian • Hari Boukis • Steve Case
Kalyana Krishna Chadalavada • Brian Conner • Caleb S. Cullen • Keir Davis
Michael Eager • Nick Faltys • Dennis Franklin Frey • Alicia Gibb
Victor Gregorio • Philip Jacob • Jay Kruizenga • David A. Lane
Steve Marquez • Dave McAllister • Carson McDonald • Craig Oda
Jeffrey D. Parent • Charnell Pugsley • Thomas Quinlan • Mike Roberts
Kristin Shoemaker • Chris D. Stark • Patrick Swartz • James Walker

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: +1 818-487-2089
FAX: +1 818-487-4550
TOLL-FREE: 1-888-66-LINUX
MAIL: PO Box 16476, North Hollywood, CA 91615-9911 USA
Please allow 4–6 weeks for processing address changes and orders
PRINTED IN USA

LINUX is a registered trademark of Linus Torvalds.

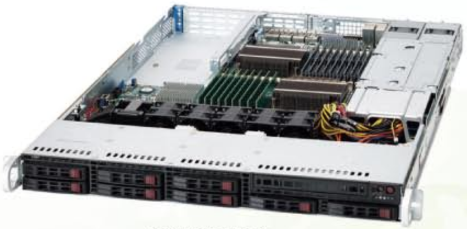
PRINTED WITH SOY INK®

SUPERMICRO®

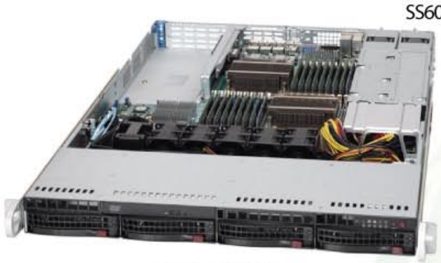
Resource Optimized

WITH ONBOARD DUAL 10GbE

SS6026T-URF4+
SS6026T-6RFT+



SS1026T-URF4+
SS1026T-6RFT+



SS6016T-6RFT+
SS6016T-URF4+

Reduce IT Investments and Extend Technology Refresh Cycles

SS2026T-URF4+
SS2026T-6RFT+

- Reduce Complexity and Cost for Enterprise, Virtualization, and Datacenter Environments
- Low Latency 10GbE Connection to the Virtual Machines for Near Native Performance
- Redundant 94%+ Platinum Level / 93%+ Gold Level High-Efficiency Power Supplies
- 50% Lower Per-Port Cost Compared to Industry Standard 10GbE Add-On Card
- Onboard SAS2.0 Hardware RAID (on select motherboards) / IPMI / Onboard 10GbE or 4x 1GbE LAN
- 18 DIMMS Support Up to 192GB DDR3 1333/1066/ 800MHz ECC Registered DIMM / 48GB Unbuffered DIMM
- UIO (Universal I/O) with Flexible Add-On Card Support Including 3 Add-On Cards (2FHFL, 1LP) in 1U and 6 Add-On Cards in 2U
- 2.5" and 3.5" SAS2.0/SATA HDD Solutions Available
- Intel® Xeon® 5600 / 5500 Series Processor Support

intel® Xeon® inside™

Powerful. Intelligent.

80 PLUS PLATINUM

80 PLUS GOLD

94 93%

AMAX	Arrow Electronics	ASI	Bell Micro	Ingram Micro	MA LABS	Synnex	Tech Data
1-800-800-6328	1-888-427-2250	1-800-2000-ASI	1-800-232-9920	1-800-456-8000	1-408-941-0808	1-800-756-5974	1-800-237-8931
www.amax.com	www.arrownacp.com	www.asipartner.com	www.bellmicro.com	www.ingrammicro.com	www.malabs.com	www.synnex.com	www.techdata.com

www.supermicro.com

SHAWN POWERS

Hack the Universe!

This month, we show you how to create black holes, swallow the Earth in a cloud of exotic matter, pummel the very fabric of space-time into nothingness and possibly even divide by zero. If you think you accidentally picked up a copy of *Mad Scientist Monthly*, fear not. Although I may have exaggerated the details a bit, it's still an exciting issue—the focus is "Hack This".

Reuven M. Lerner starts things off with the notion that knowledge is power and provides a list of his favorite books for this year. Although my personal tastes lean a little more toward science fiction, Reuven has a great list of resources that should be on every mad scientist's, er, I mean programmer's shelf. Dave Taylor follows up with a solution to a frustrating problem: renaming files. It may seem like a simple thing, but anyone who ever has renamed thousands of files really appreciates the power of scripting. Dave shows how to do it.

The hacking issue certainly makes a few of us nervous, so security seems a wise topic to include this month. Mick Bauer helps us feel a bit more at ease with his continuing series on transparent firewalls. Jes Fraser also helps us test the security of our Web applications with her article on Samurai. Because Kyle Rankin just got back from DEF CON, we're all feeling a bit paranoid ourselves. It's nice that Mick and Jes have our back. Speaking of Kyle, unlike those Internet bad guys full of secrecy and evil, he's willing to share his knowledge with the rest of us. This month, he has a handful of useful hacks he learned while at DEF CON. If you want to stay off the "Wall of Sheep", you'll want to check out his column.

What hacking issue would be complete without discussing the most complex fabricated system ever developed in the history of mankind? No, I don't mean Emacs keybindings, but rather the Large Hadron Collider, or LHC, over in Switzerland. What better way to analyze exotic particles than with open-source software! Carl Lundstedt takes us behind the scenes at CERN and shows how the data is getting crunched in that little-black-hole-creation factory.

If the LHC is a little too large for your hacking appetite, and at 17 miles in circumference, we certainly understand, perhaps Chinavision's Pico Projector is more your cup of tea. Kyle Rankin not only reviews the tiny embedded Linux device, but also covers how he hacked it—and bricked it. You'll have to read the details for yourself, but it's nice to see the elite run into trouble from time to time as well. Add to that Daniel Bartholomew's hack for finding his lost Nokia N900, and we have the small end of the device-hacking covered along with the big end.

For many of us, hacking is much more about programming than it is about soldering irons. Thankfully, Koen Vervloesem shows how to control our desktops with D-Bus. Whether you want to make applications talk to each other or tweak the way programs interact with humans, D-Bus is that back channel you can use to orchestrate the perfect desktop harmony. If you want to take it further and manipulate your computer remotely with a smartphone, Jamie Popkin walks through that process as well.

Things get a little more in-depth as Bryan Childs introduces Rockbox. Sure, most of us know Rockbox as the alternate operating system for a bunch of media-playing devices, but what many don't know is that although it's open source, it's *not* actually Linux. Why did we print such an article? Well, we figured this month we'd hack your expectations a bit too. Anyone interested in Linux will likely be interested in Rockbox. Check out Bryan's article and see for yourself.

This issue is bound to push you over the edge from intelligent Linux user to evil genius. At the very least, it will help prepare you for a battle against the bad guys. We hope you enjoy the Hack This issue; we definitely enjoyed putting it together for you.■

Shawn Powers is the Associate Editor for *Linux Journal*. He's also the Gadget Guy for LinuxJournal.com, and he has an interesting collection of vintage Garfield coffee mugs. Don't let his silly hairdo fool you, he's a pretty ordinary guy and can be reached via e-mail at shawn@linuxjournal.com. Or, swing by the #linuxjournal IRC channel on Freenode.net.

servers ▶ DIRECT®

MORE PRODUCTS, BETTER SERVICE, GUARANTEED.

GO STRAIGHT TO THE SOURCE!

1.877.727.7887 | www.ServersDirect.com

WANT YOUR BUSINESS TO BE MORE PRODUCTIVE?

The ServersDirect® Systems powered by Intel® Xeon® Processor provides the quality and dependability to keep up with your growing business.

HIGH EFFICIENCY POWER & HIGH STORAGE CAPACITY SERVER SDR-S4314-T36

ENTERPRISE-LEVEL HIGH CAPACITY STORAGE SERVER

+ Maximum 3.5"" hot-swap drives density
  - 36x (24 front + 12 rear) Hard Drive Bays
  - E1: Expander supports SAS
+ Redundant (1+1) 1400W Gold Level power supply with PMBus function
+ Redundant, Hot-pluggable cooling system, Power Supplies, Hot-swap drives
+ Support up to 192GB DDR3 1333/ 1066/ 800MHz ECC Reg DIMM
+ Intel® 82576 Dual-Port Gigabit Ethernet Controller

SDR-S1343-T04

1U INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 4X 3.5" HOT-SWAP SATA DRIVE BAYS

- Supermicro 1U Rackmount Server with HE 560W Power Supply
- Supermicro Server Board w/Intel® 5520 Chipset
- Support up to Dual Intel® 5500 series Xeon® Quad/ Dual-Core, with QPI up to 6.4 GT/s
- Support up to 96GB DDR3 1333/ 1066/ 800MHz ECC Reg. DIMM
- 4x 3.5" Hot-swap SATA Drive Bays
- Intel® 82576 Dual-Port Gigabit Ethernet Controller

SDR-S2318-T06

2U INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 8X 3.5" HOT-SWAP SAS/SATA BAYS

- Supermicro 2U Rackmount Server w/650W HE Power Supply
- Supermicro Server Board w/Intel® 5520 Chipset
- Support up to Dual Intel® 5500 series Xeon® Quad/ Dual-Core, with QPI up to 6.4 GT/s
- Support up to 192GB DDR3 1333/ 1066 / 800MHz ECC Reg DIMM
- 6x 3.5" Hot-swap SAS / SATA Drive Bays
- Dual Intel® 82574L Gigabit Ethernet Controller

SDR-S3308-T08

3U INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 8 X 3.5" HOT-SWAP DRIVES TRAYS

- Supermicro 3U Rackmount Server w/650W HE Power Supply
- Supermicro Server Board w/Intel® 5520 Chipsets
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 192GB DDR3 1333/ 1066 / 800MHz ECC Reg DIMM
- 8 x 3.5"" Hot-swap Drives Trays 6 x SATA Hard Drives Supported
- Dual Intel® 82574L Gigabit Ethernet

SDR-S4309-T08

TOWER/4U INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 8 X 3.5" HOT-SWAP DRIVES TRAYS

- Supermicro Tower 800W Redundant Power Supply
- Supermicro Server Board w/ Dual Intel® 5520 Chipsets
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 192GB DDR3 1333/ 1066 / 800MHz ECC Reg DIMM
- 24x 3.5" Hot-swap SATA Drive Bay
- Intel® 82576 Dual-port Gigabit Ethernet Controller

SDR-S4314-T08

TOWER/4U INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 8X 3.5" HOT-SWAP SATA BAYS (SUPPORT UP TO 4 DOUBLE-WIDTH GPU)

- 4U Rackmountable / Tower with 1400W Gold Level Redundant Power Supply Optional Rackmount Kit
- Supermicro Server Board w/ Dual Intel® Dual Chipset ICH10R
- Support up to Dual Intel® 5500 series Xeon® Quad/Dual-Core, with QPI up to 6.4 GT/s
- Support up to 192GB DDR3 1333/ 1066 / 800MHz ECC Reg DIMM
- "• 8 x 3.5"" Hot-swap Drives Trays Supports up to 6 SATA Drives"
- Intel® Dual 82574L Gigabit Ethernet
- **4x NVIDIA Tesla C1060 GPU Cards**

SDB-S7002-T00

9U INTEL® XEON® PROCESSORS 5500 SERIES SERVER W/ 50X HOT-SWAP SATA II / SAS BAYS

- Supermicro SBE-714D-D28 Enclosure chassis with two 1400W power supplies. Up to 14 hot-plug processor blades
- Support Intel based blades
- One Management modules comes standard with each enclosure
- One hot-plug management modules providing remote KVM and IPMI 2.0 functionalities
- Support one hot-plug Gigabit Ethernet switch, Pass - Through Module

10th year

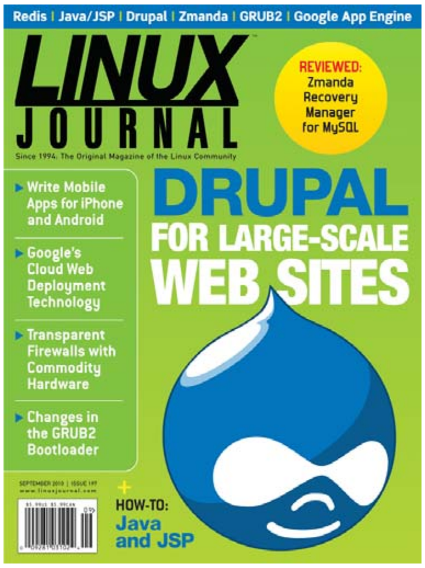
SERVERS DIRECT CAN HELP YOU CONFIGURE YOUR NEXT HIGH PERFORMANCE SERVER SYSTEM - CALL US TODAY!

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 to assemble systems that require a little extra. Servers Direct - your direct source for scalable, cost effective solutions.

1.877.727.7886 / www.ServersDirect.com

intel Xeon inside™ Powerful. Intelligent.

letters



Why I Gave Up on Linux and Returned to Windows XP

I am a senior IBM mainframe software engineer and database administrator, and in its day, I was highly proficient in MS-DOS line commands. I have written thousands of lines of computer code and tested hundreds of thousands of lines of computer code. I recently tried several Linux distributions, and after too much time wasted chasing solutions for issues that simply work or work simply in Windows, I made the decision to return to Windows XP to catch up on other projects.

I would like to begin by complimenting all the software engineers, developers and testers that have diligently worked and contributed to Linux.

On the positive side, Linux is void of one of the key architectural design faults of all MS Windows versions since Windows 95, that being what I often refer to as the GFR (God Forsaken Registry) or Windows Landfill. I have been told that the entire Windows Registry is assembled from various files on the hard disk and loaded into memory upon startup, regardless of which applications are running or not running. The Registry contains nearly all the critical Windows operating system settings along with settings and parameters for almost all Windows applications. Unfortunately, many Windows applications do not completely remove all of their entries when uninstalled, leaving heaps of junk behind in the Registry. There are numerous Registry "cleaners", but I have yet to find one that accurately removes all the junk left behind by uninstalled applications. Furthermore, applications, including MS Windows, use the Registry as a temporary location for temporary settings but fail to remove these when they are no longer needed. Thus, the Registry is forever expanding in size. Hard disk space is not the issue here. The issue is waste of RAM, operating inefficiency, stability and OS portability.

On the negative side, I encountered a persistent font-rendering issue with Ubuntu, Kubuntu, Xubuntu and Linux Mint. I live on an HP nc8430 Notebook PC with 1680x1050 WXGA resolution, where increasing font size is a must. The feature works for system fonts, but it did not work for key applications like Firefox. My research revealed that the issue already has been documented throughout the Web and is related to these distributions failing to update the standard configuration file that Firefox and other applications rely upon for their font settings. Although the Ubuntu folks are aware of it, they seem uninterested in fixing it. The issue is also present in Linux Mint, which is based on Ubuntu. The fonts worked as expected for both Gentoo and PCLinuxOS. However, I was unable to get my Intel 3945 ABG wireless card to work with PCLinuxOS (both GNOME and KDE versions).

In every incarnation of Linux that I tried, my notebook fan was *always on*, even when no applications were running with CPU near 0%. This issue may be related to CPU throttling and fan throttling. Linux needs to resolve this before I can use Linux on a notebook without the associated side effects, such as annoying noise of the fan on max, increased fan wear and reduced battery life.

There is nothing more this user would like than to depart the world of Microsoft Windows for Linux, but the path will have to be paved with far fewer rivers to cross or mountains to climb just to get there.

--
Dave Freeman

Your frustrations are less common than even a few years ago, but indeed, there are many computer models, laptops especially, that don't behave right under Linux. I could go on a rant about vendors and closed drivers, but that won't help you get on-line with your wireless card. The best, and admittedly most frustrating, way to get great Linux support on a notebook computer is to research before buying. I realize it's too late in your case, but without massive amounts of tweaking, some laptops are just difficult to penguinify.

As to the font issue, well, one of the frustrating things about standards is that there are so many. I don't know about your specific issue, but I'm sure Ubuntu expects the Firefox folks to do things their way, and vice versa. In the short term, some things require tweaking no matter what your hardware might be. Good luck, and hopefully you'll still get to tinker with Linux on your Windows laptop. Perhaps a virtual machine with VirtualBox or VMware?—Ed.

Open-Source Software in the UK

Thank you for the article "OSS: Europe vs. the United States" by Doug Roberts on LinuxJournal.com (**www.linuxjournal.com/content/oss-europe-vs-united-states**). You may be interested in the response to the recently submitted Freedom of Information Act request to the UK Home Office: **www.whatdotheyknow.com/request/home_office_compliance_with_hmg**.

The request aims to highlight the gap between existing policy and a lack of compliance with it. The policy levels the playing field for open-source software in the UK government—see linked document referred to in the request—but in practise, this is ignored, giving undue favour to established vendors such as Microsoft.

--
Charles Taylor

Pencil

I just finished reading Dirk Elmendorf's article "Cool Project Potpourri" in the August 2010 issue. I've been using the Pencil Firefox add-on (**https://addons.mozilla.org/en-US/firefox/addon/8487**), which is an alternative to Mockingbird.

Some of Pencil's benefits over Mockingbird are:

- Open source (GPL version 2).
- Pencil is free, although you can support the project with donations.
- Mockingbird might not be free once the beta period is over.
- Your designs are hosted locally.
- Exports to HTML, PNG, OpenOffice.org document, Word document and PDF.

The project's home page is at **pencil.evolus.vn**.

Keep up with the good work. *LJ* is one of the few magazines I read cover to cover more than once.

--
JS

Flash and Linux

I have been using Linux for quite a while and recently made a switch from Mac OS X to a brand-new PC, which is the first time I have seriously run a distro for my desktop! I chose Fedora 13 because although I cut my teeth on Debian or Debian-based systems, I found that I *really* like Fedora and the support it offers for a power user.

By trade, I am a developer, and I recently have been doing development using Adobe Flash (CS4), although personally, I agree with Apple Founder and CEO Steve Jobs in that Flash technology is getting quite old and is, in my own opinion, *very kludgy* at best.

Still, sites insist on using Flash for content, and if you are using 64-bit Linux (as I am), you are left with three choices:

1. Don't install Flash (my current selection).
2. Install the 32-bit player inside a wrapper (not ideal).
3. Wait for Adobe to finish "rebuilding" the 64-bit Linux player (I may be a very elderly man by then).

I am sure readers are aware that Steve Jobs has not allowed Flash into the iPhone/iPad (including iPod Touch) platform, and I personally feel others should follow suit. It is time for a better alternative to Flash that is more open. I think HTML5 may hold the answer.

I feel we need an alternative now. HTML5 will at least be built in to browsers so that needing a plugin to use the content will no longer plague users.

Obviously, this is just my two-cents' worth on the subject, but it seems like a lot of sites (including *Linux Journal*) rely on Flash in some capacity to display content. Maybe the answer is Android? I know it is mainly for phones, but it is lightweight, and from what I have seen, it's robust enough to do the job.

Thank you for listening to my opinion, and I hope that this can be published and discussed.

--
J. Mike Needham

Your 64-bit frustration is one many folks share, myself included. In fact, on my desktop machines, I usually install 32-bit Linux and use a PAE kernel to use all my system memory. It's a kludge for sure, but on my desktop, I find it's quicker and more reliable just to stick with 32-bit.

Like you, I see HTML5 as a viable and logical replacement for many of the things Adobe Flash can do. I think Flash is used often enough, however, that for a full Internet experience (good or bad), we'll likely be using Flash for a very long time. Hopefully, before long 64-bit Flash won't be an issue, and we can live in a Flash/HTML5 world at the same time.—Ed.

Thanks Mick!

As I print out your series on OpenVPN [see Mick Bauer's column in the February, March, April, May and June 2010 issues] to set one up both at home and at work, I must write to let you know how much I look forward to your columns in *Linux Journal*. I subscribe and look forward to your column every month. But, in particular, your series on OpenVPN was great. We're using it for some Web-facing servers we're renting on Rackspace and will be using it for the computing environment we're building here at the office. As well, I'll use it at home to ensure that both my wife and myself are able to use it while on the road. I'll probably even set up accounts for my kids for when they're on the road.

As well, I did appreciate your article on locking down your desktop before going to DEF CON [see Mick's "Brutally Practical Desktop Security" in the October 2009 issue]. I was unable to attend this year, but I intend to do so next year and will use your article to help me prepare for taking a portable computing device with me.

Please keep writing Paranoid Penguin. I enjoy both the information that you present as well as your writing style. Your effort is not wasted and is greatly appreciated by me.

--
Mark

Mick Bauer replies: *When your very generous e-mail arrived, I was frantically trying to finish (past deadline of course!) a difficult article on OpenWrt, part of a series whose production has, at times, been characterized by technical failures and frustrations. Your generous words could not have come at a better time!*

Although I don't flatter myself to think I'm smarter than or even as smart as most of my readers, I have had the impression that the more trouble I have getting something to work (OpenLDAP springs to mind), the more useful my readers tend to find my tutorials. But it's hard to remember this in the heat of battle (so to speak), so again, I much, much appreciate your taking the time to write!

It's especially gratifying to hear you are using OpenVPN both at home and at work. Sometimes I worry that since I do all of my testing in a home networking lab, my stuff has less relevance to readers who need to build "production-grade" solutions at work. I take some comfort knowing that security controls tend to need to be the same whether you apply them to two systems or 200. Anyhow, I'm glad you found my OpenVPN tutorials to be so useful!

Wishing you all the best, and promising many more columns for at least a few more years.

Patents

I'm discouraged. I've been using Linux for years, and now I'm trying to switch from Windows to Linux for video editing, DVD authoring and so on. It seems difficult to do this without using a program or library that seems to be violating a "patent" somewhere. I want to respect the work of others.

From what I understand, the creators of those programs (the open-source ones) coded them themselves, but because there is resemblance with an original, copyrighted work, there is a "patent" violation. If I understand it correctly, like for MP3, there are algorithms you cannot avoid in the programming language to be able to get the same results.

I'll give you an example. Even if I don't use the Ubuntu Multiverse repository, it's giving me warnings about patents or licenses: gstreamer0.10-plugins-ugly (in the Universe repositories) is telling me that I might be breaking the law in my country and that I might need a license, but that I can use it for research.

I want to use these programs for personal use now, but I might use them professionally later, and I do not wish to end up with a big lawsuit once I make money from my work.

I don't want to exaggerate, but since you guys are professionals in the field of Linux and open source, I would really appreciate your advice and direction in this matter.

--
Jean-François Tassé

Patents are funny things. Not comical, but funny in that I don't understand them completely either. Add to that copyright, intellectual property, licensing, fair use and DRM, and you have the makings for a very boring horror film.

My suggestion is to use the tools you're comfortable with now, and if you decide to become professional with the tools, contact the software folks to see how much and whom to pay. Generally, if you stick with open audio/video formats, you should be fine. The problems usually come when using licensed codecs. If you are okay with the open-source alternatives, I suspect you'll be fine.—Ed.

LJ Addiction

I'll be picking up the latest *Linux Journal* at my local bookstore this week, which will mark a full year of feeding the addiction that reading your fine magazine has become for me.

I've been using Ubuntu for a couple years now, but since I started reading *Linux Journal*, I'm getting so much more from my new favorite OS. There's a lot for newbies to learn, especially for a cross-platform developer like myself. Your magazine feeds that need in ways I never imagined when I picked up my first issue—from the series on NoSQL databases, to the excellent column on shell scripts, to the many feature articles about inspiring cool projects, *Linux Journal* has become the centerpiece of my Linux education, often introducing me to things I didn't know I'd be interested in.

Linux Journal is the only magazine in which I read every word, from the letters to the closing comment, and I enjoy every bit of it.

I suppose I could save a couple bucks by subscribing, but I feel it doesn't hurt to raise Linux's visibility even in a small way by helping your issues fly off the shelves at my local bookstore.

Please extend my thanks to the team for producing such an uncommonly useful publication with every issue.

--
Richard Gaskin

Well shucks, if you keep talking like that, we won't be able to get our heads through the door! In all seriousness, when we received your letter, the editorial staff bounced appreciative e-mail messages back and forth. Letters like yours make the whole thing worthwhile. Thanks again.—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 +1-818-487-2089. E-mail us at subs@linuxjournal.com or reach us via postal mail, Linux Journal, PO Box 16476, North Hollywood, CA 91615-9911 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 at **www.linuxjournal.com/contact** or mail them to Linux Journal, PO Box 980985, Houston, TX 77098 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/enewsletters**.

More TFLOPS, Fewer WATTS

Microway delivers the fastest and greenest floating point throughput in history

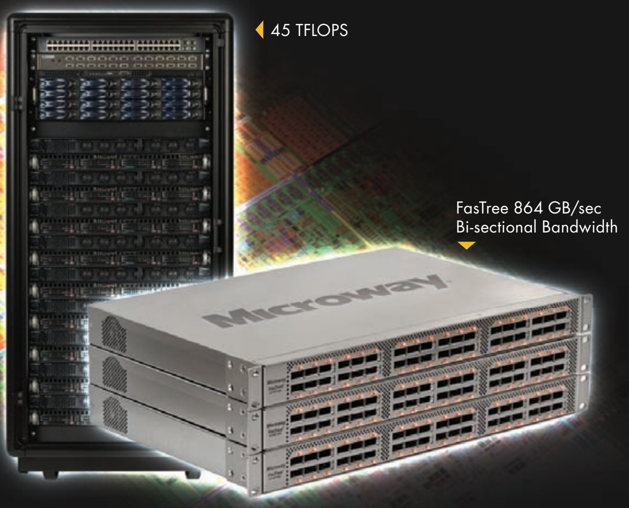
Enhanced GPU Computing with Tesla Fermi

- 480 Core NVIDIA® Tesla™ Fermi GPUs deliver 1.2 TFLOP single precision & 600 GFLOP double precision performance!
- New Tesla C2050 adds 3GB ECC protected memory
- New Tesla C2070 adds 6GB ECC protected memory
- Tesla Pre-Configured Clusters with S2070 4 GPU servers
- WhisperStation - PSC with up to 4 Fermi GPUs
- OctoPuter™ with up to 8 Fermi GPUs and 144GB memory

New Processors

- 12 Core AMD Opterons with quad channel DDR3 memory
- 8 Core Intel Xeons with quad channel DDR3 memory
- Superior bandwidth with faster, wider CPU memory busses
- Increased efficiency for memory-bound floating point algorithms

Configure your next Cluster today!
www.microway.com/quickquote
508-746-7341



FasTree™ QDR InfiniBand Switches and HCAs

- 36 Port, 40 Gb/s, Low Cost Fabrics
- Compact, Scalable, Modular Architecture
- Ideal for Building Expandable Clusters and Fabrics
- MPI Link-Checker™ and InfiniScope™ Network Diagnostics

Achieve the Optimal Fabric Design for your Specific MPI Application with ProSim™ Fabric Simulator

Now you can observe the real time communication coherency of your algorithms. Use this information to evaluate whether your codes have the potential to suffer from congestion. Feeding observed data into our IB fabric queuing-theory simulator lets you examine latency and bi-sectional bandwidth tradeoffs in fabric topologies.

NVIDIA. TESLA with CUDA | intel Channel Partner Premier 2010 | AMD Fusion Partner Premier | GSA

GSA Schedule Contract Number: GS-35F-0431N

Microway®
Technology you can count on℠

diff -u

WHAT'S NEW IN KERNEL DEVELOPMENT

David Airlie has said that he won't accept any **DRM** (Direct Rendering Manager) drivers from companies that release only an open-source-crippled version of the driver, so they can layer their proprietary user-space version on top. Amazingly, he came under criticism for that from people accusing him of playing to his employer's (**Red Hat**'s) business interests. He denied it, and several other folks also said David wouldn't do that. Personally, it seems to me that David is just upholding a solid open-source ethic. There are real technical reasons why layering proprietary user-space code over a crippled kernel driver is not the best way to go. At the same time, it's not clear how to prevent companies from doing that. If they add a certain level of functionality, the driver will make it into the kernel, and then they still can put their proprietary enhancements on top. We'll be seeing this debate heat up in the near-to-middle term, because of all the 3-D video drivers that soon will be flooding the world.

It looks like the **console code** is going to receive a major overhaul in the near future. **Mattia Jona-Lasinio** wrote up an in-kernel **VT102 emulator**, which essentially was rejected on the grounds that everything it did could be implemented as an enhancement to the console code. Not only that, but as **Alan Cox** pointed out, there's a whole big pile of 3-D graphics people who very much want the kernel console code to be revamped and enhanced. **James Simmons** also has been interested in fixing up the console code, so it looks as though he and Mattia are going to work on it together. They each compiled big lists of things that would be very good to fix, and in spite of the daunting complexity, it does seem as if they're going ahead with it. And, with the blessing of someone like Alan Cox, it seems like they'll have a lot of support for getting their code ultimately into the kernel.

Once in a while a strange and funny thing crops up on the mailing list. This time, a completely anonymous person suggested implementing **Kademlia** at the kernel level. Kademlia is a peer-to-peer networking protocol used for creating large, anonymous data-sharing networks. The anonymous person suggested that with Kademlia at the kernel level, it would obviate all the user-space implementations and allow all Linux computers to become one with the universe, in an immense, all-powerful file-sharing service that could never be stopped! Unfortunately for this plan, a lot of folks pointed out that the existence of plenty of user-space implementations of Kademlia was a good indication that there was no need to stick it into the kernel. So, the anonymous person went away unsatisfied.

Nick Piggin has started a new git repository, just to house patches for scaling the **VFS** (Virtual Filesystem). One patch on the system helped make name lookups go much faster by reducing the amount of data movement that had to happen for each lookup. VFS speedups are great because all filesystems rely on the VFS for fundamental operations. A speedup in the VFS means a speedup for every filesystem out there.

—**ZACK BROWN**

Boxee Scores a Knockout

One of my passions is seeking the holy grail of the home-entertainment center. I've often written about creating the perfect media-center PC, and I've tried almost every set-top box ever made. They all seem to be limited in one area or another—whether they lack in the playback of local media (Roku and PS3) or don't stream on-line content from Hulu and Netflix (XBMC, ASUS O-Play and GeeXboX).

Enter Boxee. My first experiences with Boxee were not great. I was disappointed with the interface, it locked up more than not, and its various media-playing abilities were awkward and hard to navigate. Recently, however, the Boxee software has matured to the point that it's a viable and beautiful media center. Although Boxee's Linux version is unable to play Netflix, due to Netflix's dependence on Microsoft's Silverlight, the interface's usability will pave the way for wide acceptance of the Boxee device being built by D-Link (assuming when it ships it can play Netflix, that is).

My advice is to play with Boxee now, and watch for our review of the sub-$200 box when it ships. If the software is similar to what you can download today, we may have a new heavyweight champion!

—**SHAWN POWERS**

They Said It

Moore's Law has been the name given to everything that changes exponentially. I say, if Gore invented the Internet, I invented the exponential.
—**Gordon Moore**

There are 10 types of people in the world: those who understand binary, and those who don't.
—**Anonymous**

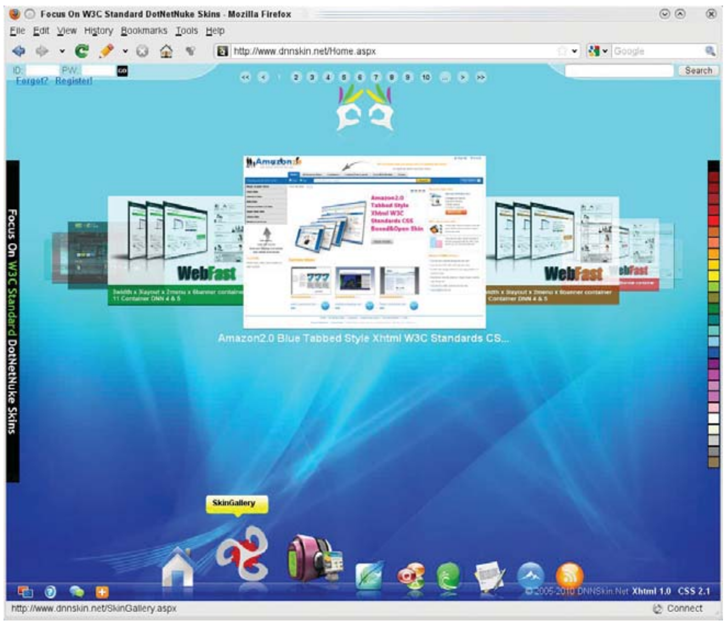
URLs are the 800 numbers of the 1990s.
—**Chris Clark**

UNIX is basically a simple operating system, but you have to be a genius to understand the simplicity.
—**Dennis Ritchie**

Almost everybody today believes that nothing in economic history has ever moved as fast as, or had a greater impact than, the Information Revolution. But the Industrial Revolution moved at least as fast in the same time span, and had probably an equal impact if not a greater one.
—**Peter Drucker**

If with the application base, with the tools that we have, with the user understanding and momentum and everything going on, we can't compete with...whatever the weird collection of Android machines is going to look like, shame on us.
—**Steve Ballmer, Microsoft CEO**

NON-LINUX FOSS


Focus On W3C Standard DotNetNuke Skins - Mozilla Firefox
File Edit View History Bookmarks Tools Help
http://www.dnnskin.net/Home.aspx
Amazon2.0 Blue Tabbed Style Xhtml W3C Standards CS...
WebFast
SkinGallery
Focus On W3C Standard DotNetNuke Skins
http://www.dnnskin.net/SkinGallery.aspx

DotNetNuke Skins (from www.dnnskin.net)

Back around 2002 PD (Pre-Drupal), the *Linux Journal* Web site ran on PHP-Nuke. PHP-Nuke is no longer free software, so it seems fitting that there's a free and open-source CMS for the .NET platform named DotNetNuke (aka DNN). The DNN Web site claims that DotNetNuke is "The most widely adopted Web Content Management Platform for Microsoft .NET powering more than 600,000 Web sites."

The open-source version of DNN is known as the community edition. In addition to the community edition, a number of nonfree versions are available that provide extra functionality and technical support.

DNN is extensible via add-on modules. The community edition includes 25 add-on modules, and there is a DNN Forge that contains additional modules and skins. There's also a separate site (**snowcovered.com**) that contains more than 8,000 third-party modules and skins for sale.

DotNetNuke is written in VB.NET, but because it is based on the .NET platform, any .NET language can be used for writing add-ons (although VB programmers and C# programmers normally don't hang out together). There is some indication in the forums that DNN will run on Mono, but expect a bumpy road if you try.

DotNetNuke is licensed under an MIT License (although in some places it mentions a BSD license), and licensing and copyright monitoring is managed actively.

—MITCH FRAZIER

Security at LinuxJournal.com

This month's issue is all about hacking, but as we all know too well, hacking is not always a good thing. Sure, sometimes it can mean taking apart your Roomba and making a toy for your hamster, or it can mean finding a creative solution to a technical problem. Unfortunately, it also can mean violating a secure system and generally wreaking havoc. Fortunately, LinuxJournal.com has some valuable information that may help you avoid such a situation. You may notice an area on the right side of the page with a list of topics we are particularly interested in. Right now, one of them is security, and we think you'll find this section incredibly valuable. Here you'll find a variety of news, tutorials and opinions about what's relevant in Linux security. Check it out at **www.linuxjournal.com/tag/security**, and as always, happy (and safe) surfing!

—KATHERINE DRUCKMAN

LJ Index November 2010

1. Thousands of Google hits for "Android Tablet": **1,320**
2. Number of Android Tablets available in the US, as of Q2 2010 (including the Archos 7): **1**
3. Number of confirmed/available/rumored Android Tablets, as of Q2 2010: **45**
4. Thousands of Google hits for "MeeGo Tablet": **216**
5. Thousands of Google hits for "Windows Tablet": **423**
6. Number of BOINC-based projects (such as seti@home) currently available: **36**
7. Number of top-level Apache Software Foundation projects: **77**
8. Percent of Web sites that use Google Analytics: **54**
9. Percent of Web sites that use Google AdSense: **31**
10. Percent of Web sites that use DoubleClick.net: **28**
11. Percent of blog sites that use WordPress: **86**
12. Percent of CMS sites that use Drupal: **41**
13. Percent of CMS sites that use vBulletin: **17**
14. Percent of CMS sites that use Joomla!: **9**
15. Percent of eCommerce sites that use IBM Websphere Commerce: **40**
16. Percent of eCommerce sites that use Yahoo Store: **15**
17. Percent of Web sites that use UTF-8: **63**
18. Percent of Web sites that use ISO-8859: **37**
19. Percent of Web sites that use Apache: **61**
20. Percent of Web sites that use IIS: **22**

Sources:
1, 4, 5: Google | 2: Common knowledge
3: www.androtablets.com | 6: boinc.berkeley.edu
7: www.apache.org | 8–20: trends.builtwith.com

Hacking, Old-School

When you mention hacking in the general public, the image most people think of is a nerdy guy breaking into a computer system from his bedroom. This month, I take a look at some of the tools available to do exactly that. Of course, this is for information purposes only, so please don't do anything nasty. Remember, with great power comes great responsibility. Most people have heard of tools like Nmap or Nessus, but here I look at some other available tools for playing with networks.

The granddaddy of network utilities is tcpdump. This utility simply listens to all network traffic going by and records the packets for later analysis. If you have more than one network interface, you can select which one to listen to with the option `-i interface`. By default, tcpdump puts your network card into promiscuous mode, so it can record all packets that exist on the network cable. If you want to limit the packets recorded simply to those destined for your machine, use the `-p` option to turn off promiscuous mode. Lots of options are available to tcpdump, so check out the man page for more details.

Say you want to find out what machines exist on your network. Several tools can do this by actively sending out queries on the network. The problem with this technique is that you end up creating traffic on the network, which may be noticed by a good network administrator. A way around this is to use the tool p0f. This utility uses passive techniques to try to guess what machines exist on the network and properties about those machines. If you have more than one network interface, you can select which interface to use with the option `-i interface`. p0f can work with tcpdump files. If you have a tcpdump file that you created earlier, you can make p0f use it rather than live capture with the `-s file` option. You also can use p0f to record network traffic into a tcpdump file with the `-w file` option. If you're using p0f in a script, use the `-o file` option to dump the output into a text file for later perusal.

By default, p0f looks only at network packets that are addressed to the machine where it is running. To look at all the packets that go by on the network, you need to set the card to promiscuous mode with the `-p` option. By default, p0f sees machines only when they open new connections. You can try to guess what's going on with already-opened connections with the `-O` option. This option can generate a lot of data, so you probably won't want to use it for an extended period of time.

More and more often, machines actually are located behind routers and NATs, so they don't really show up as individual machines. You can try to identify these types of machines with the `-M` option. This uses the masquerade-detection algorithm to try to identify individual machines in these situations.

Once you know what machines exist on the network, you may be interested in what traffic is traveling to those machines, as well as who is generating this traffic. You can use dsniff to see the user names and passwords being used to access services on the network. It can handle many different protocols, such as FTP, HTTP, POP, IMAP, X11 and many others. You can tell dsniff on which interface to listen with the `-i interface` option. Like most network tools, you can read previously recorded network data with the `-p file` option. Alternatively, you can use dsniff to record the network data rather than parsing it with the `-w file` option. You can enable automatic protocol detection by using the `-m` option. This can give you some of the gory details about people on your network.

Now that you know some details about your network, and the people on it, you may want to check the security of some of the services provided. One common target for security problems are Web servers. You can use the nikto tool to assess your Web server's security. Select the host with the `-h hostname` option. If you have a series of hosts you want to check, place the hostnames (or IP addresses) in a text file, and hand them to nikto with the `-h file` option. The default port nikto looks at is port 80. If you want to check out a Web server on some other port, simply use the `-p port` option. Tons of extra options exist in terms of what specific security issues to test for, far too many to mention here. See the manual at the project's home page for more information (**cirt.net/nikto2**).

The hack I cover this month is how to check your own backyard. Many people will use this kind of knowledge for nefarious purposes. A utility you can use is chkrootkit. This utility analyzes your systems and tries to determine whether they've been tampered with. You can get a list of the tests it can perform with the `-l` option. With the standard install on my Ubuntu box, chkrootkit has 69 available tests. You can check things like whether `ls` has been infected, or you can check for evidence of rootkits that may have been installed. Hopefully, you won't find anything when you run chkrootkit.

Now you have a few new tools you can use to play around with your networks. Hopefully, you won't find anyone doing anything nasty. And remember, if you are going to use these tools, be sure you have permission before you do anything that might be frowned on. Other than that, hack away and keep learning.

—JOEY BERNARD

WHERE'S MY CD?

I don't like to take my music collection to work. Don't get me wrong. I like having access to my Jonathan Coulton and Weird Al Yankovic albums when I'm troubleshooting servers, but I don't like having my personal music collection stored on a work computer. It just gives me the heebie-jeebies. Thankfully, with a broadband Internet connection and a little bit of work at home, I can stream my personal collection anywhere I happen to be sitting.

More options probably exist, but the two I generally fiddle with are kPlaylist and Jinzora. Both have matured during the years, and both can stream your music collection a number of different ways—even with a Web-based Flash player in a pinch.

Granted, setting them up can be a little tough, but having your music collection available anytime and anywhere is worth the effort. In fact, if you're really a tinkering sort of person, it's possible to get them to stream your video collection as well. To be fair, your boss might be a little more annoyed if you're watching your favorite television show than if you're listening to music, but it's still nice to have the option.

—SHAWN POWERS

Drobo Shmobo

Last month, I mentioned my new Drobo FS, which is a Linux-based NAS device from the folks at Data Robotics. Although I'm still moderately impressed with its functionality, I find myself frustrated at its limitations. Don't get me wrong; if you're just looking for a place to store lots of data, the Drobo FS is great. Unfortunately, I want a little bit more. For those of you considering a Drobo FS, or something similar, let me suggest the following: *build your own*.

Really, it's not that difficult. What I've done is taken the five 2TB drives out of the Drobo FS and placed them into a desktop tower. As long as your desktop's power supply is big enough to handle six SATA drives and has a gigabit Ethernet port, the setup is fairly simple:

1. Install your favorite distribution onto a sixth, smaller, SATA drive.

2. Create a software RAID5 partition with the five large drives.

3. Profit!

Granted, you lose the fancy Drobo mismatched drive ability. Yes, upgrading storage isn't quite as simple. True, in the event of a hard-drive failure, you'll have to do a little command-line work, but just think of the benefits. Any program you want to run on the server is as easy as an apt-get or yum away. Heck, install a VM system (I'm using VMware Server), and you even can spin up another operating system within your SuperNAS!

Building your own NAS might not be for everyone, but if you're reading *Linux Journal*, perhaps you're the perfect candidate. Be sure to search LinuxJournal.com for information on software RAID and installing virtual machines. Chances are, you'll enjoy building your own NAS, and you even might save some money!

—SHAWN POWERS

LJ STORE'S FEATURED PRODUCT OF THE MONTH:

May the Source Be with You T-Shirt

The force is strong with this 100% cotton T-shirt featuring Tux the penguin.

- **REGULAR PRICE:** S–XL: $19.95; XXL/XXXL: $20.95
- **SALE PRICE:** S–XL: $14.95; XXL/XXXL: $15.95
- **COUPON CODE:** luke
- **Sale ends November 30, 2010.**

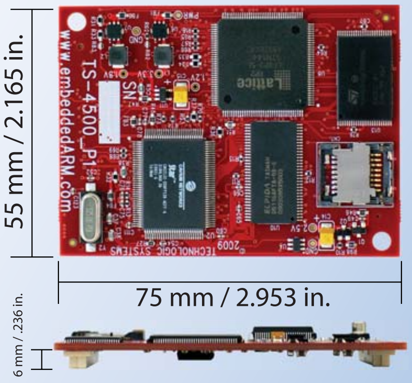
Buy yours now at **www.linuxjournalstore.com**.

May the Source Be with You

TS-SOCKET Macrocontrollers

Jump Start Your Embedded Design

Series starts at $92 qty100



TS-SOCKET Macrocontrollers are CPU core modules that securely connect to a baseboard using the TS-SOCKET dual connector standard with common pin-out interface. COTS baseboards are available or design your own baseboard for a custom solution with drastically reduced design time and complexity. Start your embedded system around a TS-SOCKET Macrocontroller to reduce your overall project risk and accelerate time to market.

- TS-4200: Atmel ARM9 with super low power
- TS-4300: Cavium ARM11 with dual 600 MHz and FPU
- TS-4500: Cavium ARM9 at very low cost
- TS-4700: Marvell PXA168 with video and 1.2 GHz CPU
- TS-4800: Freescale iMX515 with video and 800 MHz CPU
- Several COTS baseboards for evaluation & development

Design your solution with one of our engineers

- Over 25 years in business
- Never discontinued a product
- Engineers on Tech Support
- Open Source Vision
- Custom baseboards w/ excellent pricing and turn-around time
- Most products ship next day

Technologic SYSTEMS

We use our stuff.

visit our TS-7800 powered website at

www.embeddedARM.com

(480) 837-5200

2010 Book Roundup

Reuven shares his list of favorite Web/database books from the past year.

REUVEN M. LERNER

Hello, book-lovers! If you live in the Northern Hemisphere, you presumably are reading this as the weather is getting colder. But as I write this, Israel (and much of North America and Europe) is "enjoying" rather hot weather, with very high temperatures and no obvious relief in sight. Regardless of whether the temperature is high or low, I'm always game for a new book to teach me about the latest open-source Web-related technologies. Fortunately, a number of good, new books have been published during the past year. This month, I'm taking a break from my usual discussion of the latest open-source Web-related technologies, in favor of a list of some of the more interesting books I've seen this year.

Ruby, Python and Perl

Regular readers of this column already know that for about five years now, my favorite programming language is Ruby. This is due in no small part to Ruby on Rails, the Web framework that has taken the industry by storm, against which all other frameworks seem to be measured. Given Ruby's popularity, it's no surprise that publishers continue to offer a large number of Ruby-related titles. However, many of these books are aimed at more-advanced programmers, often looking to improve their programming techniques and code maintainability.

A hot topic in the Ruby community, and in the programming world in general, is that of automated testing.

Two of the more-advanced Ruby books I've seen have almost identical names, and confusingly, both come from the same publisher, Addison-Wesley. *Refactoring Ruby* by Kevin Rutherford is a combination taxonomy and tutorial, introducing Ruby programmers to common "code smells", potential problems with code that should be addressed before they get out of hand. Rutherford is the author of one of my favorite tools, Reek, which automatically identifies code smells and, thus, potentially problematic code.

The other book, *Refactoring (Ruby Edition)*, is by Jay Fields, Shane Harvie and Martin Fowler, with Kent Beck. As the title implies, this is a Ruby version of Fowler's classic *Refactoring*, which introduced the idea to many programmers that it's both possible and good to improve code without changing its functionality. The book includes many examples of how to make your code more readable and emphasizes the use of many small methods rather than a small number of long ones. As I read through this book, I alternated between feeling good about my current techniques and understanding how I might improve my code's maintainability. It's easy to read, with many clear examples, and it's good for intermediate and advanced Ruby programmers.

A hot topic in the Ruby community, and in the programming world in general, is that of automated testing. Two books released in the past year, both from the Pragmatic Programmers, address the issues of testing from different angles. One, *Rails Test Prescriptions* by Noel Rappin, is a great introduction to all the different ways you can test code in your Rails application, starting with the basics (models and controllers), moving on to mock objects and factories, and finally looking at Cucumber, Webrat and Capybara. If you are interested in the alternative RSpec test framework for Ruby, as well as testing with the scenario-driven Cucumber system, *The RSpec Book* written by RSpec's current maintainer, David Chelimsky, might be worthwhile reading. It introduces both RSpec and Cucumber as complementary, and it shows how to use the two of them to test your code effectively and easily, both before and as you write it.

Although I do spend most of my time working with Ruby, it's true that I still do use other programming languages on occasion. In particular, I often teach courses in Python programming. To date, I hadn't seen a book that went into many of the more-advanced aspects of Python programming, with a particular emphasis on object-oriented development, automated testing and metaprogramming. However, the book *Pro Python*, written by Marty Alchin and published by Apress, aims to fill in this gap, and from what I've seen, it does an excellent job. I expect I will recommend this book to students in my advanced Python development classes. Although the book is written for Python 3.x, the author recognizes that much of the Python world still is using the 2.x series, so he includes a large number of sidebars and notes indicating differences between 2.6, 2.7, 3.0 and 3.1 where appropriate.

Finally, I should add that although Perl is no longer the first language for which I reach when writing code, it still occupies a warm spot in my heart. It was, thus, something of a homecoming for me when a client recently requested that I do a project in Perl, and that I use Catalyst as the framework for that project. Perl hackers would describe Catalyst as an excellent MVC Web framework (which Ruby hackers would describe as a Perl knockoff of Rails).

Regardless of your perspective, Catalyst is a good way to write modern Web applications in Perl. And, although there certainly are similarities between Catalyst and Rails, there are enough differences that I needed an introduction to Catalyst to get me up and running relatively quickly. *The Definitive Guide to Catalyst*, published by Apress and written by Kieren Diment and Matt Trout, was just what I needed. When I had questions about how Catalyst worked, I found myself turning not only to the excellent on-line documentation, but also to this book.

New Languages

It's nice to work with languages that you already know, but there's a lot to be said for learning and working with new languages as well. Each language teaches you something new and (I would argue) improves your understanding of other languages you already know. So, I was pleased to discover *Seven Languages in Seven Weeks* by Bruce Tate, published by the Pragmatic Programmers. The book reviews (as you might expect) seven programming languages, each quite different from the others: Ruby, Io, Prolog, Scala, Erlang, Clojure and Haskell.

I had been meaning to learn several of these languages for the last year or two, and although I didn't take seven weeks to read through the book, I did use it as a way to familiarize myself with several of these languages and where they might be useful. (I'm particularly interested in Clojure and Scala, in no small part because of the possibilities for Web development in those languages.) I found the writing to be clear, the examples good, and perhaps most important, the comparisons with other languages were useful. Even if I'm not going to program in Io, for example, it's interesting to see how the data types work and how a prototype-based language operates. As the author himself writes, Io gave him a new perspective on JavaScript, undoubtedly the most popular prototype-based language in use today.

Another popular language, although we might not think of it as such, is SQL, the query language used in all relational database systems. Just as there are books and tutorials about "design patterns", examples of how you can and should structure your code, a growing number of articles and books talk about "antipatterns", examples of how you should *not* structure your code. The book *SQL Antipatterns* by Bill Karwin, published by the Pragmatic Programmers, introduces a number of ways people should not use SQL and relational databases, and then it shows ways in which database tools can and should be used. I was very happy to see that he recommended against using the "float" type, against searching through textual columns with patterns like `'%target%'`, and against storing multiple values in a single column. I have seen these (and more!) on many projects, and it's nice to have a checklist for what to avoid. I largely disagree with the author's argument in favor of putting images and other large binary objects inside BLOB columns. That said, he made a fairly convincing argument for his case and acknowledged that this is a subject of great controversy. Especially if you're fairly new to database and query design, this book might well come in handy.

World-Wide Web

When Steve Jobs publicly announced that there would be no support for Flash on iPhones and iPads, many people wondered what the alternatives were. Jobs told everyone that HTML5 offers most or all of the same capabilities. My response was, "Hmm, I guess HTML5 is further along than I thought—I'd better check it out." I've since been reading up on the various parts of the emerging HTML5 standard, and I must admit that it seems very compelling, at least at this stage.

My main source of information is Mark Pilgrim's *Dive into HTML5*, published both for free on-line and also as a book from O'Reilly. I have long found Pilgrim's writing to be clear and entertaining, and I was not disappointed by his description of HTML5. The Web version of his book has the advantage of being able to demonstrate the features alongside the description, letting you see what your browser can do, with graphical depictions of what other browsers would show instead. If you are a Web developer of any sort, you should read Pilgrim's book, and then bookmark it as a reference to which you will turn many times during the coming years.

Drawing on the deserved praised for Douglas Crockford's *JavaScript: The Good Parts*, O'Reilly is publishing more books of this type. Crockford's volume is still the biggest bang for the buck, in no small part because he manages to tell you what parts of JavaScript you should avoid. But another book in this series, *HTML & CSS: The Good Parts* by Ben Henick, is readable, interesting and has given me the confidence to (once again) try to use CSS for something more advanced than basic boxes.

A surprisingly interesting book, also published by O'Reilly, is *Web Reputation Systems* by Randy Farmer and Bryce Glass. My dissertation software includes a rating system, so I was interested in what these authors had to say on the subject. I didn't expect there would be enough to fill an entire book, but I found that they provided interesting food for thought, as well as practical considerations for the implementation and incorporation of rating systems into a Web site. If you're considering the inclusion of such a rating system into your Web application, this book is worth at least a look.

Podcasts and Screencasts

Screencasts have become an increasingly popular way to deliver tutorials and information. Because everyone can see the speakers' screens, it's possible to follow along as they program, show diagrams and even make mistakes. I've become quite the fan of screencasts and often watch them to better understand a problem or learn a new topic.

Ryan Bates has been producing his excellent, weekly and free "Railscasts" series for several years now. I try to watch every one of them. Particularly as I start to make the transition from Rails 2 to Rails 3, "Railscasts" comes in handy, showing me where I will need to change my code and where I can leave it as is.

The weekly "Rails Envy" podcast is no more, but Gregg Pollack and his colleagues at Envy Labs now are producing a twice-weekly podcast called "Ruby5", which tries to summarize the latest Ruby and Rails news in less than five minutes. It has more of a newsy feel than "Rails Envy" did, and we no longer get the cheesy (but funny, in my opinion) sound effects and music, but the information is up to date and solid, and it's delivered with a large degree of thought and clarity.

Particularly as I start to make the transition from Rails 2 to Rails 3, "Railscasts" comes in handy, showing me where I will need to change my code and where I can leave it as is.

For me, the biggest surprise of the year was my discovery of Bruce Momjian's Webinars (then turned into screencasts/movies) about PostgreSQL. If you work with PostgreSQL, I'm sure you'll get something out of these excellent, carefully written and clearly explained seminars. Momjian works for EnterpriseDB, a PostgreSQL consulting and support company. His screencasts are available free of charge from **enterprisedb.com**.

Non-Computer Books

I don't spend all of my time reading about computers, although it might seem that way to anyone who sees the stacks of books in my home office. In this space, I mention a few of my favorite non-programming books from the past year, although you'll see that there's still a technical bent to most of them.

Perhaps the most computer-related book in this section is *Coders at Work* by Peter Siebel, published by Apress. If you read and enjoyed *Founders at Work*, in which the author interviews the founders of many startups, you likely will enjoy *Coders at Work*, which interviews many impressive and famous programmers. It's always interesting to hear what some of these programmers have to say and learn from the insights they have gained during their years of software development.

Everyone in my family (including my children) really enjoys cooking. I often have said that one of my reasons for cooking is that it lets me perform science experiments with edible results. For this reason, one of the most important books in our house (often referred to as "The Book") is the 2nd edition of Harold McGee's *On Food and Cooking*, a seemingly endless collection of history, facts and interesting tidbits about the science of food and cooking. Well, if McGee's book provides the science, a new book published by O'Reilly, *Cooking for Geeks*, provides the engineering, offering practical advice, recipes, interviews with cooks and a wealth of techniques you can use right away. (I've already stopped peeling garlic before putting it into the press. Who knew?) If you enjoy cooking, or just want to get into it, I strongly recommend this book.

Another non-computer O'Reilly book is *Your Money: The Missing Manual*. *The Missing Manual* series, started by *New York Times* columnist David Pogue, tries to provide reference guides for users who need a bit of hand-holding, but who aren't going to pay or wait for support. Recently, the *Missing Manual* series has branched out into non-computer topics, including *Your Body* (a fascinating introduction to human biology) and the realm of personal finance. Living outside the United States, many aspects of the book weren't relevant to me. However, the general attitude of the book was a welcome one, especially for those of us who aren't as disciplined as we should be when it comes to our money. I wouldn't say that the book changed my life, but it did give me many new ideas about how to get a better handle on my family's finances, which is good for all of us.

Conclusion

Amazon recently announced that it's selling more ebooks than hardcover books, and I can believe it. There is a compelling argument to be made for ebooks, although I prefer the paper versions when I need to concentrate or just for convenience. Regardless of the medium you use to read, let me suggest to readers of this column what I tell my own young children, that the important thing is just to read and keep learning new things all the time. This year's crop of new books, of which I barely scratched the surface here, should provide enough food for thought for at least another year, and probably well beyond that.■

Reuven M. Lerner is a longtime Web developer, architect and trainer. He is a PhD candidate in learning sciences at Northwestern University, researching the design and analysis of collaborative on-line communities. Reuven lives with his wife and three children in Modi'in, Israel.

MPLS for the masses

$39.95

Usually MPLS routers cost more than $1000, but not anymore. MikroTik gives you the ability to use MPLS in any network. No more big box prices for MPLS! A chicken in every pot!

MPLS stands for Multi Protocol Label Switching. It can be used to replace IP routing - packet forwarding decision is no longer based on fields in IP header and routing table, but on labels that are attached to the packet.

MPLS makes it easy to create "virtual links" between nodes on the network, regardless of the protocol of their encapsulated data. It is a highly scalable, protocol agnostic, data-carrying mechanism. MPLS allows one to create end-to-end circuits across any type of transport medium, using any protocol.

Features:

- Label Distribution Protocol for IPv4
- Virtual Private Lan Service
  * VPLS LDP signaling
  * VPLS MP-BGP based autodiscovery and signaling
  * split-horizon bridging
- RSVP TE Tunnels
  * explicit paths
  * CSPF path selection
  * OSPF extensions for TE tunnels
- Virtual Routing and Forwarding
- MP-BGP based MPLS IP VPN
- OSPF and RIP as CE-PE protocols

Benefits:

- higher speed forwarding in network core
- ability to implement transparent L2 and L3 VPNs (VPLS & VRF)
- reduced VPN overhead compared to legacy tunneling solutions
- traffic engineering to implement QoS and optimize network usage
- ability for the ISP to create VPNs without user interaction
- separate tunnels for voice, video, or data

All MikroTik RouterBOARDs support MPLS, including the **RB750** which costs $39.95 The RB750 is a SOHO router with a 400MHz Atheros CPU, five ethernet ports, plastic case and PSU. With MPLS, RB750 is capable of wire speed throughput for 1000byte packets and up, maximum 80000 pps with smaller packets.

MikroTik routerboard

www.mikrotik.com/mpls

DAVE TAYLOR

Scripting Common File Rename Operations

If you find yourself always typing the same set of commands, it's time to write a script. This month, it's a script to rename and renumber files.

I'm guessing that each of us uses the command line differently and seeks to accomplish different tasks. Mine are sometimes very specialized, like the script I wrote that lets me easily transform the unique filenames from the Mac OS X built-in screen-capture utility into a Web-friendly format.

In the past few weeks, I realized I needed another fairly specialized script for file renaming, but this time, I wanted to write something as generally useful as possible.

There's already a utility included in some flavors of Linux called rename, but, alas, I couldn't find it on my Linux/NetBSD systems. If you have it, it probably duplicates the functionality I create this month. Still, read on. Hopefully, this'll be useful and interesting!

Rename/Pattern/Newpattern

It's surprising how often I find myself on the command line typing in something like:

```
for name in xx*
do
    new="$(echo $name | sed 's/xx/yy/')"
    mv $name $new
done
```

So, that's the first part of the script I want to create, one that lets me just specify the OLD and NEW filename patterns, then simply renames all files matching "OLD" with the "NEW" pattern substituted.

For example, say I have test-file-1.txt and test-file-2.jpg and want to replace "test-file" with "demo". The goal is to have an invocation like:

```
rename test-file demo
```

and have it do all the work for me. Sound good?

How Many Matching Files?

The first step is actually the most difficult: matching an arbitrary pattern and catching any possible error conditions gracefully. The loop is going to end up looking like this:

```
for name in $1*
```

If there aren't any matches, however, you get an ugly error message and the script looks amateurish. So, the goal is actually to ascertain before the for loop how many matches there are to that given pattern.

Ah, okay, so `ls $1* | wc -l` does the trick, right? Nope, that'll still generate the same ugly error message.

Fortunately, there's a way in Bash that you can redirect stderr to go to stdout (that is, to have your error messages appear as standard messages, able to be rerouted, piped and so on).

The test for the number of matches, thus, can be done like this:

```
matches="$(ls -1 $1* 2>&1 | wc -l)"
```

I know, it's complicated. Worse, a quick test reveals that when there are zero matches, `ls -l` actually generates an error message: `ls: No such file or directory`. That's not good. The solution? Add a grep to the sequence:

```
matches="$(ls -1 $1* 2>&1 | grep -v "No such file" | wc -l)"
```

That's even more complicated, but it works exactly as we'd like. "matches" is zero in the situation where there aren't any matches; otherwise, it has the number of matching files and folders for the given pattern.

A test now lets us produce a meaningful and informative error message:

```
if [ $matches -eq 0 ] ; then
    echo "Error: no files match pattern $1*"
    exit 0
fi
```

Because we're looking at stderr versus stdout, we also could more properly route that error message to stderr with >&2, and to be totally correct, we should exit with a nonzero error code to indicate that the

The first step is actually the most difficult: matching an arbitrary pattern and catching any possible error conditions gracefully.

script failed to execute properly. I'll leave those tweaks as an exercise for the reader.

Now that we know we'll never hit the for loop without at least one match, the core code is straightforward:

```
for name in $1*
do
     new="$(echo $name | sed "s/$1/$2/")"
     mv $name $new
done
```

Notice in this instance that you can't use the single quotes within the `$( )` command substitution; if you do, $1 and $2 won't be expanded properly.

We certainly could just stop here and have a useful little script, but I'm into wicked cool scripts, so let's push on, shall we?

Sequential File Numbering

The other feature I constantly find myself needing is the ability to number a series of files sequentially. For example, a final set of photos from a photo shoot might be DSC1017, DSC1019, DSC1023 and DSC1047. It would be more useful to be able to renumber those before sending them to a client, so that they're DSC-1, DSC-2, DSC-3 and so on.

This is pretty easily accomplished too, now that we have a script that renames a sequence of files. Here's how I accomplish it in the script itself:

```
if [ $renumber -eq 1 ] ; then
    suffix="$(echo $name | cut -d. -f2- | tr '[A-Z]' '[a-z]')"
    new="$2$count.$suffix"
    count=$(( $count + 1 ))
    mv $name $new
    chmod a+r $new
fi
```

Here I am expecting to replace the entire filename, so I strip out and save the filename suffix (for example, DSC1015.JPG becomes JPG), so I can re-attach it later. While I'm at it, filename suffixes also are normalized to all lowercase using the handy tr command.

The count variable keeps track of what number we're on, and notice the built-in shell notation of `$(( ))` for mathematical calculations.

Finally, the new filename is built from the new pattern ($2), plus the count ($count), plus the filename suffix ($suffix) in this line:

```
new="$2$count.$suffix"
```

The two conditions need to be merged, however, so the final script ends up with an if-then-else-fi structure.

I can't leave well enough alone, so I continued to

The other feature I constantly find myself needing is the ability to number a series of files sequentially.

tweak the script by adding a few starting flags too. To parse it all, our friend getopt is utilized:

```
args=$(getopt npt $*)

if [ $? != 0 -o $# -lt 2 ] ; then
    echo "Usage: $(basename $0) {-p} {-n} {-t} PATTERN NEWPATTERN"
    echo "
    echo " -p  rewrites PNG to png"
    echo " -n  sequentially numbers matching files with"
    echo "     NEWPATTERN as base filename"
    echo " -t  test mode: show what you'll do, don't do it."
    exit 0
fi

set -- $args
for i
do
    case "$i" in
    -n ) renumber=1 ; shift ;;
    -p ) fixpng=1   ; shift ;;
    -t ) doit=0     ; shift ;;
    -- ) shift      ; break ;;
fi
```

I've written about getopt and its complicated usage in shell scripts before if you want to read up on it [see "Parsing Command-Line Options with getopt" in the July 2009 issue of *LJ*, **www.linuxjournal.com/article/10495**]. Note that three flags are available to the script user: `-n` invokes the renumbering capability (which means the filenames are discarded, remember); `-p` is a special case where .PNG also is rewritten as .png; and `-t` is a sort of "echo-only" mode where the rename doesn't actually happen, the script just shows what it would do based on the patterns given.

How am I using it now? Like this:

```
rename -n IMG_ iphone-copy-paste-
```

Every matching .PNG file (IMG_*) has that portion of its name replaced with "iphone-copy-paste-", and as it proceeds, "PNG" is also rewritten as "png".

The entire rename script can be found on the *Linux Journal* FTP server at **ftp.linuxjournal.com/pub/lj/listings/issue199/10885.tgz**.■

Dave Taylor has been hacking shell scripts for a really long time, 30 years. He's the author of the popular *Wicked Cool Shell Scripts* and can be found on Twitter as @DaveTaylor and more generally at www.DaveTaylorOnline.com.

MICK BAUER

Building a Transparent Firewall with Linux, Part III

Hack your cheap wireless gateway into a stealth firewall.

In this series of articles, I'm showing how to build a transparent firewall using OpenWrt (Linux) running on an inexpensive Linksys WRT54GL wireless router. In Part I, I explained why firewalls are still important and the difference between a traditional IP firewall and a transparent firewall.

In Part II, I sketched out a simple design for deploying a transparent firewall in a home network setting (probably the best application of any OpenWrt-based firewall). I also showed the step-by-step process by which I replaced the native Linksys firmware on my WRT54GL with OpenWrt Kamikaze (v. 8.09.2, running a Linux 2.4 kernel) and then upgraded it to OpenWrt Backfire (v. 10.03, running a Linux 2.6 kernel).

This month, I recompile and configure OpenWrt Backfire, hopefully the last major OpenWrt-specific task covered in this series. Next time, I'll begin writing a custom iptables firewall script, which will apply to *any* Linux system you want to use as a transparent firewall.

Before diving back in, a quick note on OpenWrt performance: OpenWrt is a hobbyist's distribution, and it runs on cheap hardware with less RAM and slower processors than any modern Linux desktop system. I'm writing about it because it's fun to play with, and because I've long wanted to do some hardware hacking in this column. OpenWrt is not, however, a good choice if you need a firewall that is either very fast or very stable.

Recompiling the OpenWrt Kernel

Before configuring OpenWrt, you need to recompile it. That is, you need to recompile the Linux 2.6 kernel in Backfire so that iptables can run in bridging mode, rebundle the kernel into a new firmware image and re-flash that to your gateway. This is less work than it probably sounds like.

The OpenWrt build process has some prerequisites. First, you need all of these Ubuntu packages (or your distribution's equivalents): gawk, gcc, binutils, patch, bzip2, flex, bison, make, gettext, pkg-config, unzip, libz-dev, libcheaders and subversion.

If you've compiled a Linux kernel before, your system may have most of these already; on mine, I needed to install only gawk, flex, bison, subversion and gettext.

Next, you need 3.5GB of free disk space on a non-Windows-formatted volume (msdos, fat32 and ntfs don't support Linux user/group-ownerships and permissions). I don't know why so much space is necessary to compile a firmware image for devices with only 4MB of RAM, but if you run out of disk space during the compile, you'll get strange, cryptic error messages.

The compile process is time consuming but simple. As a nonroot user, change your working directory into your 3.5GB-free volume, and execute this sequence of commands:

```
backfireimage-$ svn co svn://svn.openwrt.org/openwrt/branches/backfire
```

This fetches the source code tree for the current version of OpenWrt into your working directory. Now, enter that source code tree:

```
backfireimage-$ cd ./backfire
```

By adding this line to your kernel configuration, you make iptables able to operate in bridging mode—that is, to control packets traversing a local bridge device:

```
backfireimage/backfire-$ echo "CONFIG_BRIDGE_NETFILTER=y" >>
➥./target/linux/brcm47xx
```

Now, rebuild the entire OpenWrt firmware image—the Linux 2.6 kernel, all system commands and the compressed RAM filesystem on which they reside:

```
backfireimage/backfire-$ make
```

This one `make` command takes quite a long time, depending on how fast your CPU and hard disk are. If it ends prematurely due to errors, the likeliest causes are either that you're missing a required package or you don't have enough free disk space.

If your build fails for some other reason, or if you simply can't tell, try again with this command:

```
backfireimage/backfire-$ make V=99
```

Setting make's verbosity to 99 in this way causes it to output a very large quantity of log messages. If you end up seeking help on the OpenWrt Forums (**https://forum.openwrt.org**), including some of these log messages will improve your odds at receiving a useful answer.

Once the build completes successfully, you can change your working directory to that in which the new binary firmware images reside. Since I'm working with a Linksys WRT54GL, which uses a Broadcom chipset, and since I'm installing a Linux 2.6 kernel, the binaries I want are in bin/brcm47xx:

```
backfireimage/backfire-$ cd bin/brcm47xx
```

Now it's time to reboot the WRT54GL and re-flash its firmware. Immediately after turning your router's power off and then back on, or issuing the command `reboot` from a telnet session, enter this command to push the new image from your build system:

```
backfireimage/backfire-$ tftp -m binary 192.168.1.1 -c
 ➥put openwrt-wrt54g-squashfs.bin
```

Listing 1. Changing Time Zone and Hostname

```
root@OpenWrt# uci set
 ➥system.@system[0].timezone=CST6CDT,M3.2.0,M11.1.0
root@OpenWrt# uci set system.@system[0].hostname=sugartongs
root@OpenWrt# uci commit system
```

As you may recall from last time, OpenWrt's default IP address is 192.168.1.1. On the laptop from which I'm connecting to my broadband router, I've configured the Ethernet interface with an IP address on the same network (192.168.1.30, netmask 255.255.255.0).

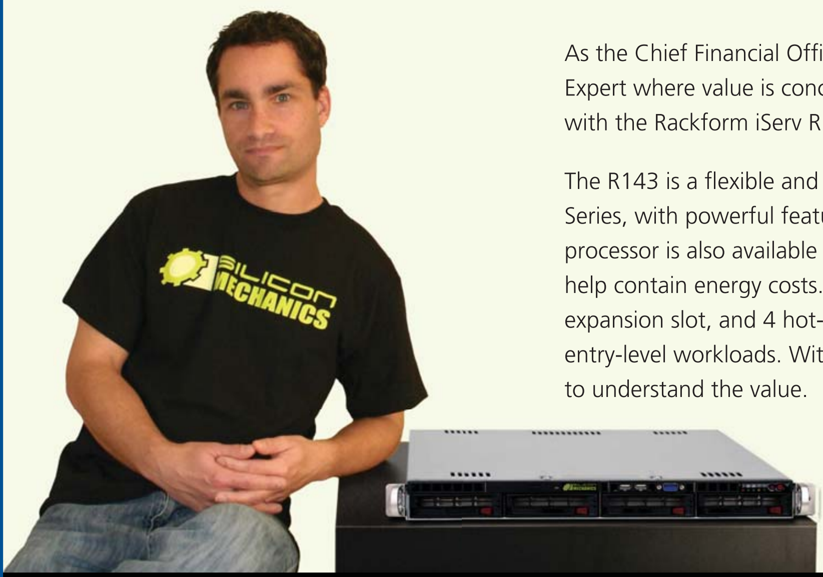
It may take a few reboot/TFTP attempts for your broadband router to "see" the TFTP push, but once it does, and after it decompresses and loads the new firmware, your router will be capable of acting as a transparent firewall! But, first you've got to do some system-level configuration.

Enabling SSH

Both examples I showed last month for how to connect to OpenWrt involved telnet. Although this is the

SILICON MECHANICS

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



intel Xeon inside™

Powerful. Intelligent.

As the Chief Financial Officer for Silicon Mechanics, Steve is an Expert where value is concerned. That's why he's pictured here with the Rackform iServ R143.

The R143 is a flexible and affordable 1U server. It features an Intel® Xeon® Processor 3400 Series, with powerful features like Turbo Boost, Hyper-Threading, and DDR3 memory. This processor is also available in a low-voltage version, which can optimize power usage and help contain energy costs. With 6 DDR3 DIMM sockets, 2 Gigabit Ethernet adapters, a PCIe expansion slot, and 4 hot-swap SAS/SATA drive bays, the R143 can handle a lot more than entry-level workloads. With a price that starts around $1250, you don't have to be a CFO to understand the value.

When you partner with Silicon Mechanics, you get more than a flexible, affordable entry-level server — you get an Expert like Steve.

For more information about the
Rackform iServ R143
visit www.siliconmechanics.com/R143

Expert included.

default way to log on to OpenWrt (at least for initial setup), it's highly insecure.

Luckily, on OpenWrt Backfire, the Dropbear Secure Shell (SSH) dæmon package is installed and runs at startup, by default. All you have to do to disable telnet logins and enable SSH logins is first to telnet in to OpenWrt and then set a root password via the `passwd` command, like this:

```
root@OpenWrt:~# passwd
Changing password for root
New password: *********
Retype password: *********
Password for root changed by root
```

You don't need to restart the router; simply log out of the telnet session, and `ssh` back in. This time, you'll be prompted for a user name (use "root") and password (the one you just entered).

Now that you've got a secure administrative session, you can get to work reconfiguring OpenWrt using the Unified Configuration Interface (uci) system.

Using uci

In earlier versions of OpenWrt, such as White Russian, you had to manage two different configuration systems: NVRAM settings, via the `nvram` command and the standard /etc system for ordinary Linux OS and application settings. With the Kamikaze and Backfire versions of OpenWrt, however, nvram settings are maintained in files stored in /etc/config, making OpenWrt a bit more UNIX-like than before.

In fact, most OpenWrt behavior, not just NVRAM-specific settings, can be managed via files in /etc/config/. The catch is that unlike ordinary configuration files, you're supposed to use the command `uci` rather than a text editor to manipulate anything in /etc/config.

uci automatically decides whether changes in a given /etc/config file need to trigger an NVRAM change, require other commands such as `iptables` to be invoked and so forth. Strictly speaking, you probably don't always *have* to use uci—for example, I was able to change my WRT54GL's time zone by editing /etc/config/system and rebooting. However, things work better on OpenWrt when you stick to uci.

Listing 1 shows a block of uci commands with which you can change your OpenWrt box's time zone and hostname.

The general syntax of the uci command is `uci [action] [config-file-name].[config-file-section].[option-name]=[value]`. Thus, the first line in Listing 1 translates to "change a setting in /etc/config/system, in its system section, called timezone, to have the value CST6CDT,M3.2.0,M11.1.0".

Why does a time zone value have so much gobbledygook after the name of the actual time zone? Why not just say "CST6CDT"? It's because of the difference in Daylight Savings Time start and end dates in different countries. See Resources for a link to a chart of different time zone strings you can use.

Setting the correct time zone is important. It allows your OpenWrt Backfire system to synchronize its time over the Internet automatically, using the `rdate` command (or you can install ntpclient to have it use ntp instead). If you don't set the correct time zone, rdate won't work correctly, which means lots of other things will fail too, such as IPsec and anything else that uses digital certificates.

Moving on, the second line in Listing 1 involves changing the setting of option "hostname" from its default of "OpenWrt" to "sugartongs". Obviously, you can specify whatever hostname you like.

The third line tells uci to commit all changes to /etc/config/ since the last time it was run—that is, to change NVRAM, execute iptables commands and so forth, as applicable. I find that with time zone and hostname settings, however, you also need to reboot the router for the changes to take effect (using the command `reboot`, naturally).

OpenWrt Documentation

In my opinion, complete documentation is not one of OpenWrt's strengths. This is a Linux distribution by and for network engineers, and its maintainers assume OpenWrt users have a higher-than-average ability and willingness to figure things out on their own.

For the rest of us, some useful OpenWrt documentation is found *not* in the OpenWrt home page's Documentation area, but within the OpenWrt Wiki at **wiki.openwrt.org**. Sooner or later you'll also probably need to use the Forum (**https://forum.openwrt.org**) or even Google to find answers to your OpenWrt-related questions.

Installing Optional Packages

I'll come back to uci in a moment. First, a here's quick word about optional software packages.

Like any Linux distribution, OpenWrt has optional software packages you can install after the base system image is in place. The majority of OpenWrt's packages are network-oriented, and they include apache, bind, freeradius, various Linux kernel modules, snort, squid, stunnel, vpnc and vsftpd.

But these are out of scope for this series of articles. Everything you need in order to build a transparent firewall using OpenWrt Backfire is included in the base image (at least that was true for my Linksys WRT54GL). Furthermore, most broadband routers have between 16 and 72 megabytes *total* combined Flash-memory and RAM; even with a compressed filesystem, this doesn't amount to much storage space either for applications themselves or for their data.

Still, if you want to install optional packages, they're available from openwrt.org in the packages directory of your architecture's download site. For example, for my system, running the Broadcom 47xx version of OpenWrt Backfire, optional packages are located in **backfire.openwrt.org/10.03/brcm47xx/packages**. See the OpenWrt Wiki page for Packages for more information on finding and managing OpenWrt packages.

Changing Network Configuration on OpenWrt

The last task to cover this month, and (I hope) the last OpenWrt-specific one, is configuring networking on OpenWrt. This is a huge topic I can't cover in depth (though I think my examples are pretty clear). There's a more complete explanation of the many different ways you can configure networking on OpenWrt on the OpenWrt Wiki (**wiki.openwrt.org/doc/uci/network**).

Speaking of which, you'll notice that the wiki article on configuring networking doesn't list any actual uci commands; it shows only the "finished product", /etc/config/network. This is because statements in OpenWrt's /etc/config files are easy to translate into uci commands once you understand the syntax.

I recommend you first sketch out on paper what you want /etc/config/network to look like, compare that to how /etc/config/network presently looks, note which lines need to change, and then translate those changes into a list of commands. It *is* somewhat easier to make mistakes when entering a long sequence of commands than it is by simply editing a configuration file. However, using the up-arrow key to call up the command you just entered, and then backspacing over the part that's different in the next command in the sequence, reduces the amount of typing you need to do and, therefore, your potential for messing up.

Listing 2 shows Backfire's default /etc/config/network file for a Linksys WRT54GL.

Let's work our way down this file, from the top. First, `config switch eth0` and `option enable 1` constitute the first configuration section. Each section consists of a config line that names some network interface, switch, vlan or other element, followed by one or more option lines that affect that element. The first section in Listing 2 enables the switch device eth0.

As is typical for broadband routers, all the Ethernet ports on a Linksys WRT54G are controlled by a single switch chipset (a Broadcom chipset in this case). Individual ports are referred to in OpenWrt by their Virtual LAN (VLAN) assignment, for example, eth0.0 (VLAN #0), eth0.1 (VLAN #1) and so forth. Subsequent configuration sections in Listing 2 define and configure those VLANs.

Listing 2. Default /etc/config/network File

```
config switch eth0
        option enable   1

config switch_vlan eth0_0
        option device   "eth0"
        option vlan     0
        option ports    "0 1 2 3 5"

config switch_vlan eth0_1
        option device   "eth0"
        option vlan     1
        option ports    "4 5"

config interface loopback
        option ifname   "lo"
        option proto    static
        option ipaddr   127.0.0.1
        option netmask  255.0.0.0

config interface lan
        option type     bridge
        option ifname   "eth0.0"
        option proto    static
        option ipaddr   192.168.1.1
        option netmask  255.255.255.0

#### WAN configuration
config interface        wan
        option ifname   "eth0.1"
        option proto    dhcp
```

The second section in Listing 2 shows the settings for VLAN 0, called `eth0_0` within the config file but subsequently known to the kernel as `eth0.0`. `option device`, specifies which switch the VLAN is associated with (eth0 being the only switch present); `option vlan` specifies the VLAN number you want to define; and `option ports` specifies which ports belong to this VLAN.

Port numbers are assigned differently from what you'd expect in OpenWrt. Port #5 is a "virtual" port associated with the kernel itself. Every VLAN must be associated with port #5. Port #4 on the WRT54GL is labeled as the "WAN port", the port you customarily connect to your DSL router or cable modem (though in OpenWrt you can assign it to whatever VLAN you like—there's actually nothing special about it).

Ports 0–3 correspond to ports 4, 3, 2 and 1, respectively, on the WRT54GL; they're numbered *backward* relative to the screen printing on the box. Therefore, in Listing 2, `option ports "0 1 2 3 5"` means "ports 4, 3, 2 and 1" and `option ports "4`

Listing 3. New /etc/config/network File

```
config 'switch' 'eth0'
        option 'enable' '1'

config 'switch_vlan' 'eth0_0'
        option 'device' 'eth0'
        option 'vlan' '0'
        option 'ports' '0 1 2 3 4 5'

config 'interface' 'loopback'
        option 'ifname' 'lo'
        option 'proto' 'static'
        option 'ipaddr' '127.0.0.1'
        option 'netmask' '255.0.0.0'

config 'interface' 'lan'
        option 'type' 'bridge'
        option 'ifname' 'eth0.0'
        option 'proto' 'static'
        option 'ipaddr' '10.0.0.253'
        option 'netmask' '255.255.255.0'
```

So, what changes do I want to make? Basically, I want to configure the *entire* block of Ethernet interfaces as a single switch, including the WAN address.

5" would translate to "the WAN port".

Therefore, it follows that the next section defines the WAN port as belonging to VLAN #1 (eth0.1).

Don't worry about the `loopback` section. If you know anything about loopback interfaces, this section's meaning is obvious. If you don't, it doesn't matter because you never should change this section anyhow. Suffice it to say that as on any other Linux system, packets sent to either IP address 127.0.0.1 or interface lo will go directly to the local kernel and be flagged as having originated locally.

The last two sections define network interfaces. The first interface section is arbitrarily named lan, its actual interface name (ifname) being eth0.0 (which as you'll recall is actually VLAN 0 on the switch eth0). Since it concerns switch ports, it's of the type switch. This interface has a static IP address of 192.168.1.1, with a netmask of 255.255.255.0.

As you'll recall from the second section of Listing 2, eth0.0 is associated with ports 1–4 on the router. This VLAN is now being defined as a standalone Ethernet switch whose IP address is 192.168.1.1/24 (implying that anything connecting to ports 1–4 needs to have an IP address in the same logical LAN).

In the second interface section, VLAN 1 (eth0.1), which is associated only with the router's WAN port (OpenWrt port 4), is configured as a standard (nonbridge) interface with a dynamic, dhcp-assigned IP address.

So, what changes do I want to make? Basically, I want to configure the *entire* block of Ethernet interfaces as a single switch, including the WAN interface. I also want to change the switch's IP address (and also its network address) to 10.0.0.253, an unused IP address on the network into which I plan to insert the firewall.

Since all bridge ports will be on the same VLAN, I can delete the entire VLAN eth0_1 section and the entire interface wan section. I'll also have to change the ports option under VLAN eth0_0 and the ipaddr option under interface lan. All this will take a total of only seven commands (including rebooting the system)!

Listing 3 shows the way you want /etc/config/network to look.

Listing 4 shows the seven commands necessary to transform /etc/config/network, as shown in Listing 2, into that shown in Listing 3. Before executing these, however, *please* read the explanatory text that follows, which will help you avoid the risk of bricking (rendering useless) your broadband router.

I'm out of space for this month, so I can't dissect Listing 4, which is hopefully similar enough to the previous uci examples to make sense. I will, however, leave you with two important notes.

First, note the `uci show network`. This allows you to check your work before committing changes. If any line is wrong, you can re-enter the relevant uci command. To start over, enter the command `uci revert network` to undo *all* changes. If you

Listing 4. uci Commands to Change /etc/config/network

```
root@OpenWrt:~# uci set network.eth0_0.ports="0 1 2 3 4 5"

root@OpenWrt:~# uci delete network.eth0_1

root@OpenWrt:~# uci set network.lan.ipaddr="10.0.0.253"

root@OpenWrt:~# uci delete network.wan

root@OpenWrt:~# uci show network

root@OpenWrt:~# uci commit

root@OpenWrt:~# reboot
```

mess things up so badly you can't `ssh` back in, you can re-flash the firmware image, which among other things will reset the router's IP address back to 192.168.1.1. Checking and rechecking your work before committing, however, is less work and easier on your nerves than re-flashing!

Second, after changing your device's IP address and rebooting, you won't be able to reconnect to your OpenWrt box until you've reconfigured your client system with an IP address compatible with the OpenWrt box's new address. For example, after I reconfigured my Linux laptop's Ethernet interface with the IP address 10.0.0.30 and netmask 255.255.255.0, I was able to `ssh` back in to my OpenWrt router with the command `ssh root@10.0.0.253`.

Conclusion

I've covered a lot of ground this month: recompiling OpenWrt for iptables bridging support, enabling SSH, using uci and reconfiguring networking. Next time, I'll show you how to disable the default OpenWrt firewall and create a custom iptables script that should work on *any* bridging-aware Linux 2.6 system. Until then, 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

Home Page for the OpenWrt Project: **www.openwrt.org**

OpenWrt's Unified Configuration Interface Documentation: **wiki.openwrt.org/doc/uci**

Chart of Time Zone Strings: **nuwiki.openwrt.org/oldwiki/openwrtdocs/whiterussian/configuration#timezone**

OpenWrt Software Package Information: **wiki.openwrt.org/oldwiki/openwrtdocs/packages**

Powerful: Rhino

Rhino M6500/E6500
- Dell Precision M6500 w/ Core i7 Quad (8 core)
- Dell Latitude E6500 w/ 2.2-3.0 GHz Core 2 Duo
- Up to 17" WUXGA LCD w/ X@1920x1200
- NVidia Quadro FX 3800M
- 80-500 GB hard drive
- Up to 16 GB RAM (1333 MHz)
- DVD±RW or Blu-ray
- 802.11a/g/n
- Starts at $1240

- High performance NVidia 3-D on a WUXGA RGB/LED
- High performance Core i7 Quad CPUs, 16 GB RAM
- Ultimate configurability — choose your laptop's features
- One year Linux tech support — phone and email
- Three year manufacturer's on-site warranty
- Choice of pre-installed Linux distribution:

Tablet: Raven

Raven X200 Tablet
- ThinkPad X200 tablet by Lenovo
- 12.1" WXGA w/ X@1280x800
- 1.2-1.86 GHz Core 2 Duo
- Up to 8 GB RAM
- 80-500 GB hard drive / 256 GB SSD
- Pen/stylus input to screen
- Dynamic screen rotation
- Starts at $2080

Rugged: Tarantula

Tarantula CF-30
- Panasonic Toughbook CF-30
- Fully rugged MIL-SPEC-810F tested: drops, dust, moisture & more
- 13.3" XGA TouchScreen
- 1.6 GHz Core 2 Duo
- Up to 8 GB RAM
- 80-500 GB hard drive
- Call for quote

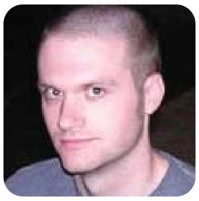
EmperorLinux

...where Linux & laptops converge

www.EmperorLinux.com

1-888-651-6686

Model specifications and availability may vary.



Some Hacks from DEF CON

KYLE RANKIN

What happens when a sysadmin goes to a hacker conference? He gets a fresh perspective on computer security.

I don't think the timing could be better on the editorial calendar than to have the deadline for the Hacking issue coincide with the Black Hat and DEF CON conferences. Although I have long been interested in security, usually it's from the sysadmin defensive posture or from a post-hack forensics angle. This year, I was fortunate enough to look over the shoulder of a talented group of hackers from the neg9 team as they competed in an open CTF tournament. Essentially, the tournament pitted multiple teams against each other as they all attempted to solve various puzzles and security challenges against locked-down servers. I thought for this column, I would list some tips for the defensive sysadmin that came to mind as I viewed computer security from an offensive perspective, especially in a restricted environment like a hacking competition.

If you want to attack a system but only know how to edit text files in Emacs, port scan with Nmap or write Ruby scripts, what do you do when those tools aren't installed?

First, before I get too many angry letters, I'm well aware of the long controversy over the word "hacker" and its multiple connotations. I refer to both clever tricks and security breaches as hacks and their perpetrators as hackers in this article for a few reasons:

1. The English language has many words that can change meanings based on context, and I think everyone reading this article is intelligent enough to make the appropriate distinctions.

2. Our community already is capable of drawing the distinction between "hack" when it refers to either a prank, an elegant solution, a quick-and-dirty kludge, or even a writer or politician based on context, so I think we can handle an extra contextual definition.

3. The word "cracker" just reminds me of 1970s lingo for white people, and it's hard for me to keep a straight face when I hear it in either context.

Learn Classic Linux Tools

Now that that's out of the way, the first thing I found interesting about security from the offensive position was just how important it is to learn basic, classic command-line tools like vi, nc and friends. In this competition and in a proper locked-down system, you not only run minimal services, you also try to restrict the set of programs you install on the machine so they won't be available to a would-be attacker. If you want to attack a system but only know how to edit text files in Emacs, port scan with Nmap or write Ruby scripts, what do you do when those tools aren't installed? Even in a restricted environment, you can count on certain tools to be installed, such as vi, nc, dd, sh and all the other great two-letter commands. From the offensive perspective, if you know how to use those commands, you won't be hamstrung if you do happen on a minimal system. From the defensive perspective, if you do limit the programs installed on your system only to those that are necessary, you *will* slow down hackers who expect newer, fancier tools to be on the system.

I think if you were going to master only one of these two-letter commands for hacking purposes besides vi, nc is the best candidate. If you are unfamiliar with nc (or netcat), it is an incredibly versatile tool that allows you to open or listen for TCP and UDP connections. It's the original network Swiss Army knife, and it's a valuable tool to have in your arsenal whether you're a sysadmin or a hacker. In the case of both hacking and troubleshooting, it's useful because you can use it like telnet to connect to a remote server and port and start an interactive session:

```
$ nc mail.example.org 25
220 mail.example.net ESMTP Postfix
. . .
QUIT
```

Netcat as a Simple Chat Service

You also could open one nc session on a port in listen mode and start a second nc session on a remote host to connect to that port and send text back and forth like a basic chat program. On the listening host, run:

```
$ nc -l 31337
```

On the remote host, type:

```
$ nc hostname 31337
```

You also can substitute the IPs for hostnames in both examples. Once the connection is made, anything typed on one end is displayed on the other, and you can press Ctrl-D in either session to close the connection.

Netcat for File Transfers

A number of sysadmins have long used this functionality as a quick-and-dirty file-transfer protocol. Start the first nc session in listen mode, and redirect its output to a file:

```
$ nc -l 31337 > output_file
```

On the remote machine from which you want to send the file, you would type:

```
$ nc hostname 31337 < input_file
```

Once the file has finished transferring, the connection will close automatically.

Netcat as a Port Scanner

Another incredibly useful function of nc is as a port scanner when something more sophisticated isn't around. Just use the -z option to have nc test only whether a port is open instead of connecting to it, add -v for verbose output, and provide a port range as arguments. So to scan a host for open ports between 20 and 25 (good for testing for open FTP, telnet, SSH and SMTP services), you would type:

```
$ nc -zv host.example.org 20-25
nc: connect to host.example.org port 20 (tcp) failed:
 ➥Connection refused
Connection to host.example.org 21 port [tcp/ftp] succeeded!
Connection to host.example.org 22 port [tcp/ssh] succeeded!
nc: connect to host.example.org port 23 (tcp) failed:
 ➥Connection refused
nc: connect to host.example.org port 24 (tcp) failed:
 ➥Connection refused
Connection to host.example.org 25 port [tcp/smtp] succeeded!
```

SILICON MECHANICS

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

intel Xeon inside™

Powerful. Intelligent.

Charles heads the web development team here at Silicon Mechanics: he's responsible for the configurators and power calculator on our site, among other things. As a software expert, he offers a useful perspective on our server and storage products.

When asked what he would do if he had a Rackform iServ R413 configured with 4 8-core Intel® Xeon® Processor 7500 Series CPUs and 32 DDR3 DIMMs, he said, "32 virtual machines . . . one per core . . . in one rack unit." But he didn't stop there.

Charles knows that to make the best use of a server with that kind of processing horsepower in a virtualized environment, he needs I/O to match. He paired the 4P server with a Storform iServ R516 storage server, configured with 24 2.5-inch Intel X25-E solid state drives. Think of it as a developer's dream team: multi-core processing and high memory counts for blistering performance, and high-performance storage for blazing I/O speed.

Need a "dream team" of your own? Talk to the Experts at Silicon Mechanics for the perfect match.

When you partner with Silicon Mechanics, you get more than just the power and performance of the latest Intel technologies—you get an Expert like Charles.

For more information about the Rackform iServ R413 visit www.siliconmechanics.com/R413

Expert included.

Write Files on a Restricted Filesystem

Another interesting point at the competition involved a system in which you could log in, however, a number of directories, including the home directory, were mounted over NFS read-only. This presented an interesting set of limitations. For instance, because the machine was being attacked by multiple teams who each could log in as the same user, everyone was kicking each other off the machine. This meant every time you wanted to log in, you had to type in the password manually, yet because the home directory was read-only, you couldn't automate the process with SSH keys. It also meant you needed to be creative with where you stored your scripts, as you couldn't write to the main directory to which your user normally had access.

For the defender, you can see why mounting secure directories as read-only over NFS presents a problem for attackers—attackers can't write to the directory as users, and also even if they become root, they also might have to exploit the NFS server to remount the filesystem in read-write mode. As an attacker, that just meant you needed to be a bit more creative with where you store your files.

The /dev/shm directory is also a hacker favorite and for good reason—it's even less well-known among administrators that it's world-writable.

Temporary Storage

Most command-line users are aware of the existence of the /tmp directory on a Linux system. This is a directory to which all users can write, and special permissions are set on the directory so that although anyone can write to files there, once a file is created, other users can't access it. When hacking this particular restricted system, you could very well store your scripts in /tmp, but that's the first place other teams would notice your files and would likely delete them (bad) or modify them to do something you didn't expect (worse). Luckily, there are at least two other less-well-known directories on a Linux system that are writable by everyone: /var/tmp and /dev/shm.

The /var/tmp directory serves much the same function as /tmp does in that all users and programs can write to it; however, unlike the /tmp directory, many users are unaware that it exists or forget to look there. Also unlike the /tmp directory, which gets erased when the system reboots, any files you put in /var/tmp persist between reboots.

The /dev/shm directory is also a hacker favorite and for good reason—it's even less well-known among administrators that it's world-writable. An even better reason than that for an attacker to use this directory is that its contents actually are stored only in RAM, not on a disk, so once the system reboots, the files are gone and not even forensics techniques can recover them.

Making the Most of root Exploits

At a particular point in the competition, some smart hackers on the neg9 team discovered a fault in a custom service running as root on one of the competition systems. They were able to turn this flaw into a root exploit that allowed them to write 33 bytes as root anywhere on the filesystem they wanted. The tricky part was that if they chose an existing file, it would be truncated with those 33 bytes.

This presents an interesting problem. If your only root access was the ability to write 33 bytes to a file, how would you convert that into full root access? You might, for instance, write a new file into /etc/cron.d/ that executes an exploit script every minute. What would you do though if that doesn't work? When you put yourself in an offensive mindset in a restricted environment like that, you end up discovering creative ways to approach a system.

There were a few ways you could exploit this option to get full root access. You could, for instance, look in /etc/inittab to see what getty process was set to auto-respawn and then replace the existing getty (for instance, /sbin/agetty) with your custom 33-byte shell code. Then, when getty respawns (which you potentially could force), it would respawn as root, and you'd have your custom code running with root permissions.

The way the team exploited the flaw was to write a new short init script that made vim SUID root. Then, they wrote to a file in /proc that forced the system to reboot, so that upon the reboot, their init script would run. With vim set to SUID root, they could edit root-owned files and do things like change root's password and enable root logins with SSH and finally undo the SUID permissions on vim. From a defender's point of view, this kind of attack is quite tricky to protect against and essentially comes down to basic security practices, such as keeping packages up to date and putting services in sandboxes when possible.

All in all, I have to say, I rather enjoyed this different perspective on security. Whether you are a sysadmin who stays on the defensive side or a hacker who thinks in offensive terms, I think both sides can benefit from putting on the other's hat every now and then.■

Kyle Rankin is a Systems Architect in the San Francisco Bay Area and the author of a number of books, including *The Official Ubuntu Server Book*, *Knoppix Hacks* and *Ubuntu Hacks*. He is currently the president of the North Bay Linux Users' Group.

Your Software is the Star...

...The Appliance is the Stage.

Your Software Rocks on Appliances Built by MBX

If you aren't thrilled with the performance of your appliance vendor, let us sway you to become a raving fan of MBX

- ★ 99% On-Time Delivery
- ★ 98.5% Defect-Free Quality Ratio
- ★ Global stocking for next-day delivery anywhere in the world
- ★ Certified Linux engineers designated for each account

If this strikes a chord with you, reach out to MBX

Based on 2009 customer surveys*, 96.6% of customers rate MBX as meeting or exceeding expectations for Overall Value, Quality and Technical Support.

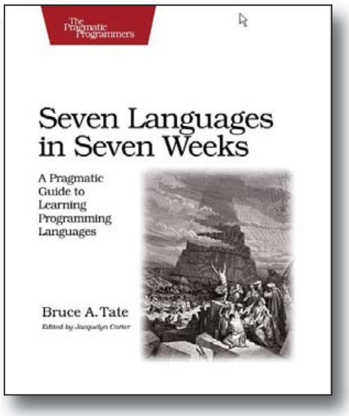
MBX systems

@MBXSystems | www.mbx.com | info@mbx.com

*Cumulative 2009 Quarterly Business Reviews

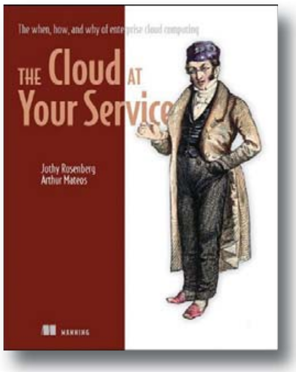
POWERED BY SUPERMICRO®
www.supermicro.com

Bruce A. Tate's *Seven Languages in Seven Weeks* (Pragmatic Bookshelf)


The Pragmatic Programmers
Seven Languages in Seven Weeks
A Pragmatic Guide to Learning Programming Languages
Bruce A. Tate
Edited by Jacquelyn Carter

In any discipline, the road to true mastery and wisdom requires a comprehensive view. If programming is the discipline in which you seek such mastery, crack open Bruce A. Tate's new book *Seven Languages in Seven Weeks, A Pragmatic Guide to Learning Programming Languages*. The book has what publisher Pragmatic Bookshelf calls an "audacious goal", which is to present a meaningful exploration of seven languages—namely Clojure, Haskell, Io, Prolog, Scala, Erlang and Ruby—within a single book. *Seven Languages* covers what's essential and unique about each language and the most critical programming models of our time. For each language, readers solve a nontrivial problem, using techniques that show off the language's most important features, strengths and weaknesses.

www.pragprog.com


The when, how, and why of enterprise cloud computing
The Cloud at Your Service
Jothy Rosenberg
Arthur Mateos
Manning

Jothy Rosenberg and Arthur Mateos' *The Cloud at Your Service* (Manning)

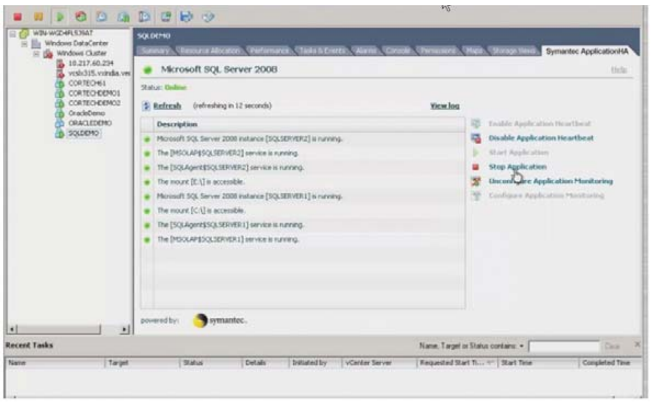
Jothy Rosenberg and Arthur Mateos' penned the new book *The Cloud at Your Service* because the market needed a book on cloud computing targeted at corporate decision-makers. The book, which is subtitled "The when, how, and why of enterprise cloud computing", seeks to inform managers, buyers, enterprise architects, CIOs and CEOs on the what, when and how of moving to the cloud. It is the book the authors wished they had when they were in that role. Cloud-computing-related topics include an introduction, the business case, private clouds, designing and architecting for scale and reliability, practical considerations, deployment and operations and a look into the future of cloud computing.

www.manning.com

STM's Revolution Laptop Backpack

STM, specialists in the toting of precious gear, has released its new Revolution Laptop Backpack line for organized and protected transport of laptops. Each of today's critical elements in the digital lifestyle gets its own space, including a concealed, plush-lined section for the laptop, an integrated pocket for the iPad, a main component for books and paperwork, two versatile side pockets, and an organizational panel for cords, drives, pens and so on. Comfort features include an adjustable suspension system, sternum strap and tuck-away waist straps. There's also a tuck-away rain cover for unexpected downpours. The Revolution is available in two sizes and two colors.

www.stmbags.com



```
Description
Microsoft SQL Server 2008 instance [SQLSERVER2] is running.
The [MSOLAP$SQLSERVER2] service is running.
The [SQLAgent$SQLSERVER2] service is running.
The mount [E:\] is accessible.
Microsoft SQL Server 2008 instance [SQLSERVER1] is running.
The mount [C:\] is accessible.
The [SQLAgent$SQLSERVER1] service is running.
The [MSOLAP$SQLSERVER1] service is running.
```

Symantec's ApplicationHA and VirtualStore

Symantec recently released two new solutions for VMware environments—that is, ApplicationHA and VirtualStore, which the company says "will enable organizations to virtualize their mission-critical applications and minimize their storage costs on the VMware platform." Symantec ApplicationHA, which is based on Veritas Cluster Server technology, provides high availability for business-critical applications through application-level visibility and control in VMware environments. Meanwhile, the Symantec VirtualStore, based on Veritas Storage Foundation technology, is a software-based storage management solution for VMware virtual machines that provides rapid provisioning of servers and virtual desktops, efficient cloning and accelerated bootup of virtual machines. Both ApplicationHA and VirtualStore are integrated seamlessly with VMware management tools, such as VMware vCenter Server, enabling customers to deploy these tools without impact to their operational models.

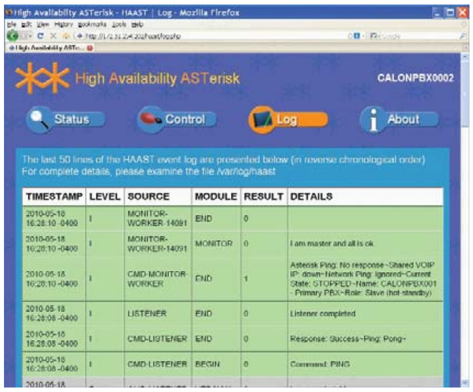
www.symantec.com

NoMachine NX

NoMachine calls its long-awaited NoMachine NX 4.0 software release “the biggest since the day the company was founded” and promises to take “remote access and desktop and application delivery software to an unprecedented level for both physical and virtualized computing environments.” New functionality includes secure remote access to Windows and Mac desktops and applications; browser-based access to NX sessions from anywhere; access from smartphones, video support, recording and playback of NX sessions; USB forwarding to remote devices; a new client GUI and more. Version 4.0 will be guaranteed to all subscribed customers, and a basic free version for personal use is available for download.

www.nomachine.com

Generation D Consulting’s High Availability ASTerisk (HAAST)

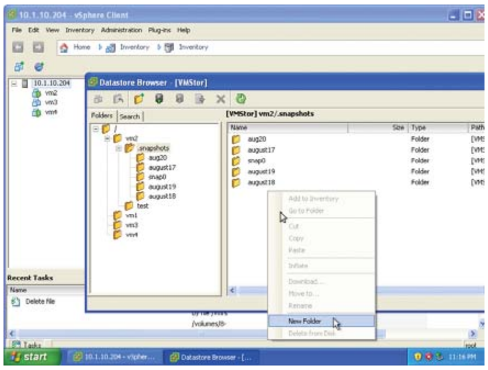

High Availability ASTerisk - HAAST | Log - Mozilla Firefox
High Availability ASTerisk
CALONPBX0002
Status
Control
Log
About

The last 50 lines of the HAAST event log are presented below (in reverse chronological order)
For complete details, please examine the file /var/log/haast

TIMESTAMP	LEVEL	SOURCE	MODULE	RESULT	DETAILS
2010-05-18 16:28:10 -0400	I	MONITOR-WORKER-14091	END	0	
2010-05-18 16:28:10 -0400	I	MONITOR-WORKER-14091	MONITOR	0	I am master and all is ok
2010-05-18 16:28:10 -0400	I	CMD-MONITOR-WORKER	END	1	Asterisk Ping: No response~Shared VOIP IP: down~Network Ping: Ignored~Current State: STOPPED~Name: CALONPBX001 - Primary PBX~Role: Slave (hot-standby)
2010-05-18 16:28:08 -0400	I	LISTENER	END	0	Listener completed
2010-05-18 16:28:08 -0400	I	CMD-LISTENER	END	0	Response: Success~Ping: Pong~
2010-05-18 16:28:08 -0400	I	CMD-LISTENER	BEGIN	0	Command: PING

The developers at Generation D have launched High Availability ASTerisk (HAAST), a software application that allows customers to create cost-effective Asterisk clusters using any pair of off-the-shelf Linux boxes. HAAST is a software application that creates a high-availability, clustered group of Asterisk servers that act as a single server. It can detect a range of failures on any single Asterisk server and automatically transfer control to its mate, resulting in a telephony environment with minimal down time. Key features include rapid failover (as low as seven seconds), detailed logging, Web interface, extendable API, intuitive installation and the ability to leverage low-cost hardware yet achieve a high-availability solution.

www.generationd.com



Gluster’s VMStor

The *raison d’être* of Gluster’s new VMStor is to simplify scalable NAS for virtual machine storage. Gluster’s strategy is eliminating the requirement to use a traditional storage area network (SAN) based environment, which can be expensive to scale. Initially supporting VMware, Gluster VMStor scales to multiple petabytes and can store an unlimited number of VM images. Key capabilities, such as VM-level snapshots and backup, can be performed with a single mouse click or automated with existing tools and cloud management software. Other features include integration with VMware UI and utilization of standard NFS filesystem architecture.

www.gluster.com

Vyatta Network OS

Vyatta calls the new 6.1 release of Vyatta Network OS a “jump forward” due to advances in IPv6 interoperability, cloud-specific features and enhanced enterprise security. Vyatta version 6.1 has received IPv6 Ready Logo Phase 2 certification, verifying the implementation of IPv6 core routing protocols. For cloud providers and enterprises moving applications or servers to the cloud, Layer 2 cloud bridging allows physically separate networks to communicate with each other over the Internet securely, as if they were on a single Ethernet network. Regarding security enhancements, Vyatta also has added stateful firewall failover and enhanced intrusion prevention services through a partnership with Sourcefire. The upshot, says Vyatta, is that the Vyatta Network OS makes it “even easier for large, distributed organizations to connect, protect and secure physical, virtual and cloud computing environments.”

www.vyatta.com

Please send information about releases of Linux-related products to newproducts@linuxjournal.com or New Products c/o *Linux Journal*, PO Box 980985, Houston, TX 77098. Submissions are edited for length and content.

Fresh from the Labs

0 A.D.—Stunning Real-Time Strategy Game

www.wildfiregames.com/0ad

When exploring these projects, sometimes I have to dig deep to find something that will arouse my audience, but other times, I'm instantly mesmerized and need discipline, so I don't run out of writing space! This is one of those times. According to the game's Web site:

> *0 A.D.* (pronounced "zero ey-dee") is a free, open-source, cross-platform real-time strategy (RTS) game of ancient warfare. In short, it is a historically based war/economy game that allows players to relive or rewrite the history of Western civilizations, focusing on the years between 500 B.C. and 500 A.D. The project is highly ambitious, involving state-of-the-art 3-D graphics, detailed artwork, sound and a flexible and powerful custom-built game engine.
>
> The game has been in development by Wildfire Games (WFG), a group of volunteer, hobbyist game developers, since 2001....In short, we consider *0 A.D.* an educational celebration of game development and ancient history.

Installation At the time of this writing, the game was in an early testing version (Build: Jul 29 2010—7732-release). Tailored packages purportedly are available in various guises for Gentoo, Arch Linux, Debian, OpenSUSE, Fedora and Mandriva. If you can, trust me, grab a binary! A package was supposed to be available for Ubuntu on the PlayDeb.net Web site, but I got an error when I tried to download it. If you're unlucky like I was, source packages are available, and as that's the usual "distro-neutral" option, that's what I'll be running with here. This project is in its infancy, and compilation will be more streamlined in the future (possibly by the time this article is printed), but for now, strap yourselves in—this will be a long one!

As far as library requirements go, the documentation states you need the following:

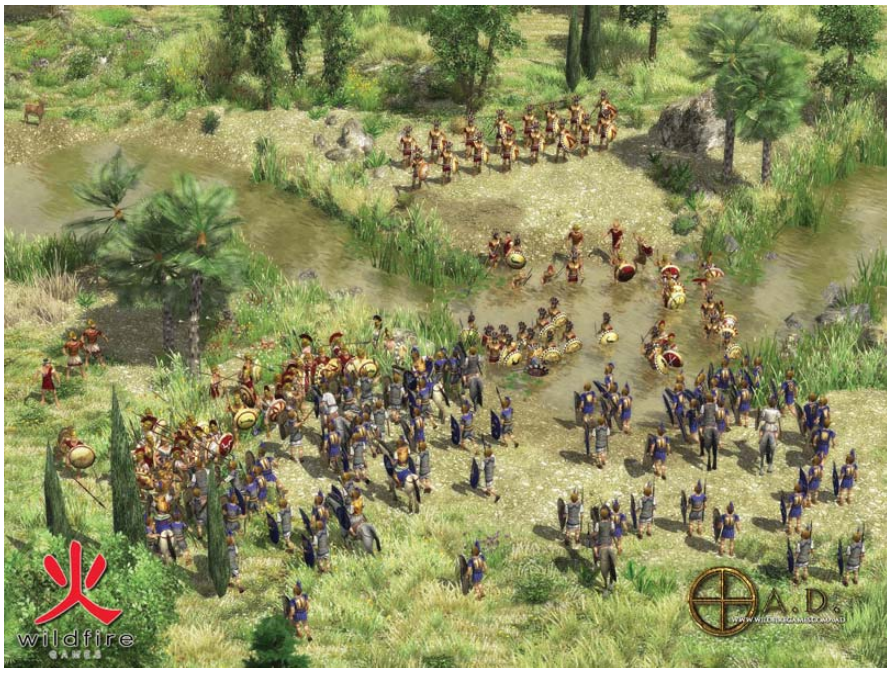
- GCC (at least 4.0, preferably 4.3).
- Subversion.
- NASM—there is a bug in NASM 2.06 on x86_64, so you might want to try a different version if you get errors.
- SDL.
- Boost.
- zlib.
- libpng.
- libxml2.
- OpenGL.
- OpenAL.
- libogg.
- libvorbis.
- wxWidgets (probably called wxgtk on Linux; optional, but required for the editor tools).
- Gamin (FAM should work too; Linux only).
- BFD (typically called something like binutils-dev; Linux only).
- ENet 1.2.x (not 1.3.x).
- DevIL.

Of course, this doesn't include the "-dev" libraries you'll need to compile it, and I needed to grab a few for my Kubuntu machine. The docs are helpful enough to recommend this command for Ubuntu or Debian (at least unstable and testing):

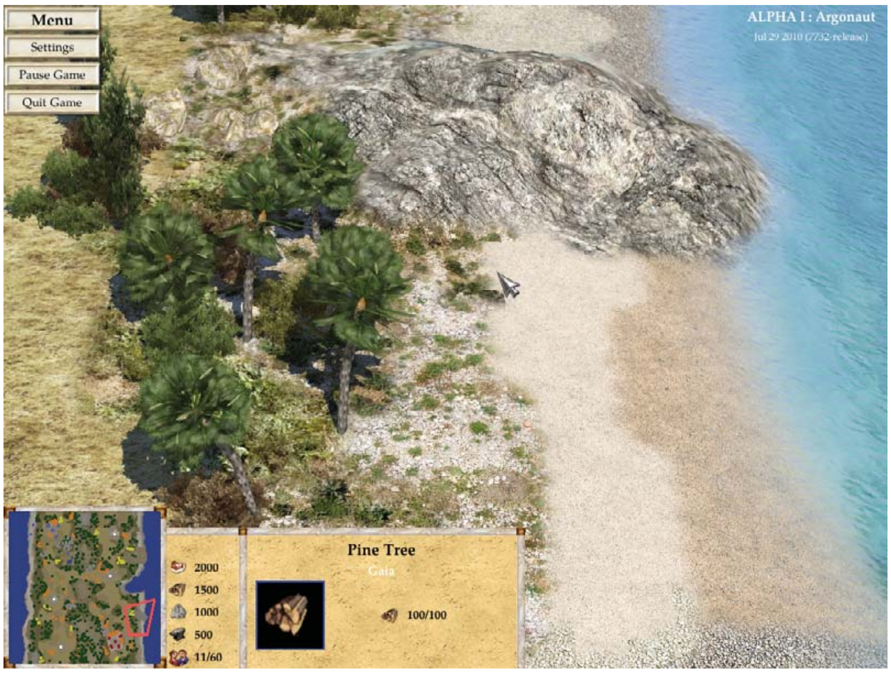
```
sudo apt-get install build-essential libsdl1.2-dev zlib1g-dev
 ➥libpng12-dev libjpeg62-dev libgamin-dev nasm
 ➥libwxgtk2.8-dev libboost-dev libboost-signals-dev
 ➥libboost-filesystem-dev libopenal-dev libalut-dev
 ➥libvorbis-dev libogg-dev libcrypto++-dev binutils-dev
 ➥libdevil-dev libenet-dev libxml2-dev
```

Sorry I don't have room for helping out with other distros, but the package names should at least give you a hint for your own system.

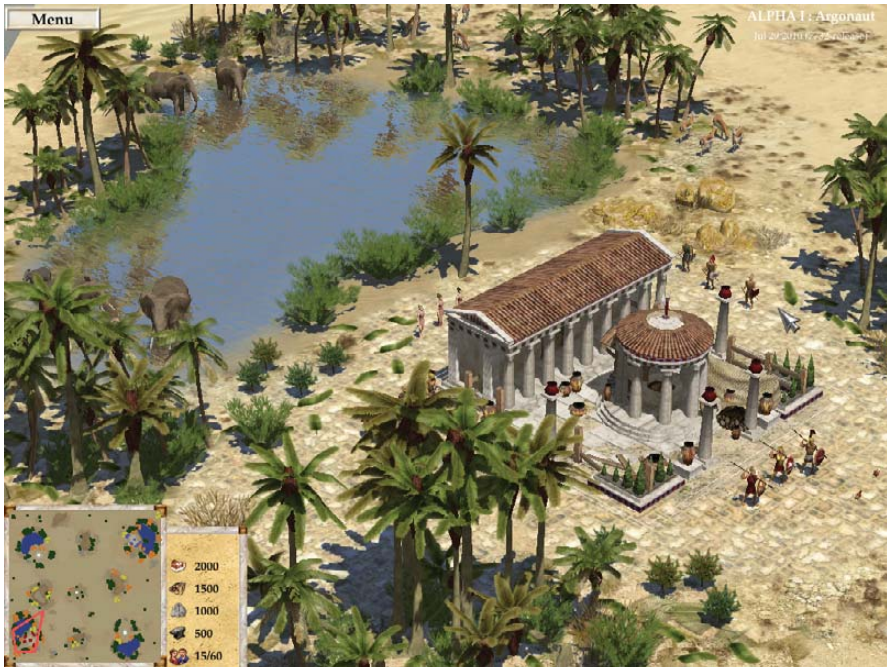
Once the packages are sorted, grab the latest tarball files from the Web site,



An Epic Battle at the *0 A.D.* Web Site



Some of This Game's Stunning Scenery



0 A.D. Showing Off Some of Its Classical Architecture Next to an Oasis

specifically the 0ad build and data files. Now, this is important: copy the two files into the *same* folder, and extract them both there.

Next, open a terminal in the (0ad main directory)/build/workspaces/ folder, and run the command:

```
$ ./update-workspaces.sh
```

A series of files will download for several minutes. Once they're done, continue with the following commands:

```
$ cd gcc
$ make
$ cd ../../../binaries/system
```

There's none of that blocky squared mapping that screams of a basic-level designer—you may as well be looking at a more convincing Google Earth!

Run the game with:

```
$ ./pyrogenesis_dbg
```

Usage This game is still in its early alpha stage, so there's no single-player AI yet, but you can at least test the game mechanics. For now, you can play multi-player over a LAN—at least I think so, I didn't get the chance. For now, fire up a single-player map and have a look around. I recommend the default choice to begin.

Once you're in the game screen, you're presented with a village that is your starting base with a number units (people). To scroll around your world, middle-click and drag your mouse. Left-click a unit or object to select it, and right-click on a map to tell it where to go or what to do.

Units seem able to go beyond simply one class, like harvesting wood or combat. Instead, a unit seems to be capable of combat, wood harvesting, hunting, stone milling and so on. Hover your mouse over a piece of rock for instance, and the mouse pointer will change to a stone-working icon. Hold your mouse pointer over an enemy unit, and an attack sign will appear. Hold it over some livestock, and an icon of meat appears.

What becomes inescapable are the gorgeous graphics, with beautiful 3-D rendering, especially when you scroll around the map. In particular, I noticed some photographically mapped rocks, and trees are lush and green, rendered in a very tasteful and intricate 3-D. There's a distinct lack of sprites around the place to keep the game from feeling tacky.

Oceans move around dynamically with rippling water, beaches graduate from pebbles to sand and move gently on to the seemingly random but convincing beach vegetation, which itself graduates into thicker inland bush. Everything's just so organic. There's none of that blocky squared mapping that screams of a basic-level designer—you may as well be looking at a more convincing Google Earth!

The soundtrack is gentle and tastefully done, with the sort of soundloops that can be played over 100 times without becoming annoying, and without turning into bland muzak. Unit animations also are wonderfully smooth, with many frames per second and shadow rendering.

A scenario editor also is included for those who want to make their own levels. Look on-line to see galleries of some absolutely stunning efforts. But, that's for another day.

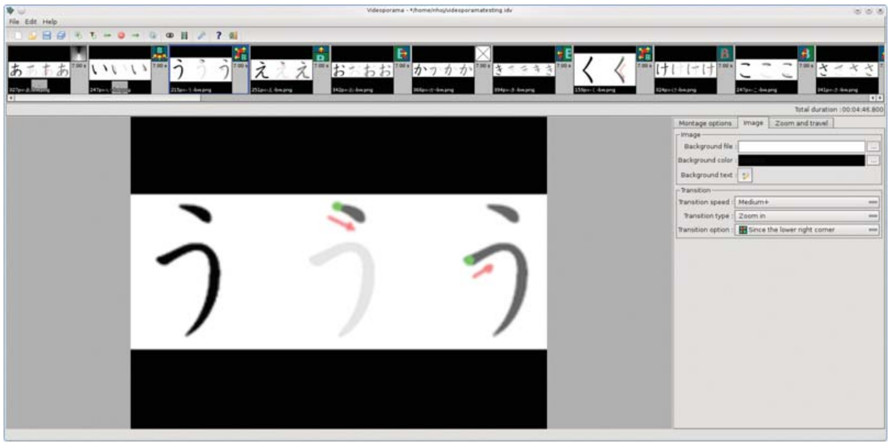
I'm not a real-time strategy fan, and I haven't been excited about an RTS game since *Command and Conquer: Red Alert!* However, this is the first RTS I'm actually enthused about, and I'll keep a close eye on it in the future. The graphics are easily on par with commercial efforts (and even better than quite a few). Hopefully, the gameplay dynamics are just as solid as development continues. My keen hope is that the *free 0 A.D.* one day dominates LAN parties and cafés, outdoing its commercial rivals and becoming a free software gaming classic.

Videoporama—Slideshow Video Sequencing

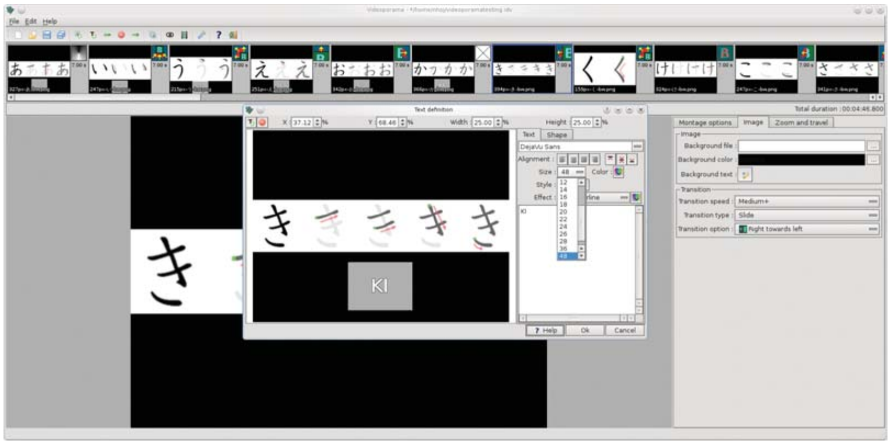
www.videoporama.tuxfamily.org

If you're chasing a picture and music slideshow maker with a decent interface, you may find yourself at home with Videoporama. According to the Web site: "Videoporama is an application for the creation of video sequences established by images or by photos, fixed or livened up. These sequences are assembled in a slideshow by means of transitions of sequence to produce complete videos."

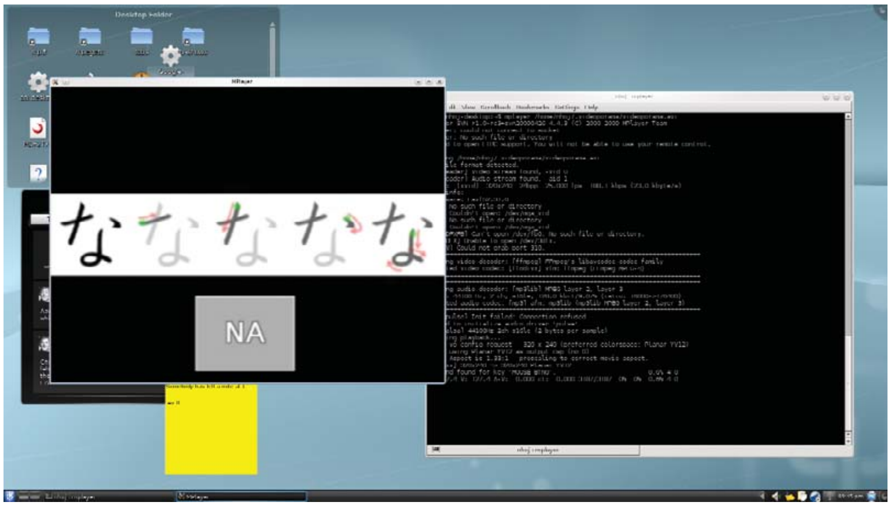
Installation A binary package is provided for those using Debian or Ubuntu, but lo and behold, the package was corrupt when I downloaded it. I'm having a bad month with packages! Still, the all-purpose source package is available, and thankfully, it isn't a pain to use. I was lucky when it came to installing Videoporama, as I had all of the needed libraries installed and didn't encounter errors. Nevertheless, it's unlikely you'll have the same luck. The documentation says you need the following libraries:



Here I am using Videoporama to brush up on my kana to the soundtrack of Alice in Chains.



You can add captions to any frame of a slideshow.



My kana video working independently of Videoporama.

- Python: version >= 2.5 but not yet 3.x.
- Qt: version >= 4.4 (best to use Qt 4.6 and above).
- pyqt: version >= 4.x.

- pil: version >= ? (Python Image Library).
- mjpegtools: version >= 1.8.

Videoporama also makes use of the following external programs: ffmpeg, sox, ppmtoy4m and mplayer.

Head to the Web site and grab the latest source tarball. Extract it, and open a terminal in the new folder. If your distro uses sudo, enter the command:

```
$ sudo python setup.py install
```

Otherwise, enter the commands:

```
$ su
# python setup.py install
```

To run the program, enter:

```
$ videoporama
```

Usage When started, a status screen appears telling you what features will and won't be available, especially in regard to the use of certain codecs. If any crucial codecs are missing, you still can add them post-compilation, and the extra sox libraries are a good place to start, especially with MP3 output and the sox ffmpeg package.

Once you're inside the main screen, if you look to your right, there is a section with three tabs. This is where most of your file tweaking will take place—you can select things such as aspect ratio, what audio track to use, overlaid text, zooming options and so on. The default ouput format is VCD/SVCD, but you'll probably want to change that. I chose AVI (XVid/MP3) myself.

Let's make a slideshow. If you look at the icons below the main menu, there's a green box with a + sign on its bottom-right corner. This will add an image, and once something's added, simply repeat the process to add more.

To add a sound file, look in the tab Montage options, where there's a sound file field. Further below is the output section where you can choose the previously mentioned codecs, as well as specify where your video file will end up.

Timing always will be an issue with a slideshow/music combo, and if you want to edit the timing of an individual frame, look in the Zoom and travel tab where there is an option called Time display fixe (presumably a typo).

I've got room for only one more feature, so I'll go with text. The Overlaid text part of the Montage Options applies text to the entire file. If you want to add text to an individual frame, the Image tab's Background text button will do exactly that, and both feature options like placement, font size and so on.

Ultimately, this is a fairly intuitive program after the first five minutes or so of usage. The project could do with some features like auto-syncing the timing of frames with the length of an audio track. And personally, I'd change some of the GUI methodology, but it's a new project, and things may change by the time you read this. At the end of the day, I still recommend Videoporama, and it's been a pretty pain-free ride so far, which is a good sign with OSS projects.■

Notes on *Paintown* from Author Jon Rafkind

Jon Rafkind is the author of *Paintown*, one of the projects covered in this column in the August 2010 issue of *LJ* (**www.linuxjournal.com/article/10765**). He wrote us the following in regard to the controversy between himself and the Beats of Rage Project mentioned in that article:

In 2004, I searched the Internet for fighting games that would run on Linux. I didn't find much, but I did find Beats of Rage (BoR). I attempted to port BoR to Linux, but it was a hassle, and so I decided to start a project from scratch. I did not start with the code from BoR and modify it; I wrote the first file starting from an empty buffer.

I did take most of the graphics from BoR, so I could get a game up and running quickly. I noted this in the LEGAL file in the project source with the following lines:

```
data/bgs/**/*.png - http://www.senileteam.com/games.php#bor.
data/chars/**/*.png - http://www.senileteam.com/games.php#bor.
data/sprites/**/*.png - http://www.senileteam.com/games.php#bor
```

I felt this was enough credit to the BoR team, seeing as how that was all I used from the project. It is not surprising that the *Paintown* and BoR engines look similar because they use the same graphics (for now), but the level of sharing ends there. I have been trying to replace the main graphics in *Paintown* for a while (I even started a river of ransom clone), but so far, I still am using the BoR graphics.

The BoR people were upset that I was "ripping them off" and made various claims about me stealing parts of their game. Here is a thread on the BoR forums in which I tried to clarify their complaints: **www.senileteam.com/forum_thread(84).html**.

I asked them in a respectful manner what elements of their game I ripped off, but unfortunately, the BoR creator, Roel, did not want to respond to my request and instead insulted me. Eventually I added "Senile" to the credits list of *Paintown*, although I don't think anyone in the BoR community knows about this. I have never been confronted by anyone in the BoR community with specific examples of things I stole from them (other than graphics, which I didn't "steal"), although a few members have made vague statements that I "stole stuff".

Paintown was rightfully pulled from PlayDeb.net because it uses copyright art. I never asked for *Paintown* to be hosted on PlayDeb.net, so I don't mind this decision.

John Knight is a 26-year-old, drumming- and climbing-obsessed maniac from the world's most isolated city—Perth, Western Australia. He can usually be found either buried in an Audacity screen or thrashing a kick-drum beyond recognition.

Brewing something fresh, innovative or mind-bending? Send e-mail to newprojects@linuxjournal.com.

HARDWARE

Chinavasion Pico Projector

A Wi-Fi-enabled pico projector that runs Linux? Find out what our Hack Editor thinks of the hackability of this device.

KYLE RANKIN

It seems like Linux runs on just about everything these days from phones to televisions, and as my August 2010 column shows, it even can power a refrigerator. In fact, I bet many Linux users have become a bit jaded about it and almost expect electronics to have Linux under the hood—I know I have. Even so, I found myself surprised and intrigued when I heard about a pico projector sold by Chinavasion that runs Linux (**www.chinavasion.com/product_info.php/pName/mini-projector-with-wifi-wireless-remote**). Once I read the specs and discovered it had built-in wireless networking, an SD card slot and supported USB host mode, I started daydreaming about all the possibilities. Sure, you could use the projector for its *intended* purpose—playing media files—but imagine if you could access the Linux user space underneath. I knew I had to get one, and I had to try to hack into it. Read on for my review of the device and a chronicle of my adventures with it so far.

In a sense, this article is two reviews. First, I review the product as it's sold, then I discuss what I found when I looked under the hood. Chinavasion was kind enough to send me a review unit and was even open to my hacking on it, as long as I made it clear that taking it apart or messing around under the hood would void the warranty. At the time of this writing, the projector costs $235 and includes the projector itself, an IR remote control, a small IR keyboard, the power cable and adapter to US outlets, a tripod and an RCA cable. The projector itself has the following specs as listed from the product page:

- Dimensions: 137mm x 77mm x 23mm
- Projector: 10 lumen LED projector at 640x480
- 802.11b/g wireless support
- SD card slot
- USB port with USB host adapter support
- Two-hour battery life
- Two-thousand-hour lamp life
- Video: VOB, MPEG1/2/4 AVI, XVID
- Audio: MP3, WMA, AAC
- Photo: JPEG, PNG, GIF, BMP

Pico Is Not Just a Text Editor

I have to say, I'm still new enough to pico projectors that I'm amazed they have been able to fit a projector into such a small space (Figure 1). I mean, I can fit this projector in my back pocket, and it's about the size of a pack of cards only longer. However, that small size comes at a price—brightness. Where your average full-size projector might sport more than 2,000 lumens, this projector sits at a fraction of that with 10 lumens. That means you will want to use this in a rather dark room, preferably at night. Also, although the Web site lists the projector as supporting up to a 70-inch projection size, you'll find better results the closer the projector is to the surface.

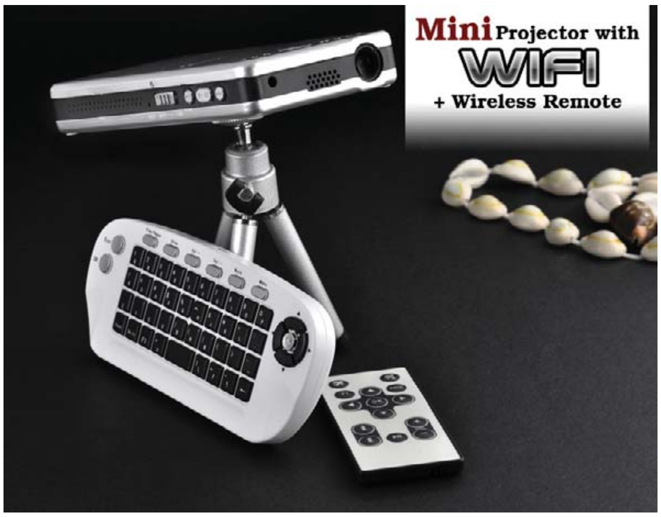


Figure 1. The Projector on Its Tripod with Both Remotes and a Decorative Shell Necklace

The Fan

The first thing you notice when you power on the projector isn't the brightness—it's the sound. No, that's not a jet engine—it's the fan. It turns out that the LED lamp gets quite hot, and as any overclocker knows, when you want to cool a computer quietly, you want a large fan so the blades can spin at a lower RPM. When you have a small space and must use a small fan, you have to resort to very high RPMs to pull enough cool air through the system. It really sounds a lot like the fans you'll find in many 1U servers. Even with the fan, you'll notice after the projector has been on for a while, the metal surface gets quite hot. Apparently this is expected, but I still wouldn't put this projector on a bed or your lap. Luckily, the speaker is loud enough to drown out most of the fan noise. I personally find the sound of fans soothing (I spent many years sleeping near a full-size desktop with fans running 24x7), but I imagine the fan noise will bother some people.

The Interface

The projector has a simple interface that you can navigate either with buttons on the device itself or through the included IR remote control or keyboard (Figure 2). Basically, you move left and right through the different operating modes and press Enter to select a particular mode. The device I received included modes to play videos and music, display photos, do limited video and music streaming, and had a calendar and a weather app. I have to say that the interface itself isn't impressive or flashy at all—it's just a few icons and simple menus to get you to the media you want to play.

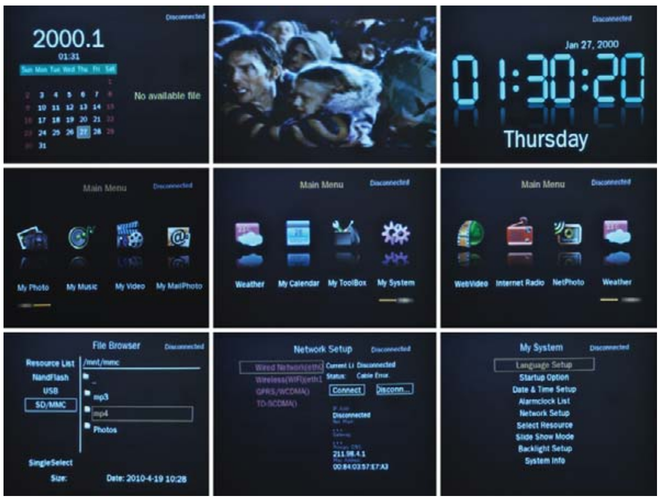


Figure 2. Sample User Interface Screens

Media Playback

The projector supports quite a number of media formats, and I was able to play the movies I had ripped to AVI files. The only real problem I encountered was that some of the videos had somewhat low volume, so the fan did drown out the sound a bit, but that will vary depending on the video. The nice thing is that you can have it play media stored on either an SD card or a regular USB hard drive or even the ~400MB of free space it has on-board. Since it has support for USB hard drives, you essentially have unlimited storage choices, but it would have been nice if they had taken advantage of the wireless card to allow for network storage. The device does have a streaming app, but your choices are limited to YouTube and a few presumably Chinese streaming sites. YouTube streaming worked okay, although the interface is clunky and somewhat difficult to use. The other apps seemed to work as advertised but offered just basic functionality.

You might have noticed I didn't mention any sort of presentation app, and that's because there isn't one! This projector is strictly for media playback, so for now, you'll have to run your presentations from something else. The device has a port to connect an RCA cable, but that's for video out, not video in.

The wireless network can be configured from a settings page, and although the interface was simple, it did work fine and detect and connect to my wireless network. When you get to that menu, you'll get excited like I did when you notice that it lists not only wireless but also wired, GPRS/WCDMA and TD-SDCMA configuration; however, none of those are actual valid options, which makes me wonder why they were added to that menu.

I'll be honest. Although it's cool that they were able to fit all these features in a small device, if it weren't for the fact that there is a computer running Linux under the hood, I don't know how excited I would be about it. It does okay with media playback, but the interface just feels incomplete. At $235, it's a neat toy, but I don't know if I'd give it a recommendation to those of you who are reading this article if we stopped the review right here. Fortunately, this is only the beginning.

Put On Your Hacker Hat

Before the projector arrived, one of the first things I asked Chinavasion was how I could access the Linux OS behind the interface. They checked with the manufacturer who said it wasn't possible to access the OS as it was "hard-wired into the unit". Well, that sounded like a challenge to me, so my follow-up question was whether I could hack on the device. As I mentioned earlier but want to reiterate, if you do any of the things I mention below, you will **void your warranty**, and Chinavasion will not accept a return if you brick your projector. Now that I've said that, let's have some fun.

My One-Minute Root Exploit

When the projector arrived, I poked at the regular interface for a second and then decided I couldn't wait—I had to get root on this device. The way the system boots, there didn't appear to be a way to change to a different virtual terminal (VT), so I figured my main route to a shell was through the network. After I connected the device to my wireless network, I ran a port scan, and wouldn't you know it, port 21 was open. "Hmmm", I thought, "they couldn't possibly have telnet running on this." Yet when I connected to the port, I got a nice ASCII art splash screen and a login prompt. I prepared myself for a day of brute-force attacks, but I was pleasantly surprised to find I could log in with root and no password. Jackpot.

Before I go into details on the user space for the projector, this a good place to list another way I discovered you could root this device, in case future versions no longer ship with telnet enabled. As I poked around on the filesystem, I noticed a script that is loaded at boot and easily could be exploited named /etc/init.d/S98z_auto_upgrade. This script is part of the init process and contains the following code:

```
#!/bin/sh

if [ -e /mnt/mmc/autorun.sh ]
then
        chmod 777 /mnt/mmc/autorun.sh
        sh /mnt/mmc/autorun.sh
        rm -rf /mnt/mmc/autorun.sh
        sync
        reboot
fi
```

It appears this script is set up so that the manufacturer easily can upgrade the software on the device. A previous init script will scan for and automatically mount any SD or USB drive under /mnt/mmc. This script automatically will execute any autorun.sh file it finds in the root of that device, so if telnet were disabled (or root were given a password), all you would have to do is create a shell script to run the commands you wanted.

So here's what I found in the user space. The system runs a

2.6.10 kernel and a unique ARM processor I hadn't seen before. Here are some selected lines from /proc/cpuinfo:

```
Processor        : ARM926EJ-Sid(wt) rev 5 (v5l)
BogoMIPS         : 124.51
Features         : swp half thumb fastmult edsp java
CPU implementer  : 0x41
CPU architecture : 5TEJ
Hardware         : VIA-VT8430
```

The device itself has 256MB of RAM and 512MB of Flash split up into two partitions: a smaller one for / and a larger, mostly empty one for /mnt/mtd. You can navigate the file structure inside a standard BusyBox shell, and you'll see the standard root-level directories you might expect, although there is a very limited number of applications installed with the custom software found under /usr/dpf. I did notice it included a version of MPlayer that outputs to the framebuffer and uses hardware acceleration for video decoding, which I thought was interesting.

My simple goal was to find some way to display a shell on the projector. I figured at the very least, with a shell, I could `ssh` in to another machine and run other applications. This turned out to be far more challenging than I thought. The first challenge was the fact that there wasn't a virtual terminal set up for a console, and when I looked at /etc/inittab, I noticed that the only getty process that was running was for ttyS0, which I assumed was for some internal serial port I couldn't see from the outside:

```
ttyS0::respawn:/sbin/getty -L ttyS0 115200 vt100
```

Although you might think you could just run another instance of getty, it turns out that the version of getty included with this device doesn't support framebuffer output nor was another getty, such as fbgetty, included. Now at this point some of you might be thinking, "What about the kernel framebuffer?" Although it's certainly possible the kernel supported a hardware framebuffer, it wasn't enabled at boot. In fact, here's the complete kernel command line:

```
mem=244M root=/dev/mtdblock1 rootfstype=yaffs2 rw
  ➥noinitrd console=ttyS0,115200n8 init=/linuxrc
  ➥lcdbpp=16 lcdpath=GOV display=TV tvsys=NTSC loadtime=0
```

Because I found no evidence of a normal distribution bootloader, I'm assuming this system uses Uboot, so without serial access to the device possibly via a JTAG header, I wasn't going to be able to change the boot arguments. This meant I had to find a frame-buffer getty compiled for ARM. At first, this seemed like an easy task. After all, Debian has been shipping ARM packages for quite some time, and all I had to do was find the right package and extract the binaries I needed. Simple, right?

Distribution Wars

As I extracted and tried to run various ARM-compiled Debian binaries on the projector, a common issue popped up—either my glibc was too new or too old. The glibc libraries are one of the core libraries for a Linux system that basically every C program needs to run, and the version this projector had seemed to fall somewhere between Debian Etch and Debian Lenny, with binaries from the former not executing and binaries from the latter complaining about a glibc that was a bit too old. It seemed like Lenny was the distribution closest to this one, but I honestly am not sure on what distribution or version this Linux install is based. The libc-2.3.3.so possibly seems to be from some ARM-compiled Fedora distribution. In any case, without glibc support, there was basically no way I could get most of my binaries to run, and I didn't want to risk bricking the device by overwriting the existing libraries, so I had to find a different approach.

Chroot and Library Hacking

My next plan was to track down a small Debian Lenny-based ARM distribution so I could copy the full root filesystem to the 400MB /mnt/mtd partition. Then, I could just chroot into that and run the commands I wanted. That way, all the commands would use that glibc, and I could add extra ARM-compiled Lenny packages. The problem I ran into fairly quickly was that, well, chroot segfaulted. Both the chroot binary that was included on the projector and the version in my Lenny install failed to work.

At this point, I started refreshing my memory on LD_LIBRARY_PATH, LD_PRELOAD and other variables I could use to tell a binary to use the version of the libraries under /mnt/mtd that I had installed. It turned out that got me a lot further. I could launch `ls` and a number of other console applications, including useful programs like strace and lsof; however, fbgetty was abandoned before Debian Lenny, so I had to try other framebuffer terminal applications in Lenny, such as jfbterm. The application would start, but it never seemed to be able to attach to the tty it wanted, and ultimately, it would error out. After trying a few things, including changing the permissions on /dev/console to be more permissive, I gave up for the moment and turned off the projector.

Bricked

The next time I turned on the projector, the splash screen started with its scrolling progress bar, but after a minute or two, the progress bar seemed to freeze, and the system was unresponsive. Great, I bricked it. But how? After all, I had just added some files to a basically empty mountpoint and changed permissions on /dev/console. At least, that's all I could remember doing. At this point, I wasn't sure what to do. I didn't want to take apart the device (at least not yet), so I rebooted it a few times and tried to see if I could get any information. One thing I thought about was that the system might have needed all that space in /mnt/mtd that I had filled up or possibly all of my changing of VTs could somehow have been remembered (even though that seemed unlikely).

At this point, I remembered the init script that automatically upgraded the device, so I decided to make a USB key with a test script that would write a file back to the USB drive. If that worked, I could just continuously boot the device with the USB key and build up a script that could repair any damage I may have done. Unfortunately, when I tried this approach, I noticed that the device didn't seem to get very far into the boot process—the script never seemed to run.

Next, I decided to connect a USB keyboard and see if the system even loaded the kernel at all. It turned out that the keyboard did work, as the Caps-Lock key lit and Ctrl-Alt-Del at the right point did reboot the device. Unfortunately, when the system froze, so did the keyboard. At this point, I tried all sorts of keyboard combinations during boot to change the VT back and forth to attempt to Ctrl-C through some script that may have stalled, and through some sort of magic voodoo, whether it was something I hit on the keyboard, or as I've come to suspect, something on the hardware that started working again, finally the system fully booted.

After I breathed a sigh of relief, I realized I needed to make an immediate backup of the filesystem, so I had some record of what was there. I also decided to take off the cover to see if I could find a serial port on the hardware and access the boot prompt and boot messages. If you look at Figure 3, you can see the bottom of the motherboard on the device, and at the top of the picture is a small five-pin white connector labeled Program that I'm assuming is some sort of serial interface. Unfortunately, this connector is incredibly tiny, so I haven't been able to track down a compatible connector yet and test this theory. Figure 4 shows the top of the motherboard—what you would see if you simply unscrewed and lifted off the top of the projector. One thing I noticed when I looked on the board was extra soldering points for an extra USB port and VGA. Moreover, seasoned hardware hackers might find even more interesting things on the board once they take a look.

Unfortunately, after I took it apart, I had the same trouble with a frozen progress bar at boot. After a number of reboots, I finally was somehow able to get it to boot completely, but since then, I've experienced the same issue a few other times. Eventually, the system will boot, but as of yet, I've been unable to track down the source of the problem. At this point, I'm leaning toward some sort of short on the device.

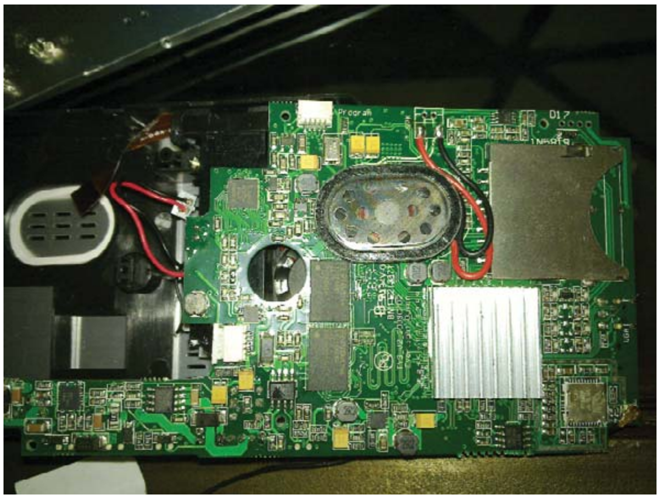


Figure 3. Bottom of the Motherboard

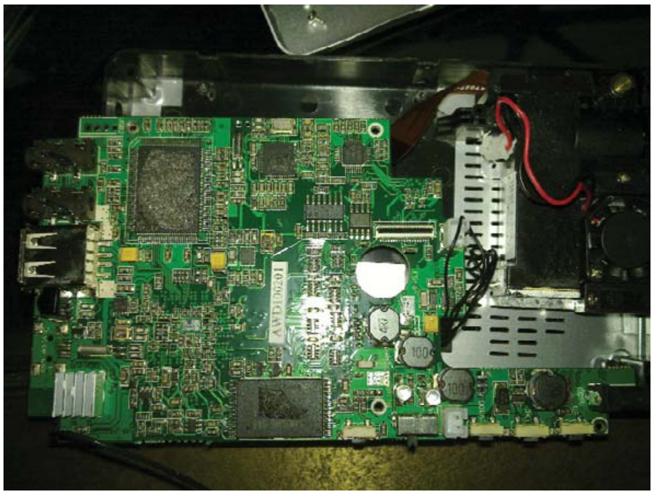


Figure 4. Top of the Motherboard

The Verdict

So what's the final verdict? Although I think that the device without Linux console access may not be too interesting to Linux geeks and that the interface needs polish, the projector has so much potential once you get access to the shell. I think this would make the start to a great pico projector project for anyone with a few hardware-hacking or framebuffer-programming skills. I just wish that the manufacturers chose a distribution that was a bit easier to work with. For $235, I think this is a pretty good buy considering how hackable it is and considering the price compared to other Linux-powered pico projector kits like the one based on the BeagleBoard.■

Kyle Rankin is a Systems Architect in the San Francisco Bay Area and the author of a number of books, including *The Official Ubuntu Server Book*, *Knoppix Hacks* and *Ubuntu Hacks*. He is currently the president of the North Bay Linux Users' Group.

Low Cost Panel PC

PDX-089T

- Vortex86DX 1 GHz Fanless CPU
- Low Power Consumption
- 1 RS232/422/485 serial port
- Mini-PCI Expansion slot
- 2 USB 2.0 Host Ports
- 10/100 BaseT Ethernet & Audio
- PS/2 mouse & keyboard
- CompactFlash & MicroSD card sockets
- Resolution/Colors: 1024 x 600 @ 256K
- Resistive Touch Screen
- Linux, Embedded XP or Windows CE OS Options

2.6 KERNEL

Microsoft Windows xp Embedded

Setting up a Panel PC can be a *Puzzling* experience. However, the PDX-089T comes ready to run with the Operating System installed on flash disk. Apply power and watch the Windows XPE user interface appear on the vivid color LCD. Interact with the PDX-089T using the responsive integrated touchscreen. Everything works out of the box, allowing you to concentrate on your application rather than building and configuring device drivers. Just Write-It and Run-It... Starting at $450 Qty 1.

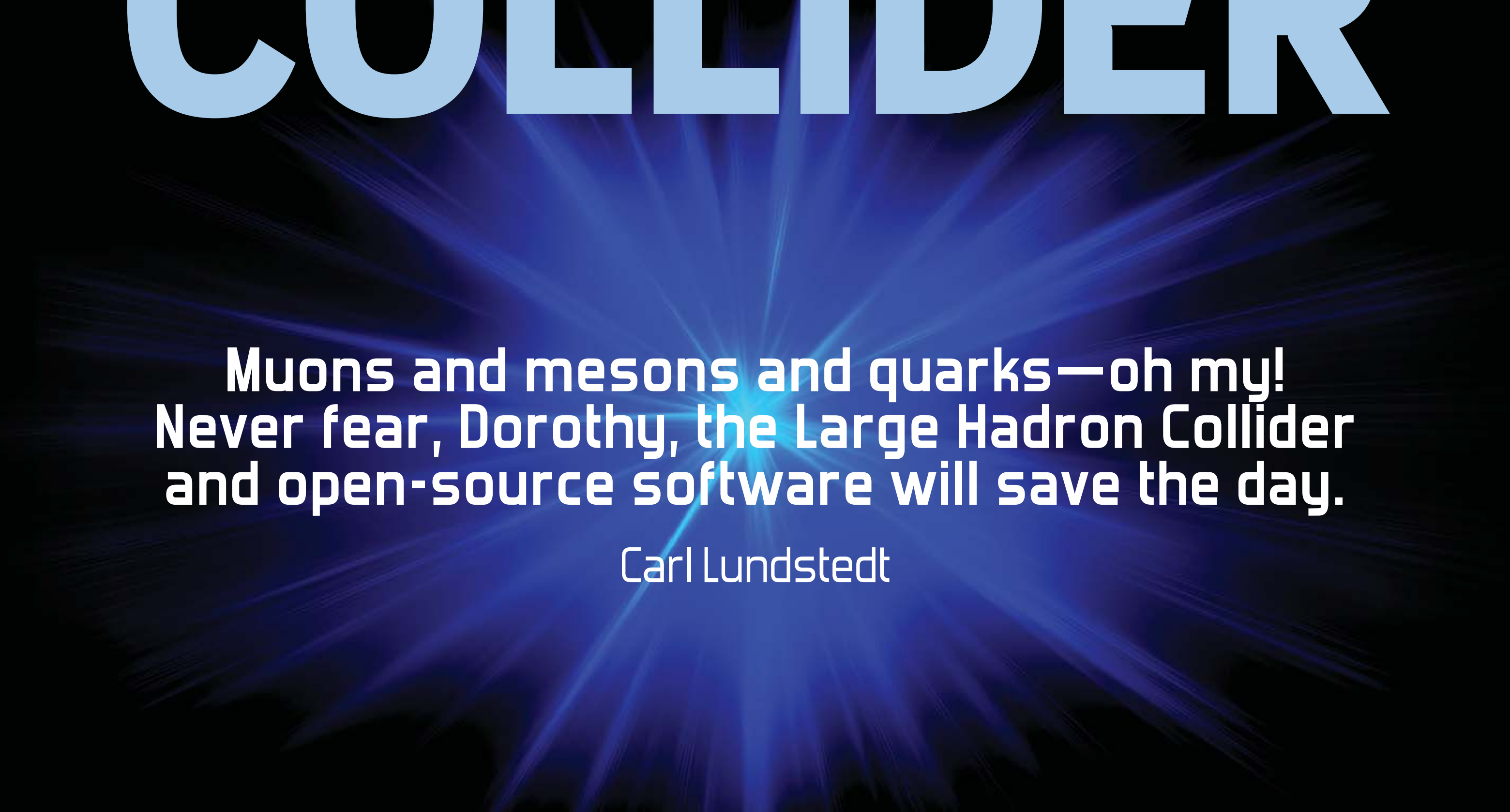
For more info visit: www.emacinc.com/panel_pc/pdx089.htm

Since 1985 OVER 24 YEARS OF SINGLE BOARD SOLUTIONS

EMAC, inc. EQUIPMENT MONITOR AND CONTROL

Phone: (618) 529-4525 • Fax: (618) 457-0110 • www.emacinc.com

THE LARGE HADRON COLLIDER



Muons and mesons and quarks—oh my! Never fear, Dorothy, the Large Hadron Collider and open-source software will save the day.

Carl Lundstedt

What is at the heart of the Large Hadron Collider (LHC) experiments? It should not surprise you that open-source software is one of the things that powers the most complex scientific human endeavor ever attempted. I hope to give you a glimpse into how scientific computing embraces open-source software and the open-source philosophy in one of the LHC experiments.

The Tiered Computing Model

The LHC near Geneva, Switzerland, is nearly 100 meters underground and provides the highest-energy subatomic particle beams ever produced. The goal of the LHC is to give physicists a window into the universe immediately after the big bang. However, when physicists calculated the level of computing power needed to peer through that window, it became clear that it would not be possible to do it with only the computers that could fit under one roof.

Even with the promise of Moore's Law, it was apparent that the experiments would have to include a grid technology and decentralize the computing. Part of the decentralization plans included adoption of a tiered model of computing that creates large data storage and analysis centers around the world.

The Compact Muon Solenoid (CMS) experiment is one of the large collider experiments located at the LHC. The primary computing resource for CMS is located at the LHC laboratory and is called the Tier-0. The function of Tier-0 is to record data as it comes off the detector, archive it and transfer it to the Tier-1 facilities around the globe. Ideally, every participating CMS nation has one Tier-1 facility. In the United States, the Tier-1 is located at Fermi National Laboratory (FNAL) in Batavia, Illinois. Each Tier-1 facility is charged with additional archival storage, as well as physics reconstruction and analysis and transferring data to the Tier-2 centers. The Tier-2 centers serve as an analysis resource funded by CMS for physicists. Individuals and universities are free to construct Tier-3 sites, which are not paid for through CMS.

Currently, there are eight CMS Tier-2 centers in the US. Their locations at universities allow CMS to utilize the computing expertise at those institutions and contribute to the educational opportunities for their students. I work as a system administrator at the CMS Tier-2 facility at the University of Nebraska-Lincoln.

By most standards, the Tier-2 centers are large computing resources. Currently, the capabilities of the Tier-2 at Nebraska include approximately 300 servers with 1,500 CPU cores dedicated to computing along with more than 800 terabytes of disk storage. We have network connectivity to the Tier-1 at FNAL of 10 gigabits per second.

LHC? CMS? ATLAS? I'm Confused.

It is easy to lose track of the entities in the high-energy physics world. The Large Hadron Collider (LHC) is the accelerator that provides the beams of high-energy particles, which are protons. It is located at CERN (Conseil Européen pour la Recherche Nucléaire). CERN is the laboratory, and the LHC is the machine. The Compact Muon Solenoid (CMS) is a large particle detector designed to record what particles are created by the collision of the beams of protons (a muon is an elementary particle similar to the electron). CMS is only one of the experiments at CERN. CMS also is used to refer to the large collaboration of scientists that analyze the data recorded from the CMS detector. Most American physicists participating in CMS are in an organization called USCMS. Other experiments at the LHC include ATLAS, ALICE, LHCb, TOTEM and LHCf. These experiments use their own analysis software but share some grid infrastructures with CMS.

Data Movement

One of the technically more difficult obstacles for CMS computing is managing the data. Data movement is managed using a custom framework called PhEDEx (Physics Experiment Data Export). PhEDEx does not actually move data but serves as a mechanism to initiate transfers between sites. PhEDEx agents running at each site interface with database servers located at CERN to determine what data are needed at that site. X509 proxy certificates are used to authenticate transfers between gridftp doors at the source and destination sites. The Tier-2 at Nebraska has 12 gridftp doors and has sustained transfer rates up to 800 megabytes per second.

It should be noted that the word data can mean a few different things to a physicist. It can refer to the digitized readouts of a detector, Monte Carlo simulation of outputs, or the bits and bytes stored on hard drives and tapes.

The network demands made by the Nebraska Tier-2 site have generated interesting research in computer network engineering. Nebraska was the first university to demonstrate large data movement over a dynamically allocated IP path. When Nebraska's Tier-2 is pulling a large amount of data from the Tier-1 at FNAL, a separate IP path automatically is constructed to prevent traffic from adversely affecting the university's general Internet usage.

Since data transfer and management is such a crucial element for the success of CMS, developing the underlying system has been ongoing for years. The transfer volume of Monte Carlo samples and real physics data already has surpassed 54 petabytes worldwide. Nebraska alone has downloaded 900 terabytes during the past calendar year. All this data movement has been done via commodity servers running open-source software.

Job Management

Once the decision was made to decentralize the analysis resources, a crucial question needed to be answered. How does a physicist in Europe run a job using data stored at Nebraska? In 2004, the computing model for CMS was finalized and embraced the emerging grid technology. At that time, the technical implementation was left flexible to allow for sites to adopt any grid middleware that might emerge. Analysis sites in Europe adopted the World LHC Computing Grid (WLCG) software stack to facilitate analysis. Sites in the US chose the Open Science Grid (OSG) to provide the software to deploy jobs remotely. The two solutions are interoperable.

The OSG's (**www.opensciencegrid.org**) mission is to help in the sharing of computing resources. Virtual Organizations (VOs) can participate in the OSG by providing computing resources and utilizing the extra computing resources provided by other VOs. During the past year, the OSG has provided 280 million hours of computing time to participating VOs. Figure 1 shows the breakdown of those hours by VO during the past year. (The Internet search for the meaning of the VO acronyms is an exercise left to the reader.) Forty million of those hours were provided to VOs not associated with particle physics. Participation in the OSG allows Nebraska to share any idle CPU cycles with other scientists. Furthermore, the CMS operational model for all US Tier-2 sites is that 20% of our average computing

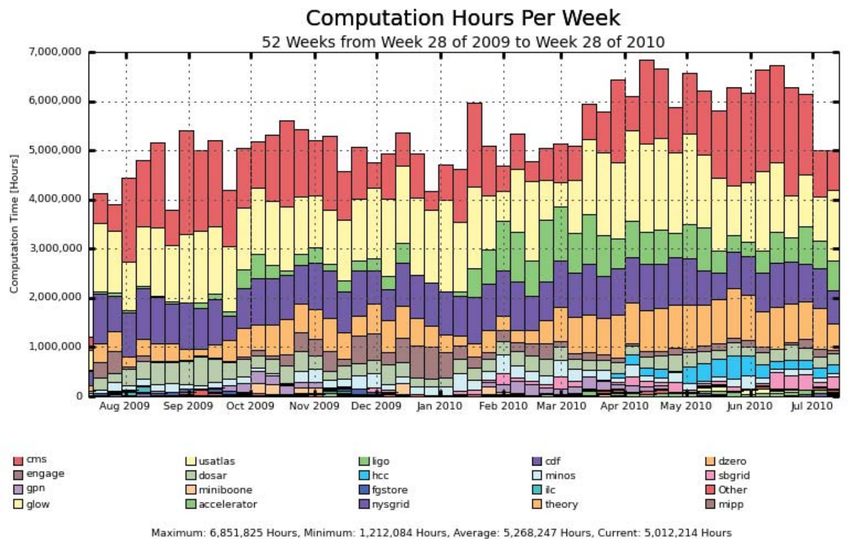


Figure 1. A week-by-week accounting of Open Science Grid usage by user VO for the past year.

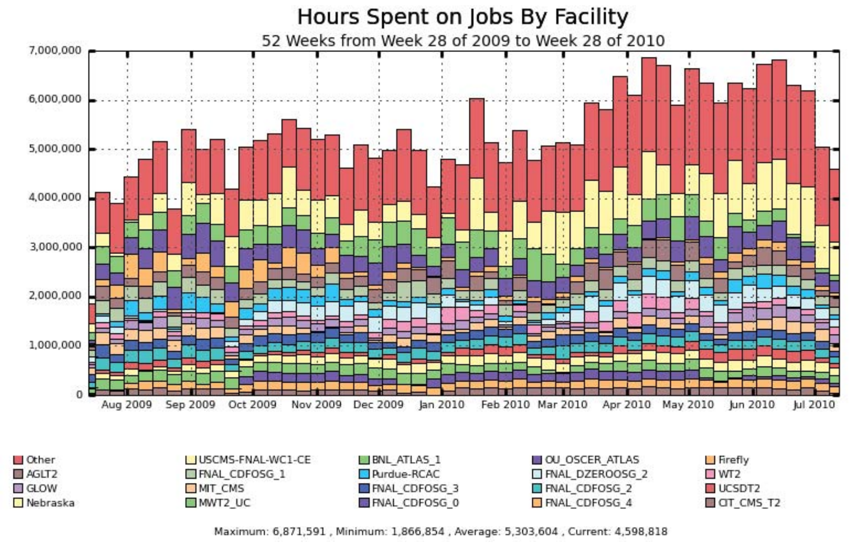


Figure 2. A week-by-week view of CPU hours provided to the Open Science Grid by computing facilities. Both "Nebraska" and "Firefly" are resources provided by the the University of Nebraska.

is set aside for use by non-CMS VOs. This gives non-CMS VOs an incentive to join the OSG. Non-CMS VO participation increases support and development of the OSG software that allows CMS to benefit from improvements made by other users. The OSG's model should serve as an example for similar collaborative efforts.

OSG provides centralized packaging and support for open-source grid middleware. The OSG also gives administrators easy installation of certificate authority credentials. Certification and authentication management is one of the OSG's most useful toolsets. Further, the OSG monitors sites and manages a ticketing system to alert administrators of problems. Full accounting of site utilization is made available by OSG so that funding agencies and top-level management have the tools they need to argue for further expenditures. See Figure 2 for CPU hours provided to the OSG by some of the major facilities.

In short, what SETI@home does with people's desktops, OSG does for research using university computing centers.

Distributed Filesystems

The CMS experiment will generate more than one terabyte of recorded data every day. Every Tier-2 site is expected to store hundreds of terabytes on-site for analysis. How do you effectively store hundreds of terabytes and allow for analysis from grid-submitted jobs?

When we started building the CMS Tier-2 at Nebraska, the answer was a software package written at the high-energy physics (HEP) experiment DESY in Germany called dCache. dCache, or Disk Cache, was a distributed filesystem created by physicists to act as a front end to a large tape storage. This model fit well with the established practices of high-energy physicists. The HEP community had been using tapes to store data for decades. We are experts at utilizing tape. dCache was designed to stage data from slow tapes to fast disks without users having to know anything about tape access. Until recently, dCache used software called PNFS (Perfectly Normal File System, not to be confused with Parallel NFS) to present the dCache filesystem in a POSIX-like way but not quite in a POSIX-compliant way. Data stored in dCache had to be accessed using dCache-specific protocols or via grid interfaces. Because file access and control was not truly POSIX-compliant, management of the system could be problematic for non-dCache experts.

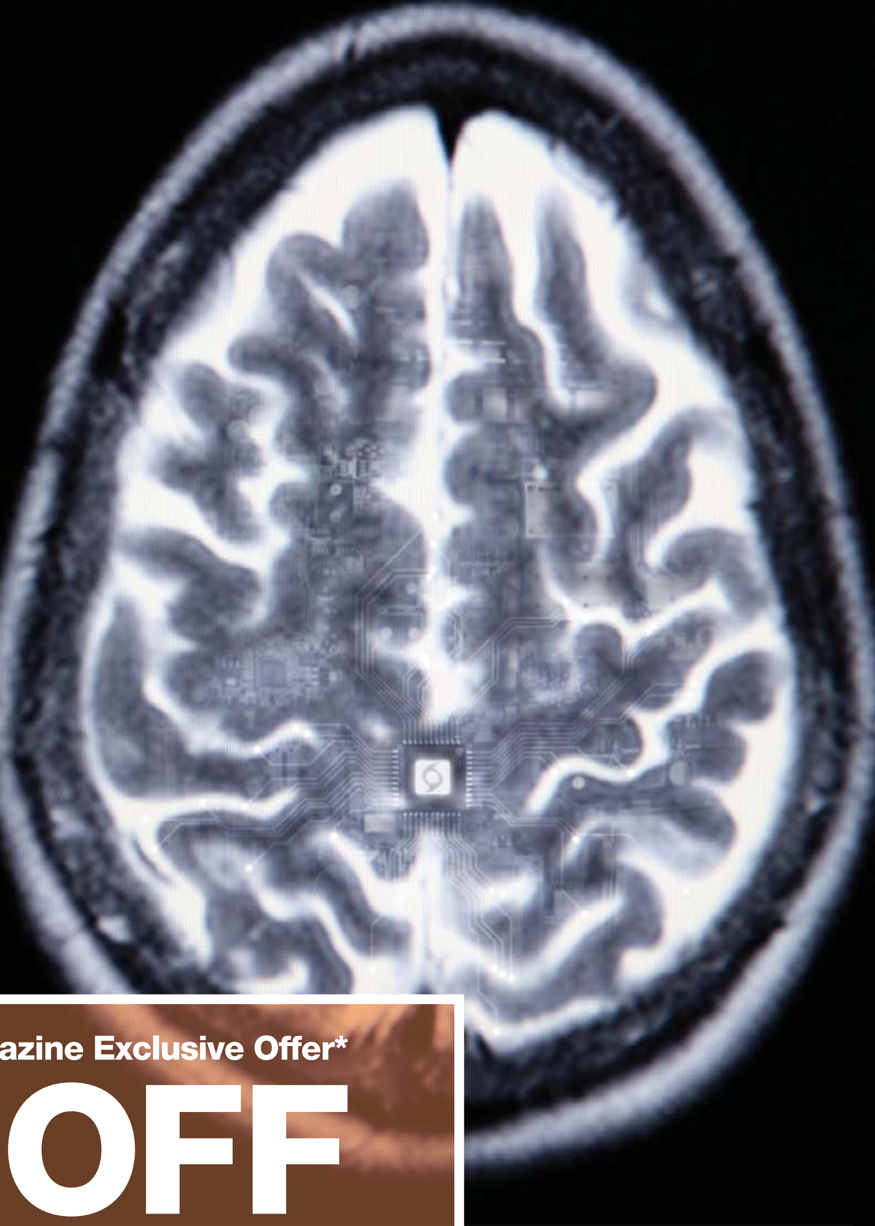
dCache storage is file-based. All files stored on disk correspond to files in the PNFS namespace. Resilience is managed via a replica manager that attempts to store a single file on multiple storage pools. Although a file-based distributed storage system was easy to manage and manually repair for non-experts using dCache, the architecture could lead to highly unbalanced loads on storage servers. If a large number of jobs were requesting the same file, a single storage server easily could become overworked while the remaining servers were relatively idle.

Our internal studies with dCache found that we were having a better overall experience when using large disk vaults rather than when using hard drives in our cluster worker nodes. This created a problem meeting our storage requirements within our budget. It is much cheaper to purchase hard drives and deploy them in worker nodes than to buy large disk vaults. The CMS computing model does not allow funding for large tape storage at the Tier-2 sites. Data archives are maintained at the Tier-0 and Tier-1 levels. This means the real strength of dCache is not being

ServerBeach

DEDICATED SERVERS. BY GEEKS FOR GEEKS.

we get how geeks think.



Linux Journal Magazine Exclusive Offer*

15%OFF

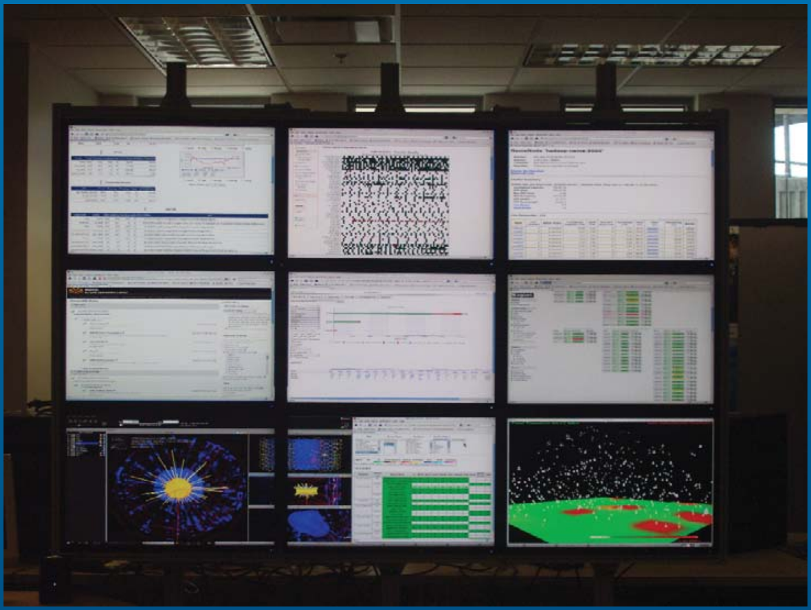
Call 1.888.840.9091 | serverbeach.com

Sign up for any dedicated server at ServerBeach and get 15% off*. Use the promo code: ***LJ15OFF*** when ordering.

* Offer expires December 31st, 2010.

HadoopViz

A student at the University of Nebraska-Lincoln, Derek Weitzel, completed a student project that shows the real-time transfers of data in our HDFS system. Called HadoopViz, this visualization shows all packet transfers in the HDFS system as raindrops arcing from one server to the other. The figure below shows a still shot.



Screens from left to right: PhEDEx Transfer Quality; Hadoop Status Page; MyOSG Site Status; CMS Dshboard Job Status; Nagios Monitoring of Nebraska Cluster; CMS Event Display of November 7 Beam Scrape Event; OSG Resource Verification Monitoring of US CMS Tier-2 Sites; HadoopViz Visualization of Packet Movement

exploited at the Tier-2 sites.

The problems of scalability and budgeting prompted Nebraska to look to the Open Source world for a different solution. We found Hadoop and HDFS.

Hadoop (**hadoop.apache.org**) is a software framework for distributed computing. It is a top-level Apache project and actively supported by many commercial interests. We were not interested in the computational packages in Hadoop, but we were very interested in HDFS, which is the distributed filesystem that Hadoop provides. HDFS allowed us to utilize the available hard drive slots in the worker nodes in our cluster easily. The initial installation of HDFS cost us nothing more than the hard drives themselves. HDFS also has proven to be easy to manage and maintain.

The only development needed on our end to make HDFS suitable for our needs was to extend the gridftp software to be HDFS-aware. Analysis jobs are able to access data in HDFS via FUSE mounts. There is continued development on the analysis software to make it HDFS-aware and further remove unnecessary overhead.

HDFS is a block-based distributed filesystem. This means any file that is stored in HDFS is broken into data blocks of a configurable size. These individual blocks then can be stored on any HDFS storage node. The probability of having a "hot" data server that is serving data to the entire cluster starts to approach zero as the files become distributed over all the worker nodes. HDFS also recognizes when data required by the current node is located on that node and it does not initiate a network transfer to itself.

The block replication mechanisms in HDFS are very mature. HDFS gives us excellent data resiliency. Block replication levels are easily configured at the filesystem level, but also can be specified at the user level. This allows us to tweak replication levels in an intelligent way to ensure simulated data that is created at Nebraska enjoys higher fault tolerance than data we can readily retransfer from other sites. This maximizes our available storage space while maintaining high availability.

HDFS was a perfect fit for our Tier-2.

Data Analysis

Once the data is stored at a Tier-2, physicists need to be able to analyze it to make their discoveries. The platform for this task is Linux. For the sake of standardization, most of the development occurs on Red Hat Enterprise-based distributions. Both CERN and FNAL have their own Linux distributions but add improvements and customizations into the Scientific Linux distribution. The Tier-2 at Nebraska runs CentOS as the primary platform at our site.

With data files constructed to be about 2GB in size and data sets currently hovering in the low terabyte range, full data set analysis on a typical desktop is problematic. A typical physics analysis will start with coding and debugging taking place on a single workstation or small Tier-3 cluster. Once the coding and debugging phase is completed, the analysis is run over the entire data set, most likely at a Tier-2 site. Submitting an analysis to a grid computing site is not easy, and the process has been automated with software developed by CMS called CRAB (CMS Remote Analysis Builder).

To create a user's jobs, CRAB queries the CMS database at CERN that contains the locations where the data is stored globally. CRAB constructs the grid submission scripts. Users then can submit the entire analysis to an appropriate grid resource. CRAB allows users to query the progress of their jobs and request the output to be downloaded to their personal workstations.

CRAB can direct output to the Tier-2 storage itself. Each CMS user is allowed 1 terabyte of space on each Tier-2 site for the non-archival storage of each user's analysis output. Policing the storage used by scientists is a task left to the Tier-2 sites. HDFS's quota functionality gives the Nebraska Tier-2 administrators an easily updated tool to limit the use of analysis space automatically.

Figure 3 shows a simulated event

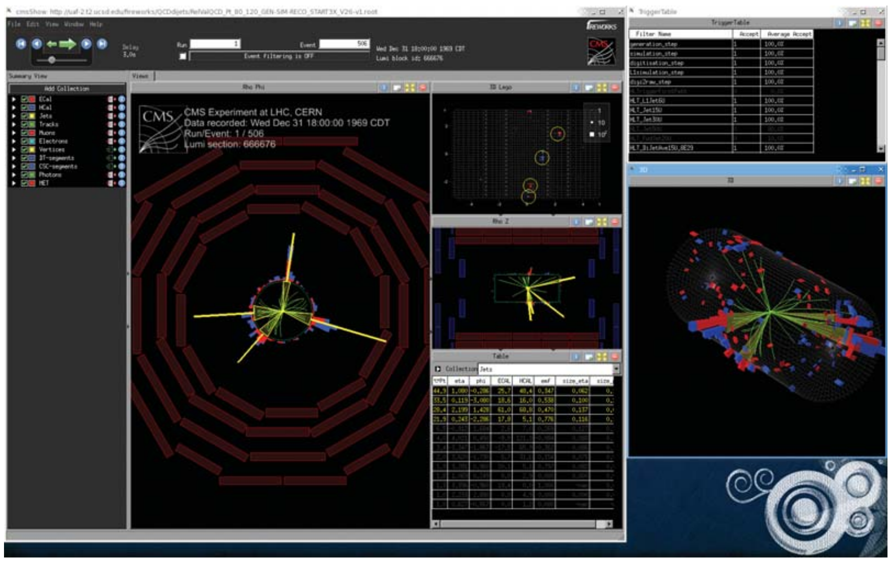

CMS Experiment at LHC, CERN
Data recorded: Wed Dec 31 18:00:00 1969 CDT
Run/Event: 1 / 506
Lumi section: 666676

Figure 3. How a physicist sees CMS—this is the event display of a single simulated event.

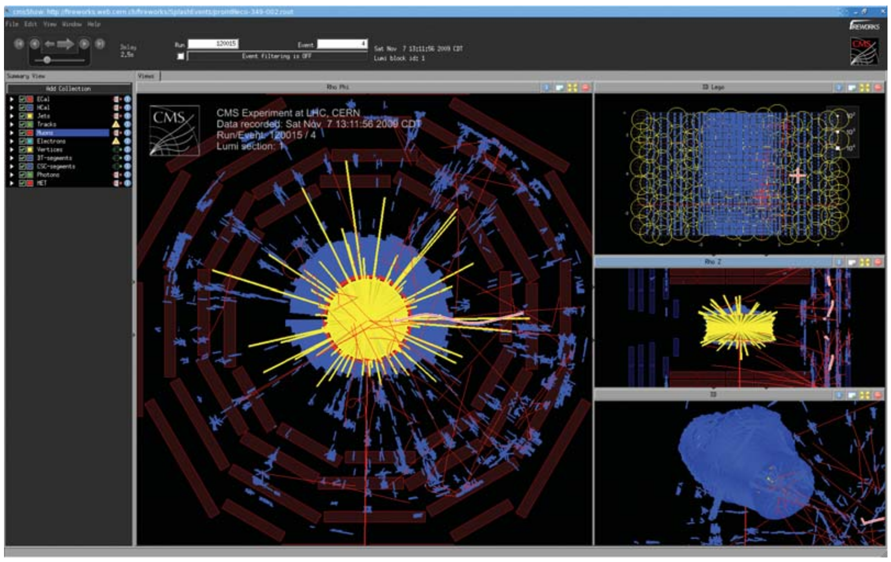

CMS Experiment at LHC, CERN
Data recorded: Sat Nov 7 13:11:56 2009 CDT
Run/Event: 120015 / 4
Lumi section: 1

Figure 4. An actual recorded event from CMS—this event shows radiation and charged particles spilling into the detector from the beam colliding with material in the beam pipe.

seen through CMS, and Figure 4 shows an actual record event.

A Grateful Conclusion

The LHC will enable physicists to investigate the inner workings of the universe. The accelerator and experiments have been decades in design and construction. The lab is setting new benchmarks for energetic particle beams. Everyone I talk to about our work seems to get glossy-eyed and complain that it is just too complex to comprehend. What I want to do with this quick overview of the computing involved in the LHC is tell the Linux community that the science being done at the LHC owes a great deal to the contributors and developers in the Open Source community. Even if you don't know your quark from your meson, your contributions to open-source software are helping physicists at the LHC and around the world.■

Carl Lundstedt received his PhD in high-energy particle physics from the University of Nebraska-Lincoln (UNL) in 2001. After teaching introductory physics for five years, he is now one of the administrators of the CMS Tier-2 computing facility located at UNL's Holland Computing Center.

Finding Your Phone, the Linux Way

If you have a Nokia N900, you can set it up so that it helps you find it the next time you lose it. The methods shown here, with modifications, can be used on other Linux phones as well.

DANIEL BARTHOLOMEW

During the years, I've misplaced my phone more times than I like to admit. Whenever it happens, I get a sinking feeling, much like when I misplace my keys or pull into the driveway and realize the groceries are still sitting in the cart...back at the grocery store.

When I notice my phone is missing, I don't automatically assume it has been stolen. Instead, I know I probably have just set it down somewhere. If I think it's likely that I left it somewhere other than home, my first order of business is trying to locate it as quickly as possible to prevent it from actually being stolen.

Other phone platforms have techniques to help you find your phone, but they are proprietary, and end users usually have no control over how they work. There also is the option of "bricking" your phone, where your service provider sends a signal to the phone that erases and disables it. I would rather try everything I can to find my phone before resorting to that option.

So, how do you go about finding your lost phone? The obvious thing to do is to try calling it. But, what if the ringer is on silent mode or the phone is out of earshot? If calling doesn't yield any results, and you're using a proprietary phone platform, you may need to jump directly to the bricking option, if that option is open to you.

My current cell phone is a Nokia N900, which was reviewed recently in *Linux Journal* (see Kyle Rankin's review in the May 2010 issue). Because the N900 is built on Linux and comes fully unlocked (with a terminal application and easy root access), you can do some cool things to find it that you can do on other phones only by voiding warranties (if you can do them at all).

First, here are a few packages that should be installed:

- The rootsh package enables root access to the N900 via the `sudo gainroot` command.

- The OpenSSH Client and OpenSSH Server packages.

These are available from the Maemo Extras package repository. To enable it, launch the App Manager, click on the title bar to bring up the menu, and select Application catalogs. Select maemo.org (it might be called Maemo Extras on some phones), uncheck the Disabled check box, and tap the Save button (Figure 1).

If the repository isn't there, you can add it by tapping on the New button on the Application catalogs screen. The details are as follows:

- Catalog Name: maemo.org.

- Web address: **repository.maemo.org/extras**.

- Distribution: fremantle.

- Components: free, nonfree.

Once you have the repository enabled, tap Download on the main screen of the Application manager and install the three packages.

With these packages installed, I launched the terminal application on the N900 and copied over my public SSH key from my desktop, like so:

```
$ mkdir .ssh
$ scp me@desktop:.ssh/id_rsa.pub .ssh/authorized_keys
```

Now, when my N900 is connected to my home network, I can `ssh` in to it from my desktop without needing a password, like so:

```
$ ssh user@n900.ip.addr.here
```

In theory, I also should be able to `ssh` to the phone when it is connected to the cell-phone network; however, my cell-phone provider does not appear to allow it—even when I configure alternate ports, all my attempts to log in fail.

Another minor issue is that the SSH dæmon binds to whatever network interface is up when it starts, and it won't respond after you switch between networks because it still is listening on the other network. To fix this, I created a tiny shell script and placed it at /etc/network/if-up.d/01_ssh. To create it, I used vi, as root:

```
$ sudo gainroot
$ vi /etc/network/if-up.d/01_ssh
```

The script itself is only two lines:

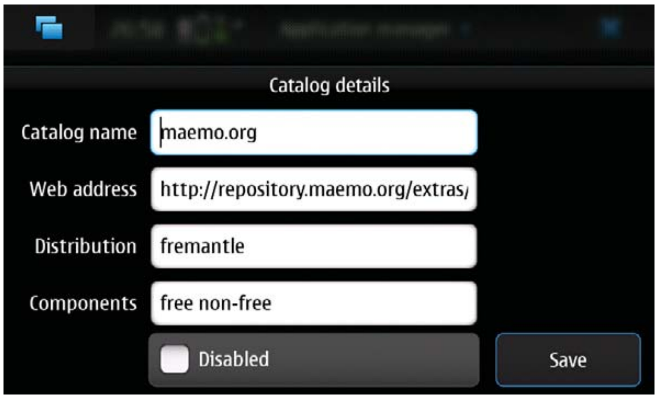
```
#!/bin/sh
/etc/init.d/ssh restart
```

After creating it, I chmodded it:

```
$ chmod 755 /etc/network/if-up.d/01_ssh
```

Any script in the /etc/network/if-up.d/ folder is run when an interface is brought up, so when that happens, the SSH dæmon will be restarted, and SSH always will be listening on the correct interface.

Now that I have reliable SSH access (when connected to a network other than the cell network), I can `ssh` in to the phone, assuming I know the IP address. At my house, knowing the IP address is easy, because my DHCP server is configured always to give the N900 the same address whenever it connects. But when I'm not at home, I might not know, so my first task is to make sure the N900 tells me what its IP address is. To do this, I simply


Catalog details

Catalog name	maemo.org
Web address	http://repository.maemo.org/extras/
Distribution	fremantle
Components	free non-free

Figure 1. Settings for the maemo.org Maemo Extras Repository

Now every time the N900 goes on-line or connects to a new network, it uploads the IP address to my home server.

add the following to the 01_ssh script I created earlier:

```
/sbin/ifconfig -a \
    | awk '/inet addr/ { print $2 }' \
    | awk -F: '{ print $2 }' \
        > /tmp/n900.ipaddr
```

This one-line script parses the output of `ifconfig -a` to extract all of the IP addresses, and it stores them in a file called N900.ipaddr in the /tmp/ folder. Then, I add the following line to upload the file to a special user I created on my home server:

```
/usr/bin/scp -i /home/user/.ssh/id_rsa \
    /tmp/n900.ipaddr n900user@home.example.net:
```

Prior to setting this up, I needed to generate an SSH key on the N900 with the `ssh-keygen` command and then copy the .ssh/id_rsa.pub file into the ~/.ssh/authorized_keys file of the unprivileged user I created to receive the file on my server.

Now every time the N900 goes on-line or connects to a new network, it uploads the IP address to my home server. Wherever I am, I simply log in to my server and look in the home folder of my N900 user, and I know what the current IP address is.

If I can `ssh` in to my phone, I can make it do several things.

For one, I can use MPlayer (easily installed from the App manager) to play a ringtone (or any other supported audio file):

```
$ mplayer /home/user/MyDocs/.sounds/Ringtones/OuterSpace.aac
```

This works even if the phone is in silent mode. Speaking of which, I can turn silent mode off with:

```
$ dbus-send --type=method_call --dest=com.nokia.profiled \
     /com/nokia/profiled \
         com.nokia.profiled.set_profile \
             string:"general"
```

With silent mode off, I can try calling the phone again.

One issue that may come up is if I have headphones plugged in. In this case, it might make more sense for me to vibrate the phone. To make the phone vibrate, I do this:

```
$ dbus-send --print-reply --system --dest=com.nokia.mce \
     /com/nokia/mce/request \
         com.nokia.mce.request.req_vibrator_pattern_activate \
             string:PatternIncomingCall
```

The phone vibrates until I shut it off with:

```
$ dbus-send --print-reply --system --dest=com.nokia.mce \
     /com/nokia/mce/request \
com.nokia.mce.request.req_vibrator_pattern_deactivate \
             string:PatternIncomingCall
```

If I still can't locate the phone at this point, it's time to get fancy.

The N900 has an accelerometer and a front-facing camera. Both of these can be triggered from the command line. The camera can be told to take a picture like this:

```
$ /usr/bin/gst-launch v4l2src \
      device=/dev/video1 \
          num-buffers=1 ! \
          ffmpegcolorspace ! \
          jpegenc ! \
          filesink location=/tmp/front-camera.jpg
```

The resulting front-camera.jpg photo then can be uploaded to my server, or some other device, and I can look at the picture and see whether I can tell from it where the phone is. Unfortunately, the front camera is a typical cell-phone camera (not very good). If the camera lens is not blocked by anything, and there is excellent lighting, you might get a decent shot, but probably not (Figure 2).

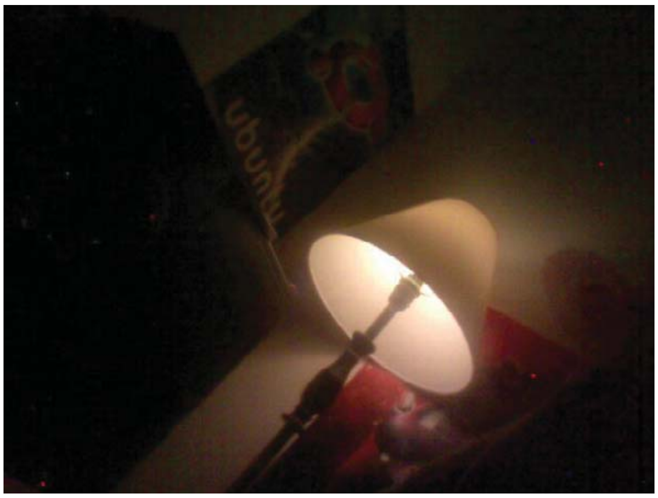


Figure 2. Pictures from the N900 front camera are not the best, but with luck, they might help you find the phone. Here I can see that the phone is sitting somewhere on my desk.

The accelerometer can tell me whether the phone is moving. I can view the output with:

```
$ watch cat /sys/class/i2c-adapter/i2c-3/3-001d/coord
```

If the numbers change by a lot every two seconds, the phone is in motion. If they aren't changing much (or not at all), the phone is just sitting somewhere.

One thing that might help at this point is to have the phone call me so I can listen in on what is going on where the phone is. To have the phone call me, I do this:

```
$ dbus-send --system --dest=com.nokia.csd.Call \
      --type=method_call \
      --print-reply /com/nokia/csd/call \
          com.nokia.csd.Call.CreateWith \
              string:"3215551212" \
              uint32:0
```

The 3215551212 should be replaced with the phone number to call. If I left the phone in a place with a distinctive sound (like a movie theater or coffee shop), I might be able to tell where it is sitting.

If I still can't figure it out, it's time to check the GPS. I saved this method until now because it requires a Python library that is available only in the maemo.org extras-devel repository. extras-devel is a repository for developers, and it contains a lot of packages that aren't quite ready for general consumption (Figure 3).

To enable extras-devel, follow the instructions at **wiki.maemo.org/Extras-devel** (see the "How to activate Extras-devel" section).

Once the repository is enabled, I install the python-location package, like so:

```
$ sudo gainroot
$ apt-get install python-location
```

After installing the package, I disabled the extras-devel repository so that I didn't accidentally upgrade any of my stable software with potentially unstable versions. I do this by opening the App manager application, clicking on the title bar and choosing Application catalogs. Then I tap on the extras-devel entry I created earlier, select the Disabled check box and click Save. I also could have just deleted it, but I decided to save it for later, just in case.

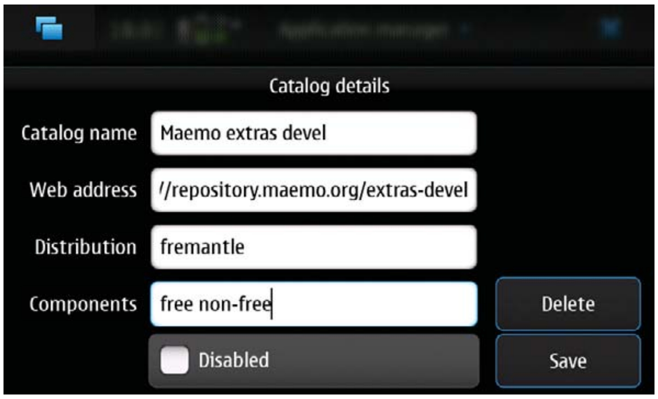
With the python-location package installed, I now could talk to the GPS using Python. There was only one problem with that plan. I don't know Python. It's been on my list of "Languages to Learn" for a long time now, and I promise I'll learn it one day, but the mythical "one day" hasn't arrived yet.

Thankfully, an N900 user who knew Python wrote a script that did exactly what I needed: **talk.maemo.org/showpost.php?p=566224&postcount=1**. The script is toward the bottom of the post. Apart from the script, the post contains many helpful ideas for recovering an N900 (many similar to what I described here), and it is well worth a read. Run the script like so:

```
$ python gps.py
```

Assuming the phone is in a place where it can get a GPS fix, it will output the phone's latitude and longitude. You can enter those coordinates into Google Maps (or any other mapping tool that lets you input latitude and longitude), and you easily can see where the phone is to within a few feet.

There is one problem with all of the above techniques. They all rely on the phone being accessible via SSH. As I stated earlier, my service provider blocks me from logging in when the phone is connected to the cell network. Additionally, any time my phone is



Catalog name: Maemo extras devel
Web address: //repository.maemo.org/extras-devel
Distribution: fremantle
Components: free non-free
Disabled

Figure 3. Settings for the extras-devel Repository

connected to a Wi-Fi network, it is behind a router, which also blocks me from logging in using SSH.

The way around this is to make the phone perform as many of the above actions as it can without me telling it to.

The first problem to solve is how to let the phone know that it is lost. I could do this by putting a specially named file on a public Web server. An example would be http://example.net/~me/my-n900-is-lost.txt. For security reasons, this file should have a name that is hard to guess (so don't use my example).

If I make the phone try to `wget` the file, and it succeeds, the phone will know it is lost and can try to tell me where it is. If the phone tries to `wget` the file and fails, it will know that it is not lost (in my case, at least not yet).

Then, the trick is how to make the N900 periodically check to see whether it is lost. On a regular Linux machine, I would do this with cron, but the N900 doesn't have cron (battery life is the reason). The N900 does have an alarm dæmon, but interfacing with it requires Python knowledge. A few projects are in the works that are trying to bring cron-like functionality to the N900, but they're not quite ready (at the time of this writing) for public consumption.

I did find one package that works well enough as a stopgap until other solutions mature. It's called dbus-scripts, and it is a dbus-monitoring dæmon that allows you to execute scripts when certain things happen on the N900. Like the python-location package, dbus-scripts is available (at the time of this writing) only in the extras-devel repository. After re-enabling the repository, I installed the package, along with a GUI configuration program, like so:

```
$ sudo gainroot
$ apt-get install dbus-scripts dbus-scripts-settings
```

After installing the packages, I disabled extras-devel again. The GUI took some of the trial and error out of configuring the triggers, so I recommend it (Figure 4). I put the commands I wanted to execute into a shell script called lost.sh and then set up the triggers in the /etc/dbus-scripts.d/dbus-scripts-settings file (Listing 1).

I decided it would make the most sense to watch for three different actions:

■ The phone changing networks.

- The keyboard being opened or closed.
- Low power.

If the network changes, that probably means the phone is on the move (maybe I left it in a car). If the keyboard is opened or closed, someone is using or looking at the phone. Finally, if the phone is just sitting there, eventually the batteries will drain away (and I'd like to find the phone before all power is gone).

The three lines in dbus-scripts-settings that watch for those actions are the following:

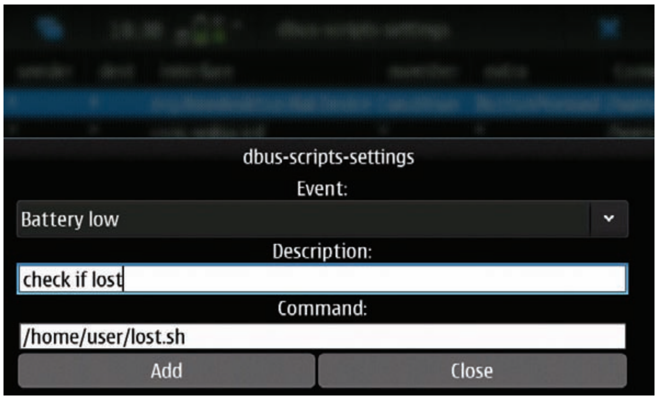
```
/home/user/lost.sh * * com.nokia.icd * * * CONNECTED
/home/user/lost.sh * * org.freedesktop.Hal.Device \
                              Condition ButtonPressed cover
/home/user/lost.sh * * com.nokia.bme.signal battery_low
```

The lost.sh script is called each time one of those events occurs. The dbus-scripts Wiki page (**wiki.maemo.org/DbusScripts**) has more information on formatting the configuration lines.

See the lost.sh script (Listing 1) for details of what it does. One thing I found in testing is that some of the commands I can run on the command line do not work at all when called from a script by the dbus-scripts dæmon. These include the mplayer command

Listing 1. lost.sh

```
#!/bin/sh
# A script to help you find your phone.
# by Daniel Bartholomew, May 2010
#
# This script is licensed under the terms of the
# GNU General Public License, version 3 or later.
# See <http://www.gnu.org/licenses/> for details.

# Some variables you need to set:
amilost="http://example.net/my-n900-is-lost.txt" # CHANGE THIS!

username="username"      # the username at your ssh host
sshhost="hostname"
backupnumber="3215551212" # backup phone that can receive SMS messages

# A timestamp variable you can alter if you want to:
timestamp=$(/bin/date +%Y-%m-%d-%H.%M.%S-%Z)

# With the vars out of the way, we can begin:

# The first step is to check for the file on the webserver.
# If it exists, we collect data and send it,
# if not, we print a message and exit.
if wget --output-document=/tmp/${timestamp}.txt \
        ${amilost} 2>/dev/null;  then
  # If we're here, the file exists and the phone is lost.
  # Get the current IP address and send it using the
  # sendsms.py script:
  /sbin/ifconfig -a \
      | awk '/inet addr/ { print $2 }' \
      | awk -F: '{ print $2 }' \
           > /tmp/${timestamp}.ip

  # sendsms.py from:
  # http://talk.maemo.org/showpost.php?p=558430&postcount=57
  /home/user/sendsms.py ${backupnumber} \
                    "$(/bin/cat /tmp/${timestamp}.ip)"

  # take a picture with the front-facing camera
  /usr/bin/gst-launch v4l2src \
      device=/dev/video1 \
      num-buffers=1 ! \
      ffmpegcolorspace ! \
      jpegenc ! \
          filesink location=/tmp/${timestamp}.jpg

  # get the location and send it
  # gps.py from: http://talk.maemo.org/showthread.php?t=47301
  /usr/bin/python2.5 /home/user/gps.py 2>&1 > /tmp/${timestamp}.gps
  /home/user/sendsms.py ${backupnumber} \
                    "$(/bin/cat /tmp/${timestamp}.gps)"

  # More things can be done.
  # See http://wiki.maemo.org/Phone_control for
  # ideas. Experiment and have fun!
  # (But don't blame me if you accidentally brick your phone.)

  # We've gathered the data we want, now copy everything to the sshhost
  # server
  # (need to have ssh-keys set up for this to work).
  /usr/bin/scp -i /home/user/.ssh/id_rsa /tmp/${timestamp}.* \
      ${username}@${sshhost}:

  # What if I want to run more commands than are listed above,
  # such as making the phone vibrate or having it call me?
  # The file I downloaded from the server can contain special
  # commands. If the file is empty, nothing will happen.
  # This probably is not the most secure thing to do, but if the
  # phone really has been stolen, this will allow you to pull
  # data off of it and/or brick the phone on purpose. Be careful!
  # Only uncomment the following line if you know what you are doing
  # and are comfortable with the consequences. I am not responsible
  # if you break your phone.

  #/bin/sh /tmp/${timestamp}.txt

else
  # if the file on the server doesn't exist the phone is not lost:
  echo "I am not lost."
fi
# That's it! We're done.
exit 0
```


dbus-scripts-settings

Event:

Battery low

Description:

check if lost

Command:

/home/user/lost.sh

Figure 4. Adding an Item to /etc/dbus-scripts.d/dbus-scripts-settings Using the GUI Configuration Program

and setting the phone to nonsilent mode. Thankfully, other commands work fine when called from the script.

The final piece of the puzzle was getting the N900 to send at least some of the data it gathered to another phone via SMS. I found a good Python script for sending SMS messages at **talk.maemo.org/showpost.php?p=558430&postcount=57**.

With the SMS script in place and called by the lost.sh script, all I have to do is create the special file on my Web server, and suddenly my N900 starts sending my designated backup phone (my wife's cell phone) SMS messages whenever one of the events I configured happens.

Conclusion

My current setup is not perfect. The main problem is that if the phone is sitting somewhere not changing networks and not being used, it might be several hours before the low-battery warning event happens and the phone discovers it is lost. As soon as one of the cron alternatives is ready, I probably will switch to using that and have the phone check every half hour to see if it is lost.

Also, the scripts I've created so far are pretty rough. With some more hacking, this will improve. But even in their current state, although they may not guarantee I will recover my lost or stolen phone, they will give me a better chance.

Finally, I don't know if there's any way for me to prevent myself from setting my phone down and forgetting about it. As much as I tell myself it will not happen, I know that someday it will. Thanks to the openness of the N900, I can do things other phones can only dream about.■

Daniel Bartholomew works for Monty Program (montyprogram.com) as a technical writer and system administrator. He lives with his wife and children in North Carolina and often can be found hanging out on both #linuxjournal and #maria on Freenode IRC. He also would like to apologize to his wife for all of the SMS messages he sent her while testing the scripts.

Small, Portable Devices with Ubuntu Linux

ubuntu

Small Form Factor Intel® Atom™ Platform

No fans, no moving parts. Just quiet, reliable operation.
Incredibly compact and full featured; no compromises.

Low-Profile Intel® Atom™ Industrial System

Small footprint platform featuring solid state storage.
System is less than 1.5" thick, yet rugged and sturdy.

Value only an **Industry Leader** can provide.

Selecting a complete, dedicated platform from Logic Supply is simple: Pre-configured systems perfect for both business & desktop use, Linux development services for greater system customization, and a wealth of online resources all within a few clicks.

LOGIC SUPPLY

Learn More > www.logicsupply.com/linux

Control Your Linux Desktop with D-Bus

Using D-Bus, you can personalize and automate your desktop.

Koen Vervloesem

Every modern Linux desktop environment uses D-Bus, a system for allowing software applications to communicate with each other. Thanks to D-Bus, you can make your desktop work the way you want. In this article, I illustrate some of the things that are possible with D-Bus. Get ready for some desktop tweaking.

D-Bus (Desktop Bus) is an interprocess communication (IPC) system, which provides a mechanism for applications to talk to each other. The D-Bus designers built it from scratch, but were heavily influenced by KDE's DCOP (Desktop COmmunication Protocol) system. Currently, D-Bus is everywhere—KDE 4 has ditched DCOP for D-Bus, and GNOME is moving toward D-Bus instead of its own Bonobo system. So, D-Bus has become a desktop-agnostic IPC mechanism. Software that uses D-Bus seamlessly integrates with your desktop, regardless which desktop environment you use. D-Bus is part of the cross-desktop project freedesktop.org, and Red Hat is the primary contributor.

With D-Bus, every program that offers services to other programs registers itself. Other programs then can look up which services are available. A program also is able to register itself for events, which some system services do, for example, to detect hot-swapping hardware.

D-Bus does not allow direct process-to-process communication, but works by providing a "bus". The bus dæmon routes messages between processes on a bus, and in this way, processes can speak to one or more applications at the same time. Each application can send messages to the bus or listen for events on the bus.

Usually, D-Bus creates two buses: a privileged system bus and a session bus. The system bus allows system-wide communication between processes with the right access permissions, and its main use is to deliver events from HAL (Hardware Abstraction Layer) to processes that are interested in hardware events. Some possible hardware events might be notification that a new hardware device has been added or that a printer queue has changed. The second bus, the session bus, is created when you log in, and it lets your applications communicate.

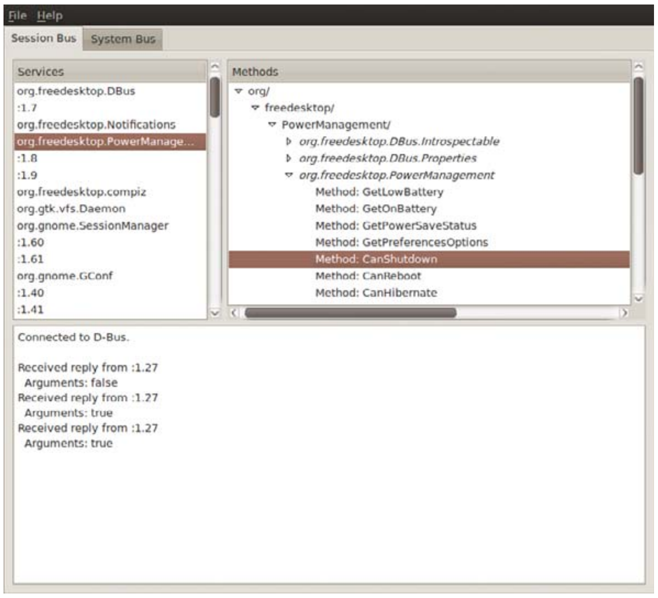
Talking D-Bus

Each application using D-Bus exposes some objects, which generally map to internal GObject, C++ or Python objects. One application can send a message to a specific object in another application. You address each D-Bus object with a unique pathname, which looks like a filesystem pathname. To ensure that each application uses unique pathnames, a D-Bus object pathname generally is prefixed with the developer's domain name, such as /org/kde or /com/redhat. The Java programming language uses the same system with package names (for example, org.sun). A D-Bus path is made up of three parts: the service name, the object path and the interface (I give examples of them a bit later in this article).

So, how do you support or use D-Bus in your own application? The core API is written in C and is rather low level. It is not really designed for application programmers to use. Different programming languages and environments have binding layers built on top of this API, such as GLib, Qt, Python, Ruby, Perl and Mono. I don't go into C or GLib (GNOME's base library) programming here, but I give some examples written in script languages like Python and Ruby, as well as shell scripts.

Which Applications Are Using D-Bus?

The freedesktop.org project has an incomplete list of applications using D-Bus on its Web site, and the bus name for each application is listed there also. However, you can find the bus names yourself by using some interesting tools to help you explore D-Bus on your own system. For example, Qt has a graphical D-Bus browser called qdbusviewer (Figure 1). In Ubuntu, you can find the application in the qt4-dev-tools package. Although it's part of KDE, the application works perfectly on other desktop environments, including GNOME.



```
org.freedesktop.DBus
:1.7
org.freedesktop.Notifications
org.freedesktop.PowerManage...
:1.8
:1.9
org.freedesktop.compiz
org.gtk.vfs.Daemon
org.gnome.SessionManager
:1.60
:1.61
org.gnome.GConf
:1.40
:1.41
```

```
org/
  freedesktop/
    PowerManagement/
      org.freedesktop.DBus.Introspectable
      org.freedesktop.DBus.Properties
      org.freedesktop.PowerManagement
        Method: GetLowBattery
        Method: GetOnBattery
        Method: GetPowerSaveStatus
        Method: GetPreferencesOptions
        Method: CanShutdown
        Method: CanReboot
        Method: CanHibernate
```

```
Connected to D-Bus.

Received reply from :1.27
  Arguments: false
Received reply from :1.27
  Arguments: true
Received reply from :1.27
  Arguments: true
```

Figure 1. QDBusViewer Running on GNOME

When you run qdbusviewer, it shows two tabs: Session Bus and System Bus. Within each tab, the left pane shows a list of service names. If you click on a service name, the right pane shows information about that service, such as the available methods and signals. For example, if you click on the service org.freedesktop.PowerManagement, and then click in the right pane through the hierarchy org/, freedesktop/ and PowerManagement/, you will have navigated two parts of the D-Bus path: org.freedesktop.PowerManagement in the left pane is the service name, and the org/freedesktop/PowerManagement in the right pane is the object path.

The object path has one final part in the right pane, three so-called interfaces, with a dot-separated name: org.freedesktop.DBus.Introspectable, org.freedesktop.DBus.Properties and org.freedesktop.PowerManagement. Each interface implements some methods and signals. These are the things you can interact with. Here, we're interested in the interface org.freedesktop.PowerManagement, as this one implements the concrete power management actions. When you click on it, you can see all implemented methods and signals in a list. If you click on the method Suspend, your computer suspends, and wakes up only when you press the power button.

Some methods, such as Shutdown, Reboot, Hibernate and Suspend implement actions, and other methods give you some information about the system. For example, the org.freedesktop.PowerManagement object implements some methods like GetLowBattery, GetOnBattery, CanShutdown and so on. If your system (laptop) is running on battery power but with enough battery time remaining, clicking on GetOnBattery gives a reply "Arguments: true" in the pane below, but if you click on GetLowBattery, it gives the reply "Arguments: false".

It's worth pointing out that qdbusviewer can show only the service names that are registered at the moment. For example, if you haven't started Pidgin, the viewer won't list the Pidgin service. Take this into account when you are exploring the D-Bus services you can use on your system.

If you are more of a command-line person, you don't have to fire up qdbusviewer. The command-line application qdbus exposes the same information. If you run `qdbus` in a terminal, you get the list of service names available on the session bus. If you run it with the `--system` flag, the services known to the system bus are shown. If you want to know the different objects a service exposes, run, for example:

```
$ qdbus org.freedesktop.PowerManagement
/
/org
/org/freedesktop
/org/freedesktop/PowerManagement
/org/freedesktop/PowerManagement/Backlight
/org/freedesktop/PowerManagement/Inhibit
/org/freedesktop/PowerManagement/Statistics
```

Now, if you want to know which interfaces the /org/freedesktop/PowerManagement object implements, use:

```
$ qdbus org.freedesktop.PowerManagement \
          /org/freedesktop/PowerManagement
```

This will give the same list of methods and interfaces that you saw in qdbusviewer. For example, the line:

```
method bool org.freedesktop.PowerManagement.GetOnBattery()
```

Software that uses D-Bus seamlessly integrates with your desktop, **regardless which desktop environment you use.**

The `bool` means this method returns a boolean value, which can be true or false. If the method doesn't return a value, for example, org.freedesktop.PowerManagement.Suspend(), the line lists `void` instead of `bool`.

qdbus also allows you to call these methods directly. For example, if you want to call the Suspend method, execute:

```
$ qdbus org.freedesktop.PowerManagement \
            /org/freedesktop/PowerManagement \
            org.freedesktop.PowerManagement.Suspend
```

Playing with D-Bus on the Command Line

In the rest of this article, I show some of the D-Bus functionality exposed by some popular applications and write some scripts to communicate with those applications and automate some tasks. Hopefully, this will give you some inspiration to communicate with your own favorite applications. I use different D-Bus tools and script languages to show the different ways you can use D-Bus.

I've already mentioned the first way to make use of D-Bus: by using the KDE programs qdbusviewer and qdbus. However, if you're not into KDE, you can use the command-line programs dbus-send and dbus-monitor to send and monitor D-Bus messages, respectively. For example, you can put the system into suspend mode with the following command:

```
$ dbus-send --dest=org.freedesktop.PowerManagement \
                /org/freedesktop/PowerManagement \
                org.freedesktop.PowerManagement.Suspend
```

As you can see, the dbus-send calls are almost identical to the ones with qdbus. The only difference is that you have to use the --dest parameter for the service name. But, let's look at something new. If you are watching a long YouTube video in your browser, the screensaver can kick in, because the Flash plugin doesn't communicate with the rest of your system. With D-Bus, you can stop this annoying behavior. The magic command is this:

```
$ dbus-send --print-reply \
            --dest=org.gnome.ScreenSaver / \
            org.gnome.ScreenSaver.Inhibit \
                string:"YouTube" \
                string:"Inhibit Screensaver"
```

With this command, you call the Inhibit method of the org.gnome.ScreenSaver interface with two arguments. The first one is the application's name. I use "YouTube" here, but it can be an arbitrary name. The second argument is the reason to inhibit the screensaver. dbus-send expects each argument to be preceded by its type, such as string, boolean, int16 and so on. The two arguments here are strings. I also use the argument --print-reply, because I need the reply of the command: the Inhibit method returns a uint32 number, which is a "cookie" identifying the inhibit request. If you want to uninhibit the screensaver, you have to send this cookie as the argument:

```
$ dbus-send --dest=org.gnome.ScreenSaver / \
            org.gnome.ScreenSaver.UnInhibit \
                uint32:1234567890
```

With these two commands, you can hack your own personal screensaver-inhibition shell script. Note: you need to save the cookie to a variable or a file when the first command runs and then substitute the actual cookie value in the command above.

If you are debugging D-Bus scripts or observing the methods and signals of other D-Bus applications, the command-line program dbus-monitor comes in very handy. Just fire it up in a terminal, and you will see all D-Bus activity scrolling by. dbus-monitor is useful for seeing all D-Bus activity in real time. So if something is happening on your system, for example, your network goes down, you can see in the output of dbus-monitor how this message is sent to the D-Bus bus. This way, you know which signals to listen for or which methods to call to tap in to the same events.

dbus-monitor also allows you to specify a set of expressions you want to watch—for example:

```
$ dbus-monitor --system "interface='org.freedesktop.NetworkManager'"
```

This will monitor all NetworkManager events. I use the --system argument because NetworkManager uses the system bus.

Scripting the Liferea Feed Reader

The Liferea feed reader has a small but interesting set of D-Bus methods. The most interesting method is Subscribe, which allows you to add a feed to Liferea from another application. One application that uses this is FeedBag, a Firefox extension that modifies the feed button in the browser's address bar: if you click on the button, it will add a subscription to Liferea instead of to the Live Bookmarks. Under the hood, this works because FeedBag calls the org.gnome.feed.Reader.Subscribe method. You can do the same from a terminal:

```
$ feed="http://feeds2.feedburner.com/linuxjournalcom?format=xml"
$ dbus-send --dest=org.gnome.feed.Reader \
            /org/gnome/feed/Reader \
            org.gnome.feed.Reader.Subscribe \
                string:"$feed"
```

Liferea provides a script, liferea-add-feed, which does exactly this, but with some added error handling.

Liferea also exposes some information via D-Bus, which is interesting if you have an alternative window manager that is not using Liferea's notification area. Then, you can brew your own notification system—just ask for the number of new and unread items in Liferea and show the output:

```
$ dbus-send --print-reply \
            --dest=org.gnome.feed.Reader \
            /org/gnome/feed/Reader \
            org.gnome.feed.Reader.GetNewItems
```

```
$ dbus-send --print-reply \
            --dest=org.gnome.feed.Reader \
            /org/gnome/feed/Reader \
            org.gnome.feed.Reader.GetUnreadItems
```

Away from the Keyboard

If you want to do more complex tasks than calling a single method, you can write a shell script with dbus-send commands or use a higher-level language to simplify the task. There are D-Bus bindings for languages such as Python, Ruby and Java.

In this next example, I implement a Python script that sets your status on Pidgin to "Away from keyboard" if your screensaver activates. This shows two aspects of D-Bus: the script waits for a signal from the screensaver, and then it calls a method in Pidgin. The script is shown in Listing 1.

Let's dissect this script. The function pidgin_status_func sets your status in Pidgin. It gets the im/pidgin/purple/PurpleObject object and then the im.pidgin.purple.PurpleInterface interface from the session bus. Then, it calls this interface's methods. It creates a new "saved status" type by first checking if the status type with name "afk" exists, and if not, it creates it ("afk" stands for "Away From Keyboard", and 5 is the "away" status type).

Then, the function checks the state variable that is an argument to the pidgin_status_func function call (I explain what this argument means later). If the argument is true, it sets the status message of the new "afk" status to "Away from keyboard" and activates the new status. The effect is that Pidgin shows your status as "afk" with the status message "Away from keyboard".

Now you need to call this function when the screensaver

Listing 1. pidgin_screensaver.py

```
#!/usr/bin/env python

def pidgin_status_func(state):
    obj = bus.get_object("im.pidgin.purple.PurpleService",
                         "/im/pidgin/purple/PurpleObject")
    pidgin = dbus.Interface(obj, "im.pidgin.purple.PurpleInterface")
    status = pidgin.PurpleSavedstatusFind("afk")
    if status == 0:
        status = pidgin.PurpleSavedstatusNew("afk", 5)
    if state:
        pidgin.PurpleSavedstatusSetMessage(status,
                                           "Away from keyboard")
        pidgin.PurpleSavedstatusActivate(status)

import dbus, gobject
from dbus.mainloop.glib import DBusGMainLoop

dbus.mainloop.glib.DBusGMainLoop(set_as_default=True)
bus = dbus.SessionBus()

bus.add_signal_receiver(pidgin_status_func,
                        dbus_interface="org.gnome.ScreenSaver",
                        signal_name="ActiveChanged")

loop = gobject.MainLoop()
loop.run()
```

GENSTOR

Linux - FreeBSD -OpenSolaris - etc.

Proven Technology.

Proven Reliability.

When you can't afford to take chances with your business data or productivity, rely on a Genstor server customized to your specifications.

Fly into the Cloud with Genstor Systems

- Up to 48 cores in a 1U.
- AMD Opteron 6100 series.
- Single high-efficiency power supply.
- Up to 512GB DDR3 memory.
- Ideal as front end processing servers.

- Up to 12 cores in a 2U
- Dual redundant high efficiency power.
- Up to 96GB DDR3 memory.
- Server Power Capping via Intel Intelligent Power Node Manager.
- Ideal as front end processing and/or storage.

- Up to 4 GPU cards.
- Dual redundant high efficiency power.
- Up to 192GB DDR3 memory
- Up 24 cores using 2 CPUs.
- Up to 8 3.5" disks.

Genstor Systems, Inc.

intel Xeon inside™

Powerful. Intelligent.

780 Montague Express. # 604
San Jose, CA 95131
www.genstor.com
E-mail: sales@genstor.com
Phone: 877-25 SERVER
408-383-0120

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

activates. Therefore, start the dbus main loop and connect to the session bus. Then, add a signal receiver that listens to the signal ActiveChanged from the org.gnome.ScreenSaver interface. If/when the signal fires, it calls out pidgin_status_func function. As the ActiveChanged signal has a boolean argument that signifies the current state of the screensaver (1 for active and 0 for non-active), you have defined one argument called state in the pidgin_status_func function. To keep listening, let the loop run indefinitely, as long as the script is running.

Pidgin has an extremely rich D-Bus interface; you can do almost anything with it. So let this example give you some inspiration to do some creative tasks in Pidgin!

Playing D-Bus

Let's look at another example, this time in Ruby. We're going to create a script that shows the currently playing song in Rhythmbox as your status in Pidgin (Listing 2).

Here you see the same type of commands as I used in the Python script: open a D-Bus session, define D-Bus services, objects and interfaces, and I define a signal receiver. And, a loop runs indefinitely to keep listening to the D-Bus signals.

Of course, this could be tidied up a bit. For example, you now are showing only the file path of the song as the status message. I'll leave it to the reader to extract the relevant ID3 tags out of the file and show them instead of the file path.

Listing 2. pidgin_rhythmbox.rb

```
#!/usr/bin/env ruby

require 'dbus'

bus = DBus::SessionBus.instance
rhythmbox = bus.service("org.gnome.Rhythmbox")
player = rhythmbox.object("/org/gnome/Rhythmbox/Player")
player.introspect
player.default_iface = "org.gnome.Rhythmbox.Player"

pidgin = bus.service("im.pidgin.purple.PurpleService")
purple = pidgin.object("/im/pidgin/purple/PurpleObject")
purple.introspect
purple.default_iface = "im.pidgin.purple.PurpleInterface"

player.on_signal("playingUriChanged") do |uri|
  status = purple.PurpleSavedstatusFind("rhythmbox").first
  if status == 0
    status = purple.PurpleSavedstatusNew("rhythmbox", 2).first
  end
  purple.PurpleSavedstatusSetMessage(status, uri.to_s)
  purple.PurpleSavedstatusActivate(status)
end
```

Conclusion

Now that you know how to perform D-Bus calls and how to handle D-Bus signals, you can get started automating tasks on your desktop. If you are a Linux power user, D-Bus definitely should be in your vocabulary.

There is much more to D-Bus than I could show you in this article, but with qdbusviewer, qdbus, dbus-send and dbus-monitor, you can explore the possibilities yourself. If you want to create some more complex automation tasks with D-Bus, the Python and Ruby programming languages are a good fit. Consider the tutorials mentioned in the Resources, and then just let your imagination flow.

If you're a software developer, the part I didn't cover here is registering a service. This is the other side of the D-Bus story. If you register a service, you can provide objects to other applications via D-Bus.■

Koen Vervloesem has been a freelance journalist since 2000, writing mainly about free and open-source software. He has Master's degrees in computer science and philosophy and cannot decide which is the most interesting domain. In the meantime, he likes to think "I code, therefore I am."

Resources

D-Bus: **www.freedesktop.org/wiki/Software/dbus**

D-Bus Bindings:
www.freedesktop.org/wiki/Software/DBusBindings

Python D-Bus Tutorial:
dbus.freedesktop.org/doc/dbus-python/doc/tutorial.html

Ruby D-Bus Tutorial:
trac.luon.net/data/ruby-dbus/tutorial/index.html

The first and only radio show broadcast in the USA dedicated exclusively to spreading the word about the LINUX OPERATING SYSTEM and FOSS.

GUTSY GEEKS COMPUTER SHOW

gutsygeeks.com

NOV. 7–12 2010

24TH LARGE INSTALLATION SYSTEM ADMINISTRATION CONFERENCE

SAN JOSE CALIFORNIA

LISA '10

UNCOVERING THE SECRETS OF SYSTEM ADMINISTRATION

PROGRAM INCLUDES:

UNRAVELING THE MYSTERIES OF TWITTER INFRASTRUCTURE, LEGAL ISSUES IN THE CLOUD, AND HUGE NFS AT DREAMWORKS

STAPLE HERE

INSERT

SPONSORED BY USENIX in cooperation with LOPSA & SNIA

KEYNOTE ADDRESS BY Tony Cass, *CERN*

**JOIN US FOR 6 DAYS OF PRACTICAL TRAINING ON TOPICS INCLUDING:

*6-day Virtualization Track by instructors including John Arrasjid and Richard McDougall
*Advanced Time Management: Team Efficiency by Tom Limoncelli
*Dovecot and Postfix by Patrick Ben Koetter and Ralf Hildebrandt
*5-day Linux Security and Administration Track

**PLUS A 3-DAY TECHNICAL PROGRAM

Invited Talks
Refereed Papers
Workshops
Vendor Exhibition

REGISTER BY OCTOBER 18 AND SAVE
www.usenix.org/lisa10/lj

Controlling Your Linux System with a Smartphone

Using simple Web technologies, you can turn your smartphone into a multipurpose device to control your computers.

Phone technology has come a long way recently. The gap between personal computers and handheld devices is becoming smaller. I keep hearing the phrase "death of the PC", and there may be some truth to this statement. However, I believe many of us will continue to need access to larger and more powerful computers that are too big to fit in our pockets. To me, the best of both worlds is to have full control over a larger computer from my phone.

Many new smartphones have advanced Web browsers built in. With this technology, you can access an interface configured to run any command on almost any computer. It is fairly trivial to run a Web server on a Linux box. If you take the appropriate security measures, you quickly can build a Web interface designed specifically for handheld devices.

JAMIE POPKIN

Security

The approach shown in this article is to use a user account to run commands on the system. Of course, there are security concerns in doing this, but with the appropriate precautions, it can be made reasonably secure.

The system will rely on Wi-Fi. This makes sense when dealing with handheld devices, so configure your Wi-Fi router with a password. Users that want to connect to the local intranet will have to enter a password into their device before seeing anything. Most devices will remember the credentials and connect automatically once in range.

To minimize the risks in the event of a security breach, let's also create the user account with minimal permissions. This is a good safety measure, even though the interfaces will expose only "safe" commands.

Setup

Install the following from your distribution's repository if not already installed: Apache2, apache2-suexec and libapache2-mod-perl2.

The first package is the Web server. If it doesn't start automatically after the install, run the command:

```
/etc/init.d/apache2 start
```

The second package allows you to run the Web server with the credentials of a particular system user. When it is installed, you need to issue the following command as root to enable the apache module:

```
a2enmod suexec
```

Some of the examples presented here require Perl CGI interoperability. The last package is needed for that.

Now, you need to configure Apache to run as your little-trusted user. On our family Linux box, I created an account for all the kids. The user name is "saturn". This account can do things like play music and watch videos. However, it doesn't belong to any groups that can delete or change things of importance. So let's use this account as an example.

Edit your apache config, and add the following line to the default VirtualHost (*:80) or to the VirtualHost you want to use:

```
SuexecUserGroup saturn saturn
```

Apache runs as root, so it has the ability to run scripts as any user. The line above tells Apache to run as the user saturn and group saturn.

Now, restart Apache, also as root, with this command:

```
/etc/init.d/apache2 restart
```

The Web service now is running as the user saturn.

The Simplest Example

Playing a sound file from the command line is fairly trivial, and it's a good way to exhibit the simplicity of this setup—one button for one action.

I'm using a traditional Web stack: HTML, CSS, JavaScript and CGI. The CGI part can be accomplished with a number of different languages. This first example uses a shell script, for the sake of simplicity.

Create an index.html file in the root Web directory. For many systems, this is located in /var/www/. Some systems use /var/www/html/. In this file, add a button that calls a JavaScript function called playQuack():

```
<button id="quack-button" onclick="playQuack()">Quack</button>
```

The JavaScript for the playQuack() function is in bonkers.js. The entire index.html file looks like this:

```
<html>
  <head>
    <title>Bonkers</title>
    <meta name="viewport"
          content="initial-scale=1.0; user-scalable=no"/>
    <link rel="stylesheet" type="text/css" href="default.css" />
    <script type="text/javascript" src="bonkers.js"></script>
  </head>

  <body>
    <button id="quack-button" onclick="playQuack()">Quack</button>
  </body>
</html>
```

Some additional content worth mentioning is in the meta-tag. This tells smartphones not to scale the content of the page. Without this, the button would be very tiny on the screen.

Here is my default.css file. It defines a background color and specifies how the button will look:

```
html, body {
  background-color: #1E1E26;
}

button#quack-button {
  position: absolute;
  top: 20%;
  width: 70%;
  left: 15%;
  padding: 5px;
  border-width: 3px;
  color: #BFBFBF;
  font-size: 34px;
  font-weight: 800;
  border-color: #9C9C9C;
  background-image: -webkit-gradient(linear, left top
      left bottom, from(#BF5A34), to(#463630));
      -webkit-border-radius: 10px;
}
```

Many mobile browsers are starting to support WebKit CSS. This is exciting, because a couple lines of WebKit code can do some really fancy things. The last two lines before the last curly bracket tell the button to have rounded corners and a color gradient background.

Now you have a nice-looking button. Point the browser on your phone to the IP address of the Linux computer. You should see something similar to Figure 1.

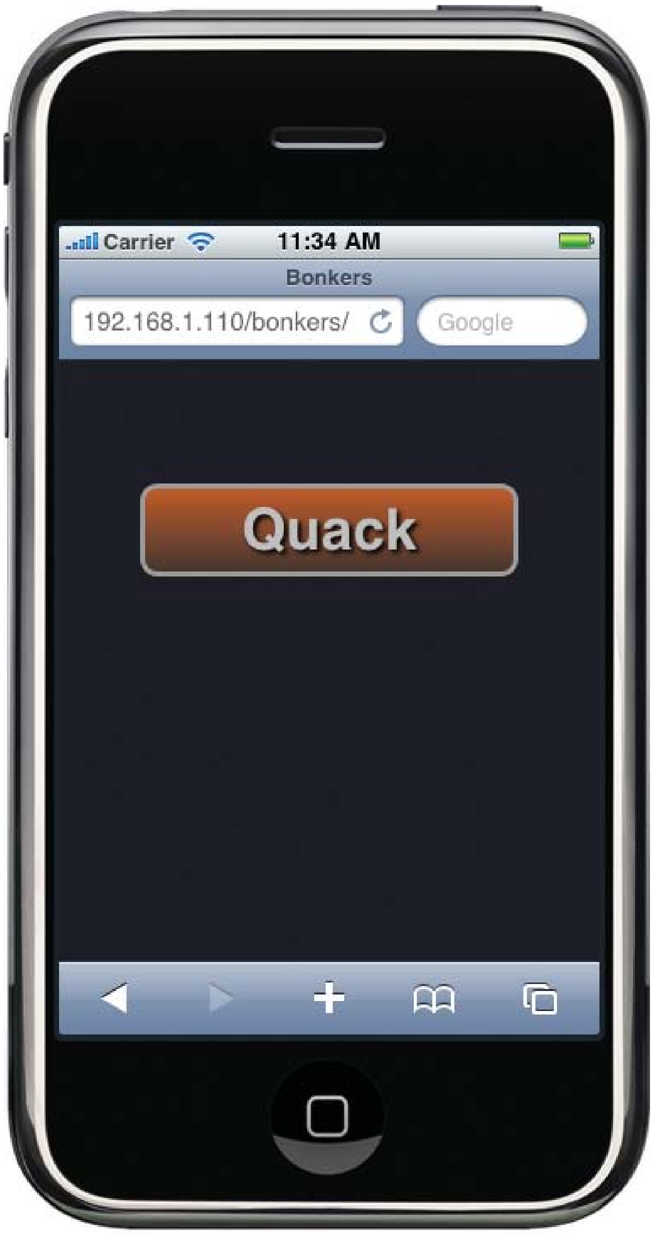


Figure 1. A Simple Button Displayed on an iPhone

Next, let's make the button actually do something. Create the bonkers.js file in the root Web directory, and enter the following:

```
var myDomain = document.domain;
var cgiURL   = "http://" + myDomain + "/cgi-bin/bonkers.cgi";
var xmlRequest;

function playQuack() {
  xmlRequest = new XMLHttpRequest();
  xmlRequest.open("GET", cgiURL, true);
  xmlRequest.send(null);
}
```

This is the JavaScript that forms the client-side process. It creates a URL that essentially runs the CGI script on your Linux box. In this example, you really don't care about the return value from the CGI script.

Believe it or not, the hard part is all done. CGI scripts are extremely simple and easy to understand—especially for someone who is used to the command line.

All CGI scripts must be located in the cgi-bin directory. This commonly is located in /var/www/cgi-bin or /usr/lib/cgi-bin and is also configurable within Apache.

Here is the CGI script, bonkers.cgi:

```
#!/bin/bash

mplayer ~/quack.wav &
```

That's all. This is a Bash shell script. A reference to the shell path is at the top. Below that is a command to run MPlayer, which plays an annoying quack sound. You essentially can place any shell command here.

There you have it. Anybody with a smartphone and the Wi-Fi password can make quack sounds come out of the computer. Now it's time to do something a little more useful.

Big Meanie

My wife and I have four kids. All of which compete for time on our family Linux box. The computer, which is named Saturn, is constantly logged in with the account "saturn". All kids use this account and are either watching TV, videos, YouTube, listening to music or playing Flash games on the Internet. I was getting sick of kicking the kids off the computer at supper time. This is why the interface "big-meanie" was created.

This next application scales up the previous example. I chose to use Perl as the CGI language, mostly because I am a longtime Perl user and fan. In the following example, I omit the HTML and CSS code because it's not significantly different from the code in the first example.

The main difference being that this example has three buttons: one is for starting a five-minute countdown until all fun is over. Another is to stop the countdown, just in case I have a change of heart. The last is when I really mean business and want to terminate all applicable programs immediately.

Figure 2 shows the layout of Big Meanie.

The first button gives the kids a five-minute countdown to get off the computer. I thought it appropriate to have visual indicators pop up over the current windows on the desktop. You need to set a few shell variables in order to output to the local display: XAUTHORITY, HOME and DISPLAY. The following Perl commands accomplish this:

```
$ENV{'XAUTHORITY'} = '/home/saturn/.Xauthority';
$ENV{'HOME'} = '/home/saturn';
$ENV{'DISPLAY'} = ':0';
```

This allows windows to open on the local display, even though the command was instigated remotely.

The next thing to note is that you have more than one function to perform, so you need to inform the script which action to perform (that is, which button was pressed). Since you're are dealing with Web URLs here, appending your parameter to the URL is the easiest way to accomplish this. Using a ? sign tells your script that the following text is a parameter. That parameter will be the action to perform.

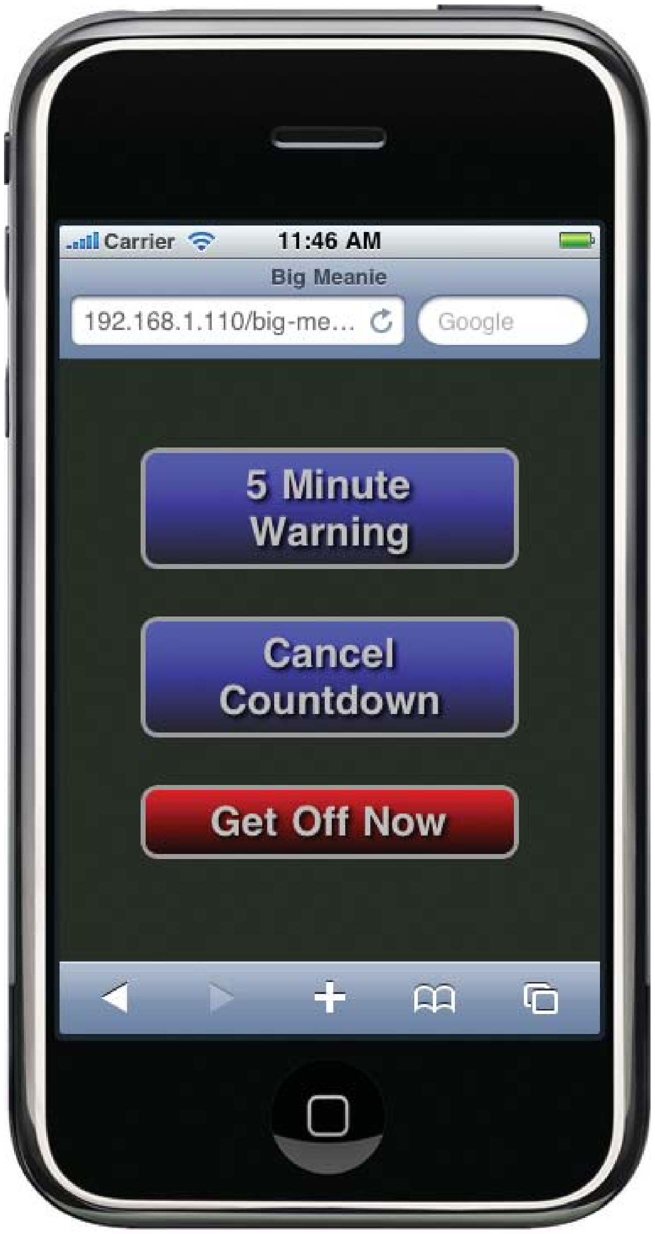


Figure 2. Big Meanie Displayed on an iPhone

Here is the full JavaScript file for Big Meanie:

```
var myDomain = document.domain; // Grab current domain.
var cgiURL  = "http://" + myDomain + "/cgi-bin/big-meanie.cgi";
var xmlRequest;

// This is the logic for the "5 Minute Warning" button
function warn5() {
  xmlRequest = new XMLHttpRequest();
  // Add the 5min variable to the URL
  xmlRequest.open("GET",cgiURL + "?5min", true);
  xmlRequest.send(null);
}

// This is the logic for the "Cancel Countdown" button
function cancel() {
```

Advertiser Index

CHECK OUT OUR BUYER'S GUIDE ON-LINE.
Go to www.linuxjournal.com/buyersguide where you can learn more about our advertisers or link directly to their Web sites.

Thank you as always for supporting our advertisers by buying their products!

Advertiser	Page #	Advertiser	Page #
1&1 Internet, Inc. www.oneandone.com	C2, 1	MBX www.mbx.com	33
Aberdeen, LLC www.aberdeeninc.com	3	Microway, Inc. www.microway.com	C4, 13
ASA Computers, Inc. www.asacomputers.com	77	Mikro Tik www.routerboard.com	21
Cari.net www.cari.net	71	Ontario Linux Fest oglf.ca	78
Emac, Inc. www.emacinc.com	43	Polywell Computers, Inc. www.polywell.com	79
EmperorLinux www.emperorlinux.com	29	ServerBeach www.serverbeach.com	47
FSCONS www.fscons.org	76	Servers Direct www.serversdirect.com	9
Genstor Systems, Inc. www.genstor.com	59	Silicon Mechanics www.siliconmechanics.com	25, 31
Gutsy Geeks www.gutsygeeks.com	60	SuperComputing SC10 sc10.supercomputing.org	69
Intel www.intel.com	5	SuperMicro Computer, Inc. www.supermicro.com	7
iXsystems, Inc. www.ixsystems.com	C3	Technologic Systems www.embeddedx86.com	17
Logic Supply, Inc. www.logicsupply.com	55	USENIX LISA www.usenix.org/lisa10/lj	61
Lullabot www.lullabot.com	67	Zend/PHP Conference & Expo www.zendcon.com	75

ATTENTION ADVERTISERS

February 2011 Issue #202 Deadlines
Space Close: November 22; **Material Close:** November 30

Theme: Desktop

BONUS DISTRIBUTIONS: Black Hat DC, PyCon

Contact **Joseph Krack, +1-713-344-1956 ext. 118, joseph@linuxjournal.com**

Listing 1. Server-Side Script

```
#!/usr/bin/perl

# Grab the URL variable
$variable = $ARGV[0];

# Set some important environment variables.
$ENV{'XAUTHORITY'} = '/home/saturn/.Xauthority';
$ENV{'DISPLAY'}    = ':0';
$ENV{'HOME'}       = '/home/saturn';

# if the 5 minutes warning button is pushed
if ($variable =~ /^5min$/) {
    # Lets create a child process to deal with the notifications
    defined(my $childpid = fork);

    if ($childpid) { # If a child pid exists... this is the parent
        # Send response so the browser is happy
        print "Content-type: text/html\n\n";

        # Show a popup warning message
        # displaying 5 minutes remaining.
        `zenity --warning --text='5 minutes left to play'`;
    } else { # Otherwise it's the child
        # Print the amount of time left with a subtle gnome
        # notification message.
        sleep(60);
        `notify-send '4 minutes left to play'`;
        sleep(60);
        `notify-send '3 minutes left to play'`;
        sleep(60);
        `notify-send '2 minutes left to play'`;
        sleep(60);
        `notify-send '1 minutes left to play'`;
        sleep(60);

        # We are now out of time.
        # Let's close all the fun applications
        `/usr/bin/tvtime-command quit`; # Close the TV
            `pkill mplayer`;                # Close mplayer
            `pkill totem`;                  # Close Totem movie player
            `pkill rhythmbox`;              # Close the music player
            `pkill firefox`;                # Close the web browser
    }
}

# If the Cancel Countdown button has been pushed.
if ($variable =~ /^cancel$/) {
    defined(my $childpid = fork);

    if ($childpid) { # If parent
        print "Content-type: text/html\n\n";
    } else {
       `pkill big-meanie`;
    }
}

# If the Get Off Now button has been pushed.
if ($variable =~ /^off-now$/) {
    defined(my $childpid = fork);

    if ($childpid) { # If parent
        print "Content-type: text/html\n\n";
    } else {
        `/usr/bin/tvtime-command quit`;   # Close the TV
        `pkill mplayer`;                  # Close mplayer
        `pkill totem`;                    # Close Totem movie player
        `pkill rhythmbox`;                # Close the music player
        `pkill firefox`;                  # Close the web browser

         # Just because we can... Send out a tweet that
         # someone has been kicked off the computer
         `curl -u <twitterUser>:<twitterPassword> -d
            status='Someone has been rudely kicked off the computer.'
            http://twitter.com/statuses/update.xml > /dev/null`;
    }
}
```

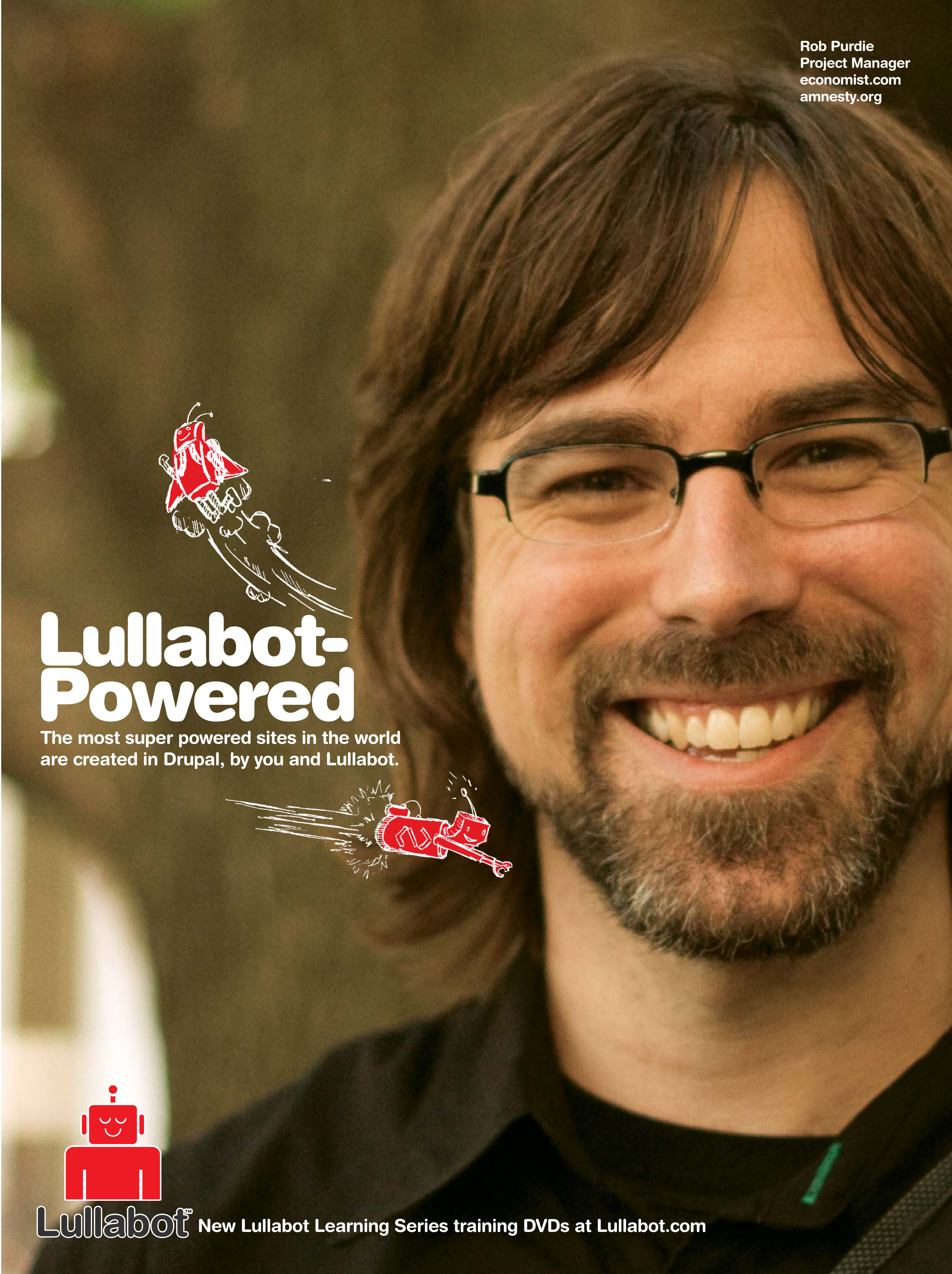
```
  xmlRequest = new XMLHttpRequest();
  // Add the cancel variable to the URL
  xmlRequest.open("GET",cgiURL + "?cancel", true);
  xmlRequest.send(null);
}

// This is the logic for the "Get Off Now" button
function offnow() {
  xmlRequest = new XMLHttpRequest();
  // Add the off-now variable to the URL
  xmlRequest.open("GET",cgiURL + "?off-now", true);
  xmlRequest.send(null);
}
```

Each button has its own JavaScript function. The URL to the CGI script is appended with the appropriate variable within each function. The variable then is passed to the CGI script with the xmlRequest.send() function.

All action up until now has been happening on the phone browser. Now, let's dive into the script on the server, big-meanie.pl.

There is one last important thing to mention before going over the server-side script (Listing 1). Internet browsers expect a response from sites when making a URL request. The browser hangs if there is no response. You can get around this by creating a second thread that responds automatically with an empty message. The browser doesn't get stuck when the server is executing the command this way, which is important for what you're about to do.

The action variable from the URL is grabbed through the $ARGV[0] array. The if statements that follow test which variable has been specified and, therefore, which action to execute. The first action, `5min`, executes a five-minute warning by forking the CGI script (creating a copy of the process). The "parent" branch of the fork executes the if part of the if statement, and the "child" branch executes the else part. The parent sends an empty HTML



Rob Purdie
Project Manager
economist.com
amnesty.org

Lullabot-Powered

The most super powered sites in the world are created in Drupal, by you and Lullabot.

Lullabot™ New Lullabot Learning Series training DVDs at Lullabot.com response to the browser so the client doesn't hang and pops up a five-minute warning message box using Zenity. Zenity provides a simple GUI interface that can be accessed from scripts, and it should be installed by default on most GNOME desktops. There are alternatives for KDE users, such as kdialog or whiptail. The advantage of putting the pop-up dialog in a separate thread from the countdown notifications is the script doesn't stall if the current user doesn't click the OK button on the Zenity window.

The child portion of the fork does the actual five-minute countdown. I wanted to be a little humane here and provide subtle notifications of how much time is left before the user is kicked off. This allows the older kids to save their session or work. My younger kids just need to be eased into the idea of stopping.

The child part of the script sleeps for 60 seconds after the dialog appears. A simple notification then is sent to the desktop once the child wakes up. It appears as a standard GNOME notification in the upper right of the screen. Figure 3 shows how this appears to the user. This procedure repeats until the countdown is complete. Next, the meanie part of Big Meanie kicks in and sends a kill signal to any fun type of program the kids could be enjoying.

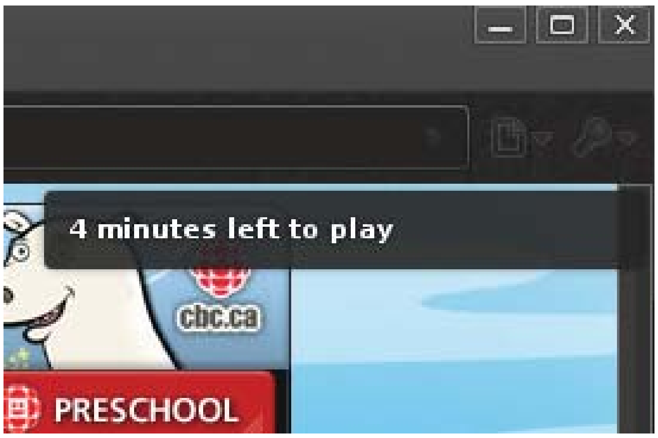

4 minutes left to play

Figure 3. GNOME Notification on the Desktop Displaying the Time Remaining

I used the notify-send command commonly installed with GNOME. It is included in the libnotify-bin package, and it should be in most distribution repositories. KDE users can use kdialog with the --passivepopup flag instead of notify-send.

The second if statement is for the cancel countdown button. It sends a kill signal to all instances of big-meanie, essentially killing itself and all sleeping instances, which will kill any previously started five-minute wait child processes that are waiting to shut down things.

The last if statement doesn't need much explanation. It kicks the user off immediately and is almost identical to the first, excluding the countdown logic. You already may have noticed the juicy little addition at the very end of the code: big-meanie sends a status update to Twitter stating somebody has been kicked off the computer. It seemed appropriate for big-meanie to gloat to the whole world that someone's fun has been terminated. It also acts as a safety feature informing me if one of the kids has discovered the program and is possibly using it in any unfriendly fashion on fellow siblings.

Other Possibilities

The most-used application I have built is a television remote control. We use the fantastic tvtime program to watch all our TV. The tvtime package includes the `tvtime-command` command that allows you to control every conceivable function of the running tvtime instance. For example, `tvtime-command chan-up` changes the channel up one increment. Full documentation can be found in the tvtime man page.

I used all the same techniques described earlier and built a full functioning remote shown in Figure 4.

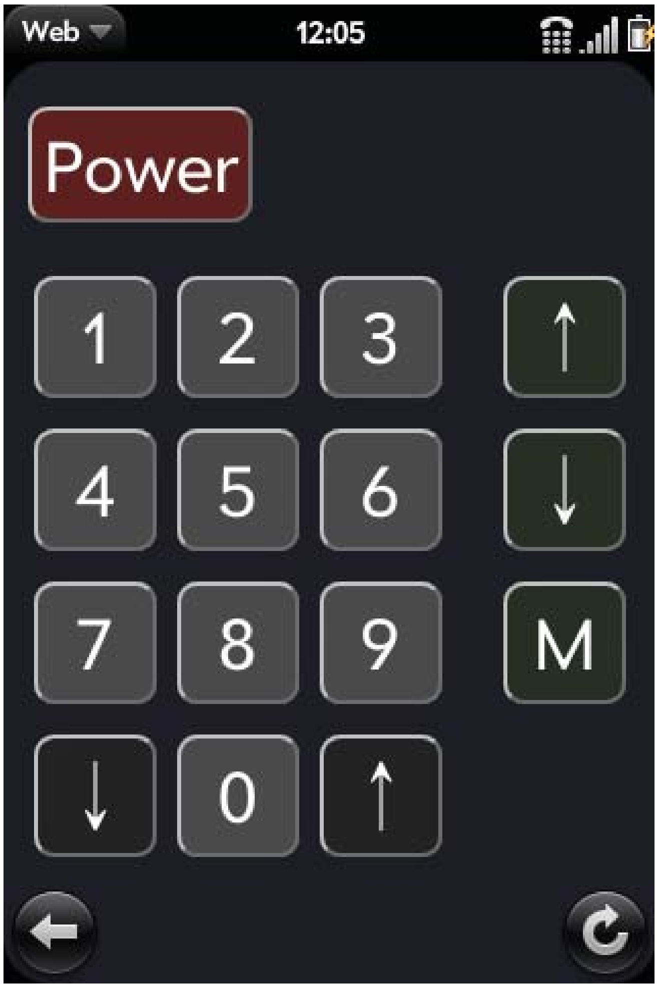


Figure 4. Remote Control Application Shown on a Palm Pre

Now friends and family can control the TV from the comfort of their very own smartphones. There are no more complaints about using our giant Bluetooth keyboard as a remote either. The kids simply can grab the closest iPod Touch, Palm Pre or whatever.

Conclusion

Smartphone technology is advancing extremely rapidly. Without modifying the phone, you can access almost any possible function on a remote computer through the built-in browser. Linux users are especially lucky. It doesn't take much effort to turn a Linux computer into a back end for a phone. Hopefully, this article inspires you to come up with your own unique and interesting uses.■

Jamie Popkin lives in Lantzville, British Columbia, with his wife and four kids. He is a consultant specializing in geographic data portrayal on the Web. Recently, he has started developing for smartphones, utilizing modern Web/HTML5 technology. He can be reached via Twitter (@jamiepopkin) or e-mail (popkinj@littleearth.ca).

The International Conference for High Performance Computing, Networking, Storage, and Analysis

SC10
New Orleans, LA

SC10.supercomputing.org

Sponsors:
IEEE Computer Society
ACM SIGARCH

IEEE computer society

acm Association for Computing Machinery

SC10 – The Global Supercomputing and Networking Community Gathers in New Orleans, Louisiana

Every November, over 10,000 of the leading experts in high performance computing, networking, analysis and storage come together for seven days to present their accomplishments, share technical expertise and learn about the latest developments from industry.
In 2010, this meeting known as the SC10 Conference which will be held Nov. 13-19 in New Orleans, Louisiana. Plan now to be a part of the world's largest gathering of the global supercomputing and networking community as we explore The Future of Discovery.

Ernest N. Morial Convention Center • New Orleans, Louisiana
Exhibition Dates: November 15-18, 2010 • Conference Dates: November 13-19, 2010

Rockbox

People unfamiliar with Rockbox often make the assumption that because it's an open-source project providing an operating system (or firmware) for digital audio players, it must be based on GNU/Linux. In this article, I set the record straight and tell you what Rockbox really is, and why you might be interested in it. BRYAN CHILDS

Let me start by disabusing you of the notion that Rockbox is Linux of some flavor, or that it started out that way. It's not. You are, of course, reading *Linux Journal*, so you could be forgiven for thinking it is, but I just want to make sure you are aware right from the get-go that it isn't. With that small formality out of the way, let me get on to what it actually is, which is a replacement firmware for a growing number of digital audio players.

Back in 2001 or so, Rockbox started life as a replacement firmware for just one digital audio player (DAP), which was the Archos Jukebox (sometimes referred to as the Archos Player, Figure 1). A few owners of this very early device were left feeling dissatisfied with the buggy firmware that came with the device, and also felt it was lacking some important features. Those folks decided they could do better, and so Rockbox was born. That original device was very limited in what it could be made to do when compared with modern DAPs. It featured a very slow 11MHz CPU and, therefore, had to include a dedicated chip for decoding MP3 audio. This meant that in the early days of Rockbox, the only music format it could play was limited to MP3 or uncompressed PCM Audio (WAV). These days, that list has expanded greatly, and the firmware now plays some 27 formats. Those of us associated with the project are fairly confident (not to say a little smug) that this is way more than any other firmware made by anyone, be it open or closed, anywhere in the world.

Of course, in addition to now supporting this rather large and, in not a few cases, esoteric list of formats, Rockbox runs on a vastly larger variety of players these days too. The list of supported DAPs has grown steadily over the years, with the iRiver H100 series being the first software codec target that was added sometime in 2004, and from there growing to include other

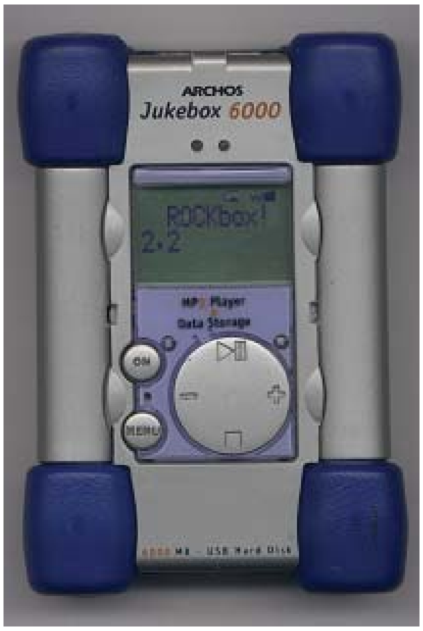


Figure 1. Original Archos Player (Photographer: Björn Stenbeg)

players made by Apple, Archos, Cowon, iRiver, Logick, Meizu, MPIO, Olympus, Onda, Packard Bell, Philips, Samsung, SanDisk, Tatung and Toshiba. Among these players, the architectures on which Rockbox has to run necessarily has been extended too, from the original SH1 in the Archos player, to ColdFire, ARM and even MIPS hardware.

Porting Rockbox to new players is a time-consuming business, and doing so varies from just plain hard to nearly impossible. Where a player falls in this scale of difficulty depends on a number of factors, but the main influences tend to be how easy the manufacturer has made it to get custom code running on the device and how much documentation exists for the underlying hardware. On a great many of the players on which Rockbox runs, reverse engineering of the hardware has been done purely by Rockbox hackers; however, a large hat tip in the direction of the iPodLinux (IPL) Project is due, without whose extensive reverse engineering of the early range of iPods running on PortalPlayer hardware, the current Rockbox port to those players probably would not have been possible.

The PortalPlayer platform is notorious among embedded hackers as having no publicly available documentation, so each and every piece of functionality on those devices using it has had to be painstakingly worked out, and the IPL hackers spent a great deal of time doing this. Coming back to the first point, about how easy it is to run custom code at all, the more recent iPods (since the second generation of the iPod Nano, in fact) have raised the bar still higher, because the device requires the code it executes to be both signed and encrypted in hardware before it will run. Back when Apple launched this device, hope was very low in the Rockbox community that it ever would be possible to get a port running on it. However, a flaw in the original firmware turned up last year that allowed some very talented folks to get code running, and now there's a port of Rockbox running on that device too. Other platforms have proved somewhat easier—for instance, the Toshiba Gigabeat F had hardware for which documentation was relatively simple to come by.

Rockbox's driving philosophy always has been to provide the best portable music player experience that it can. As mentioned previously, 27 codecs is just part of that. Although it allows end users to navigate their music collection via a database based on the music's metadata content these days, it originally started with just a file browser—a feature that remains a firm favorite with the core of the Rockbox Project team. Every action to play back music requires a playlist, as this is how Rockbox works. Creation of playlists can be done on the fly where the end user isn't even aware it has happened. Or, they can be constructed manually either within Rockbox itself or in the user's favorite music management program, assuming it can generate standard M3U format files.

Another key feature the firmware offers is bookmarking—invaluable for those who listen to audio books and need to keep track of where they are within the file when they shut down their players. Rockbox allows users to keep many such bookmarks, letting them skip around between multiple files with ease.

One of the oldest features in Rockbox, which has only improved over time, is the customizable While Playing Screen (WPS). In the very early text-only versions of Rockbox, this allowed users to choose what information about the currently playing tune they wanted displayed. In the most recent versions of Rockbox, this now allows users to decide where, or indeed if, they want Album Art displayed, along with all the various pieces of metadata, or have a colorful backdrop and pretty much anything they want (Figure 2).

Another favorite feature, particularly with Rockbox users who are sight-impaired, is the implementation of a voice feedback system. Once this is turned on, all menus within the Rockbox interface announce themselves to the user, so the entire interface can be navigated without needing to see the screen. In its default state, this also allows the player to spell out filenames, so users can navigate their music collection. This can be enhanced further with the use of speex format .talk files, which are pre-generated with a PC-based tool and allow the player actually to speak the names of files and directories.

A somewhat more recent feature, added in the last couple years, has brought movie playback to a large quantity of supported Rockbox players. Once you have transcoded your movie of choice and resized it to fit on the device's screen, you can watch MPEG1 and MPEG2 movies at your leisure, all made possible by Rockbox's rich plugin API.

The features I've highlighted here barely scratch the surface of what is possible with Rockbox, and if you are at all interested in discovering more, check out the rather comprehensive manual available on the Rockbox home page (**www.rockbox.org**). The project couldn't possibly provide all the features it does without standing on the shoulders of a number of other open-source giants. A great deal of the codecs that Rockbox uses are derived from other open-source codebases. The folks that provide that code have helped by providing countless hours of their own work that Rockbox has built on. Libmad, ffmpeg, flac, vorbis, speex, wavpack, libmpeg2 and libmtp are just a few of the other teams that all deserve the heartfelt gratitude of the current Rockbox user base, because without them, virtually all of their favorite music wouldn't play back as well on the Rockbox platform.

Rockbox has been fortunate enough to have been accepted to the Google Summer of Code program during the last few years, and this year is no exception.

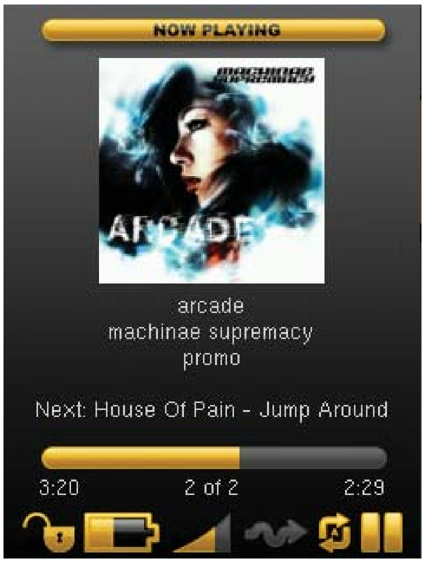


Figure 2. The While Playing Screen Used by Rockbox's Default Theme

1990s
- WWW
2000s
- Web 2.0
- Social Media
2010s
- CLOUD . . . . .

got cloud?

Well, why not? Streamline your business operations utilizing the always-on, *redundant* and fully *scalable* cloud architecture.

CariNet's "**Starter Cloud**" running 3tera's® AppLogic™ Cloud OS includes 5 one-on-one training sessions with our cloud-certified experts to get you up to speed.

Cloud is no longer just a buzzword, but is here to stay. Don't get left behind. Find out what the excitement is all about risk-free with CariNet: *the Cloud Computing Specialists.*

$500 per month
No Contract, No Risk!

carinet SAS70 Certified

www.cari.net/LJ
888.221.5902

Perhaps one of the most interesting projects that is being kindly funded by Google's deep pockets is that of "Rockbox as an Application" or RaaA.

Increasingly, consumer electronics devices that are capable of music playback are shipping with very capable operating systems already, and it makes little sense to replace an entire operating system just to get superior music playback on such a device. Because there has been code in the Rockbox repository for a long time to build a "Rockbox Simulator", which runs on a PC using SDL, an opportunity to turn this into a real application that could execute on Android, Symbian, Maemo or even Windows Mobile (god forbid!) already existed, and now is being put into action. The aim of this project is to separate the playback engine from the firmware portion of the code, allowing you to install Rockbox on your phone without sacrificing its ability to make phone calls or surf the Web. It's thought that this is a better idea than trying to add a network stack and all the other goodies you'd need to make Rockbox a convincing alternative OS on your mobile phone. In actual fact, someone forked Rockbox's simulator a few years ago to the Motorola RockR series of phones in just this way, but sadly, the folks responsible for this port never gave the project any decent code back, and that particular set of targets remains unsupported in the trunk of the Rockbox codebase.

The majority of Rockbox's codebase is written in C, but there's a smattering of routines written in assembly code where it makes sense to squeeze the absolute last ounce of performance out of the player. Most of those routines are in the codecs, where every clock cycle counts, especially with some of the heavyweight formats like Monkey's Audio, which even the most powerful Rockbox players struggle with at maximum compression.

The Rockbox kernel has been written from the ground up, and among its features there are two things that probably will stand out as substantially different to people who are familiar with other modern OS kernels. First, because the vast majority of Rockbox-capable DAPs do not possess an MMU, Rockbox's kernel does not feature any method of dynamic memory allocation. This has been a contentious issue in the Rockbox community for a long time now, and new developers constantly bring it up. The lack of MMU is, of course, not a complete showstopper. It's perfectly possible to come up with a malloc implementation that does not require an MMU, and this is often the point they make. The reason the project has resisted it historically, and continues to do so to this day, is that any implementation requires a pool of free memory from which to allocate. The Rockbox old guard all argue that this is a waste of precious resources on the often extremely memory-constrained platforms on which Rockbox executes, and that every piece of available memory should be used to buffer as much music as possible. This means people coming to programming for Rockbox for the first time have to get used to thinking about statically allocating all the memory they need at compile time, a mode of programming foreign to most developers these days.

The second interesting feature in the Rockbox kernel is the threading model, which is cooperative rather than preemptive.

The second interesting feature in the Rockbox kernel is the threading model, which is cooperative rather than preemptive. This is mainly a historical artifact these days, and there has been repeated discussion about moving to a preemptive model, which so far has failed to reach a conclusion. In the meantime, this means, again, that anything designed to run on the Rockbox kernel has to have its execution thought about carefully and must consciously yield to other threads at opportune moments if it is to play nicely with the rest of the system.

As highlighted previously, Rockbox provides a rich plugin API that can be used via either compiled C code or in up-to-date versions via a growing library of LUA functions. A few support programs also go with Rockbox, not the least of which is the Rockbox Utility, a multiplatform installation tool. This is written in C++ using the Qt toolkit, and a version for all three major PC operating systems is maintained in statically linked form to make using it as easy as possible. There are some 92 people currently listed as having commit access to the central Subversion repository for Rockbox, with 40 of those active at the current moment. In addition to them, there is a sizable community of patch authors, artists and general helpers contributing to both the project's health and also to its vibrant and friendly atmosphere.

For those not skilled with a text editor and compiler, additions to and maintenance of the documentation are always welcome. Although it would be foolish to claim that the documentation is perfect, most of those involved in the project would say that as far as most open-source projects go, the documentation is pretty good. In addition to the manual and the wiki, which provide the static documentation for the project, there are numerous other ways for people to find and receive help. There are two active mailing lists, one aimed at the end user, which has proved to be perhaps the most popular support option for the community of blind users, and another where those who are interested in Rockbox development can exchange ideas. The Rockbox forums also are extremely active, and indeed, a lot of the Rockbox "staff" have made their entrance into the community by helping out others here. As with most other open-source projects, there is also an extremely active IRC channel on the Freenode network (**www.freenode.net**)—#rockbox is active for most of the day due to the community's diverse geographical (and some might say insomniacal!) nature. The channel is logged to make sure all discussions are searchable in case a developer misses something critical or otherwise interesting, and these logs are published on the project's Web site. Also, never let it be said that the Rockbox community is slow to adopt new technology. Recently @rockboxcommits has made its debut on Twitter for those who are keen to keep track of every change in the source code repository!■

Bryan Childs is a systems manager at a financial services software house in London. He likes cocktails and behaving in a disreputable manner whenever he's out of the office. He can throw a frozen egg over a ten-story building.

Web Application Security Testing with Samurai

Web site vulnerabilities often occur in very non-obvious ways. Whether you're a Web developer or run a Web site, you need to understand how it's done and how to test your site. JES FRASER

Almost every week the media picks up on another case of sensitive data being retrieved from Web sites with bad security. Web application security never has been more important, yet many Web sites never have been audited for security in a meaningful way. Although careful application architecture can help minimize security risks in code, a complete approach to Web application security considers the entire life cycle of the application, from development to deployment. In order to test whether your Web site is truly secure, consider using the same tools attackers do.

Penetration testing is the art of assessing the security of a system by simulating an attack or series of attacks. The goal of the penetration test is not necessarily to exploit the system if a flaw is found, but to audit potential attack vectors stringently and provide data that can be used to evaluate the potential risk of an exploit, and to find a solution to secure the system.

The Samurai Web Testing Framework is a security-oriented distribution that focuses on penetration testing for Web applications. It includes a variety of graphical, command-line and browser-based tools to test for common Web vulnerabilities. It's available as a live CD image from **samurai.inguardians.com**.

In this article, I look at using Samurai to test for a couple of the top Web application security risks as defined by the Open Web Application Security Project (OWASP). This is not specifically a list of attack vectors. Technically, many of the risks listed below are exploited using various forms of SQL injection. Rather, this list was developed by OWASP combining "threat agents, attack vectors, weaknesses, technical impacts and business impacts...to produce risks". OWASP's top ten risks (**www.owasp.org/index.php/Top_10**):

1. Injection flaws.

2. Cross-site scripting.

3. Broken authentication and session management.

4. Insecure direct object references.

5. Cross-site request forgery.

6. Security misconfiguration.

7. Insecure cryptographic storage.

8. Failure to restrict URL access.

9. Insufficient transport layer protection.

10. Unvalidated redirects and forwards.

In the interests of keeping the scope of this article manageable, I focus on injection flaws and cross-site scripting.

Disclaimer: please do not try any of these examples on production Web sites. *Linux Journal* recommends that you set up a virtual environment with a copy of your Web site to test for vulnerabilities. Do not test over the Internet. Never use any of these examples on a Web site that is not yours. *Linux Journal* is not responsible for any damage to data or outages to services that may arise from following any of these examples.

Injection Flaws

Injection flaws occur when the application passes user input to an interpreter without checking it for possible malicious effects. Injection flaws can include operating system command injection, LDAP injection and injection of many other interpreters called by a Web application using dynamic queries. One of the most common injection vectors is SQL injection. Depending on the specific vulnerability, attackers could read passwords or credit-card numbers, insert data into the database that gives them access to the application or maliciously tamper with or delete data. In extreme cases, operating system files could be read or arbitrary system commands could be executed—meaning game over for the Web server.

Login forms are primary targets for SQL injection, as a successful exploit will give attackers access to the application. To start testing an application for SQL injection vulnerabilities, let's use some characters that have special meaning in SQL to try to generate an SQL error. The simplest test, using a single quote (') as the user name, failed to generate an error, so let's try a double quote followed by a single quote ("'):

```
SQL Error: You have an error in your SQL syntax; check the manual
           that corresponds to your MySQL server version for the
           right syntax to use near '"''' at line 1

SQL Statement: SELECT * FROM accounts
                 WHERE username='"'' AND password='"''
```

Not only is this form vulnerable to SQL injection, but also the error message has thrown up the exact SQL statement being used. This is all the information you need to break into this application, by using the following text in both the user name and password fields:

```
' or 1=1 --
```

This changes the original query to one that can match either the correct user name and password, or to test if 1=1. Because 1=1 will evaluate to true, the application accepts this as your login credentials and authenticates you. Even worse, assuming you're not attackers, once you're logged in, you now can see that you're the admin user. This is because the SQL query will look at each row, one by one, to see on which row the query returns true. Because you've tampered with it always to return true, MySQL will return the first row. Because the first user created is quite often the administrator or root user (Figure 1).

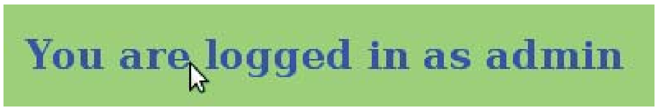

You are logged in as admin

Figure 1. The first row of the users table often contains the administrator user.

Not all applications are going to prove so easy to break into—particularly those that do not divulge as much information about the database and table structure in the error message. An auditing tool will let you iterate over a range of possible strings quickly. w3af, the Web Application Attack and Audit Framework, has a range of plugins to assist in scanning for and exploiting vulnerabilities, including SQL injection.

Launch the w3af GUI from the Applications→Samurai→Discovery menu. Enter the Web site URL you'd like to test in the Target: field, and then expand the options under discovery in the plugin box. Scroll down until you find the webSpider plugin, and check its box to enable it. In the pane to the right, the options for the webSpider plugin will be shown. Tick onlyForward, and select Save Configuration. Now, scroll back to the top of the plugins box, and expand Audit. Scroll down to sqli and check its box to enable it.

Once the scan is completed, you can look at the Results tab and see that w3af found seven separate places within the application that could be exploited by SQL injection (Figure 2).

Another tool included with Samurai that can discover SQL injection vulnerabilities is Grendel-Scan. Launch Grendel-Scan from

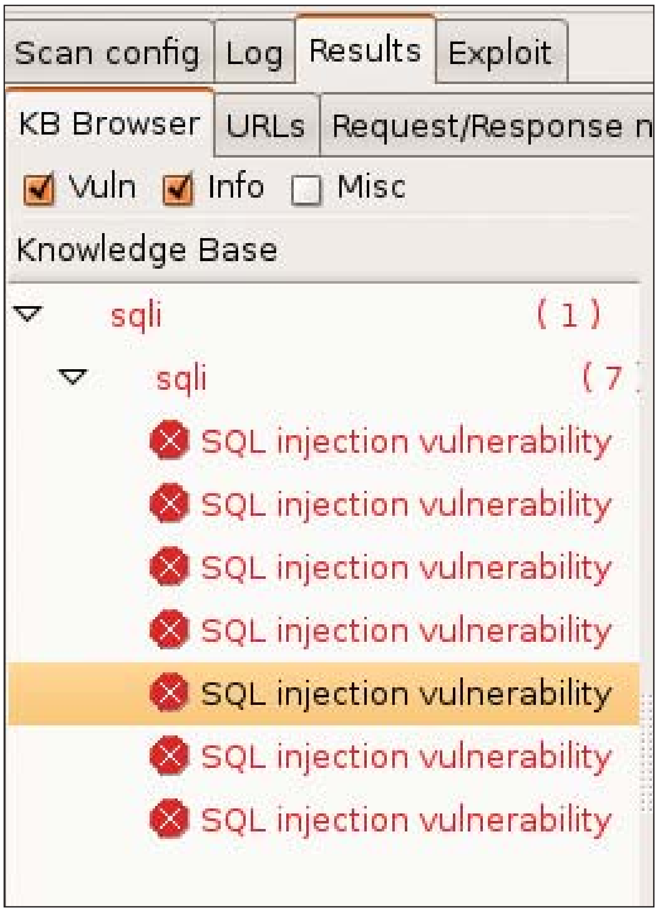

Scan config
Log
Results
Exploit
KB Browser
URLs
Request/Response n
Vuln
Info
Misc
Knowledge Base
sqli (1)
sqli (7
SQL injection vulnerability
SQL injection vulnerability
SQL injection vulnerability
SQL injection vulnerability
SQL injection vulnerability
SQL injection vulnerability
SQL injection vulnerability

Figure 2. w3af Results Tab, Showing Discovered Vulnerabilities

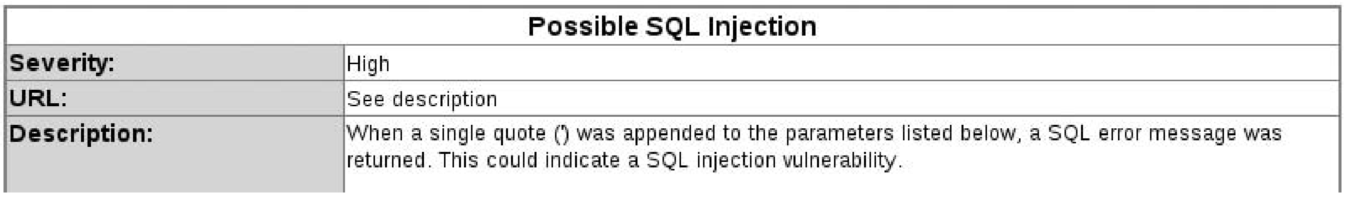
Statement of Ownership, Management, and Circulation

1. Publication Title: *Linux Journal*
2. Publication Number: 1075-3583
3. Filing Date: September 8, 2010
4. Issue Frequency: Monthly
5. Number of Issues Published Annually: 12
6. Annual Subscription Price: $29.50
7. Complete Mailing Address of Known Office of Publication: 2121 Sage Road, Ste 310 Houston, TX 77056 Contact Person: Mark Irgang 713-344-1956 x111
8. Complete Mailing Address of Headquarters of General Business Office of Publisher: 2121 Sage Road, Ste 310 Houston, TX 77056
9. Full Names and Complete Addresses of Publisher, Editor, and Managing Editor: *Publisher*: Carlie Fairchild PO Box 980985 Houston, TX 77098 *Editor*: Doc Searls PO Box 980985 Houston, TX 77098 *Managing Editor*: Jill Franklin PO Box 980985 Houston, TX 77098
10. Owner(s): Carlie Fairchild PO Box 980985 Houston, TX 77098 Joyce Searls PO Box 980985 Houston, TX 77098 Adele Soffa PO Box 980985 Houston, TX 77098
11. Known Bondholders, Mortagees, and Other Security Holders Owning or Holding 1 Percent or More of Total Amount of Bonds, Mortages, or Other Securities: None
12. Tax Status: Has not Changed During Preceding 12 Months

13. Publication Title: *Linux Journal* 14. Issue Date: October 2010

15. Extent and Nature of Circulation	Average No. Copies Each Issue During Preceding 12 Months	No. Copies of Single Issue Published Nearest to Filing Date
a. Total Number of Copies: (Net press run)	45,361	40,650
b. Paid and/or Requsted Circulation		
(1) Paid/Requested Outside-County Mail Subscriptions on Form 3541.	15,434	14,633
(2) Paid In-County Subscriptions Stated on Form 3541	0	0
(3) Sales Through Dealers and Carriers, Street Vendors, Counter Sales, and Other Non-USPS Paid Distribution	12,990	12,368
c. Total Paid and/or Requested Circulation	28,424	27,001
d. Free Distribution Outside the Mail		
(1) Outside-County as Stated on Form 3541	434	309
(2) In-County as Stated on Form 3541	0	0
(3) Other Classes Mailed Through the USPS	0	0
(4) Free Distribution Outside the Mail	3,833	2,555
e. Total Free Distribution	4,267	2,864
f. Total Distribution	32,691	29,865
g. Copies Not Distributed	12,670	10,785
h. Total	45,361	40,650
i. Percent Paid and/or Requested Circulation	86.9%	90.4%

PS Form 3526



Possible SQL Injection	
Severity:	High
URL:	See description
Description:	When a single quote (') was appended to the parameters listed below, a SQL error message was returned. This could indicate a SQL injection vulnerability.

Figure 3. A Section of Output from Grendel-Scan

the Applications→Samurai→Discovery menu. Under Base URLs, insert the URL of the Web application you would like to test and click Add. Untick Enable Internal Proxy. Let's use Grendel's Web spidering module instead. Under Scan Output, select a directory for Grendel-Scan to store its output report. This directory must not exist. The application will expect to create it. Select Start Scan from the Scan menu to start.

The scan will take some minutes to complete, depending on the size of the Web application. Once finished, navigate to the designated output directory to view the report. Here, report.html tells you that among other issues, a possible SQL injection vulnerability was found (Figure 3). The clean output of Grendel-Scan makes it a great tool to send attractive and easy-to-read vulnerability reports to upper management as part of a reporting requirement.

Cross-Site Scripting (XSS)

Cross-site scripting can occur when the application accepts user input and sends it back to the browser without validating it for malicious code. There are two types of XSS attacks: reflected and stored. A stored attack injects code into the page that is persistent and will be activated by the victim's browser when requesting the page. A common example of a stored cross-site scripting attack would be a script that is injected through a comment field, forum post or guestbook. If the application doesn't guard against potentially malicious input data and allows the script to be stored, the

Join us for the largest gathering of the PHP Community!

Zend/PHP Conference 2010

The Premier PHP Conference
November 1-4, 2010 Santa Clara, CA

zend php 2010

Conference Highlights

- Sessions focused on PHP best practices for architecture, design and development
- Sessions designed for all knowledge levels
- Intensive tutorials for accelerated learning
- UnConference sessions for attendees who share an interest in other PHP topics
- PHP Certification crash courses and testing
- Exhibit hall showcasing the latest products
- Special networking opportunities during meals and events

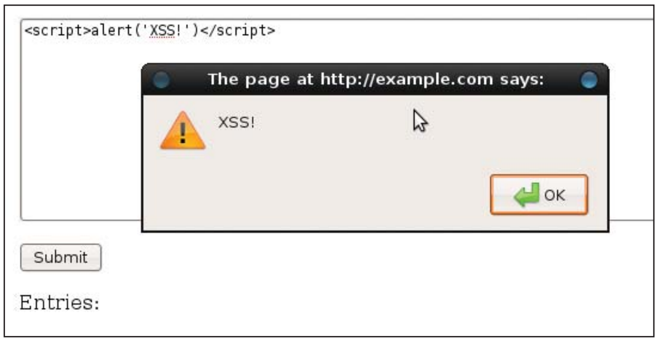
More information and registration:
www.zendcon.com

This powerhouse tool is capable of scanning for and exploiting almost any vulnerability you care to name.

browser of the next person to load the page will execute the malicious code.

By contrast, a reflected or nonpersistent cross-site scripting attack requires the victim to click on a link from another source. Here, instead of being stored in the Web application's database, the malicious script is encoded in the URL. XSS exploits can allow attackers to deface a Web site, redirect legitimate traffic to their own site or a site of their choice, and even steal session data through cookies.

This first test utilizes a formidable tool that enjoys wide popularity among attackers—a Web browser. Although automated tools can take the pain out of iterating over a long list of tests, the simplest method to start testing your site for XSS vulnerabilities is by using a browser. You'll need to find a page on the application you'd like to test that accepts user input and then displays it—something like a guestbook, comment field or even a search dialog that displays the search string as part of the results.



```
<script>alert('XSS!')</script>
```

The page at http://example.com says:

XSS!

Figure 4. XSS Vulnerability Executed

The canonical check for an XSS vulnerability is considered to be the snippet of code below:

```
<script>alert('xss!')</script>
```

If the Web site isn't properly filtering input, this script will execute in the browser and then display a pop-up message with the text "xss". There's some bad news for my little Web application here. It's so vulnerable that this simplest of strings was able to exploit it (Figure 4).

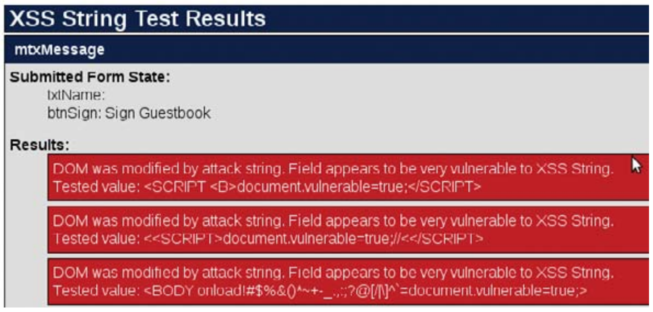
"brilliant...free software, hacktivism, and social practices acting together and demonstrating that IT experts, users, and amateurs can find common ground in discussions" Linux Magazine

- 65 Speakers in 56 talks
- Premier event for Free Culture and Free Software
- Exceptional university venue
- Unique platform for brainstorming & discussion
- Workshops hosted by leading projects & organisations
- Social events and food
- Presentations on Git, 1-Wire networks, RepRap, Qt, YaCy, Appleseed & Makerbot

THE WORLD'S PREMIER CONFERENCE FOR SOCIAL CHANGE IN TECHNOLOGY

FSCONS 2010
Free Society Conference and Nordic Summit

Gothenburg
Sweden
5-7 Nov
Standard entry €30



XSS String Test Results

mtxMessage

Submitted Form State:

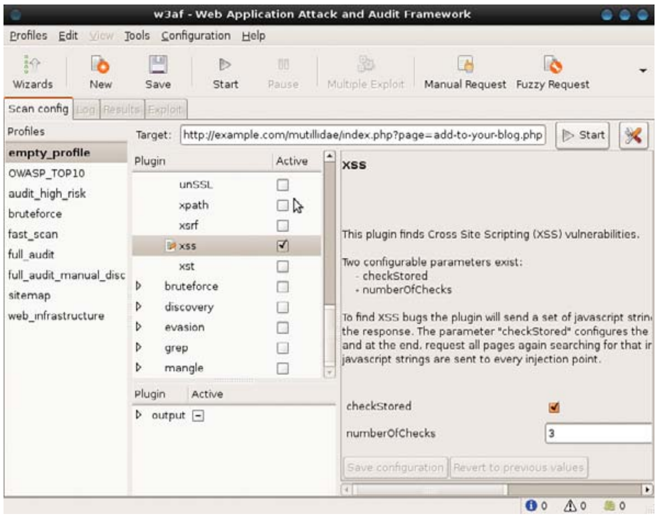
txtName:

btnSign: Sign Guestbook

Results:

```
DOM was modified by attack string. Field appears to be very vulnerable to XSS String.
Tested value: <SCRIPT <B>document.vulnerable=true;</SCRIPT>

DOM was modified by attack string. Field appears to be very vulnerable to XSS String.
Tested value: <<SCRIPT>document.vulnerable=true;//<</SCRIPT>

DOM was modified by attack string. Field appears to be very vulnerable to XSS String.
Tested value: <BODY onload!#$%&()*~+-_.,:;?@[/|\]^`=document.vulnerable=true;>
```

Figure 5. XSS Me Test Results


w3af - Web Application Attack and Audit Framework
Profiles Edit View Tools Configuration Help
Wizards New Save Start Pause Multiple Exploit Manual Request Fuzzy Request
Scan config Log Results Exploit
Start
Save configuration
Revert to previous values

Profiles: empty_profile, OWASP_TOP10, audit_high_risk, bruteforce, fast_scan, full_audit, full_audit_manual_disc, sitemap, web_infrastructure

Target: http://example.com/mutillidae/index.php?page=add-to-your-blog.php

Plugin	Active
unSSL	☐
xpath	☐
xsrf	☐
xss	✓
xst	☐
bruteforce	☐
discovery	☐
evasion	☐
grep	☐
mangle	☐

Plugin	Active
output	☐

xss

This plugin finds Cross Site Scripting (XSS) vulnerabilities.

Two configurable parameters exist:
- checkStored
- numberOfChecks

To find XSS bugs the plugin will send a set of javascript strin
the response. The parameter "checkStored" configures the
and at the end, request all pages again searching for that ir
javascript strings are sent to every injection point.

checkStored ✓

numberOfChecks 3

Figure 6. w3af Configuration

Usually, at least some form of input filtering is employed, and this string won't get through intact. By examining the source of the generated page, you can try to diagnose what characters are being filtered to craft more sophisticated strings. If you inject the same string into a slightly more secure Web form, you can see by the result that the script tags are being stripped out and the single quotes are escaped:

```
Name: Me!
Message: alert(\'XSS!\')
```

You could continue encoding different parts of the string and observing the results to try to get past the filter. Or, to speed up the process, you could use an automated tool. Samurai includes the XSS Me plugin for Firefox by Security Compass. Navigate to the page you'd like to test, then select XSS Me→Open XSS Me Sidebar from the Tools menu. Select Test all forms with top attacks from the sidebar. For the curious, the strings that the XSS Me uses can be viewed in Tools→XSS Me→Options→XSS Strings, and more strings can be added.

Figure 5 shows that XSS Me has discovered that this application is vulnerable to DOM-based XSS attacks. This is a particularly

ASA COMPUTERS

Want your business to be productive?

The ASA servers powered by the Intel® Xeon® Processor provide the quality and dependability to keep up with your growing business.

" Since 1989 Integration and service with pride "

ASA 1U Series

Intel® Atom™ processors D510 and D410 based platform and are designed for embedded industrial PC (IPC) applications. These quiet, energy saving solutions make ideal network appliances, print and email servers.

ASA 2U Series

QPI, Intel® Xeon® Processor 5500 Series in a high-density 2U form-factor are ideal for network infrastructure, front-end enterprise, and minimal downtime cluster server systems maximum upto 4 nodes in 2U.

ASA 3U Series

Server series excel under ISCSI/NAS/JBOD environments, the 3U servers support high availability storage and mission critical business applications.

ASA 4U Series

Optimized for enterprise-level high-capacity storage applications, features 36x (24 front + 12 rear) 3.5" Hot-swap HDD bays, reliable and hassle-free maintenance storage system.

ASA Blade Series

Blade server with, Highest computing density (20 DP nodes and 2.56TB mempry in 7U) Fastest and Most Cost-Effective Networking Solution (Infiniband DDR/QDR support)

E-mail - sales@asacomputers.com
Call - 1800-REAL-PCS

ASA Computers, Inc
645 National Ave,
Mountain View. CA 94043
www.asacomputers.com

intel Xeon inside

Powerful. Intelligent.

Intel, the Intel logo, Xeon, and Xeon Inside are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

insidious way of exploiting an application, as it doesn't rely on the application embedding the malicious code in the HTML output.

Another tool that can scan for XSS vulnerabilities in Samurai is, again, w3af. This powerhouse tool is capable of scanning for and exploiting almost any vulnerability you care to name. Here, let's configure it to scan the Web application for XSS vulnerabilities.

Launch the w3af GUI from the Applications→Samurai→Discovery menu. Enter the Web site URL you'd like to test in the Target: field, and then expand the options under discovery in the plugin box. Scroll down until you find the webSpider plugin and check its box to enable it. In the pane to the right, the options for the webSpider plugin will be shown. Tick onlyForward and select Save Configuration. Now, scroll back to the top of the plugins box, and expand Audit. Scroll down to xss, and check its box to enable it (Figure 6).

Click Start to start the audit process. Depending on the size of the Web application, this could take a few minutes as the spider plugin discovers every URL, and then the audit plugin tests all applicable forms for XSS vulnerabilities. Once complete, navigate to the Results tab, which will itemize the vulnerabilities found.

Conclusion

Although useful, penetration testing is only part of the picture. To truly address these risks, applications must be designed, implemented and deployed with security in mind. Code analysis tools are helpful in locating places where application code deals with user inputs, so the code can be audited for input validation, strong output encoding and safe quote handling. Careful deployment using the principle of least privilege and employing chroot jails can help minimize the damage attackers can do if they gain access to your application. Never allow your database or Web server process to run as the root user.

Hopefully, this article can serve as a jumping off point to help you approach Web application security from a more active point of view, but there are many exploits and aspects of these two specific exploits that aren't covered here. For further reading, see the OWASP Wiki at **www.owasp.org**.

Acknowledgement

Special thanks to Adrian Crenshaw from irongeek.com, the author of our example application, Mutillidae. Mutillidae is a Web application designed to be deliberately vulnerable to the OWASP top ten in an easy-to-understand form for education purposes. Check out his site for some excellent resources on Web application security.■

Jes Fraser is an IT Consultant from Open Systems Specialists in New Zealand. She's passionate about promoting open source and Linux in the enterprise.

CANCELLED

ONTARIO GNU/LINUX FEST

TORONTO, ONTARIO, CANADA

Get all of the details at http://oglf.ca

info@oglf.ca

Sadly, we will not be holding Ontario GNU/Linux Fest 2010.

Thank you to all of the attendees, speakers, and volunteers who have made it such a great event.

Please visit oglf.ca for more information.

Polywell Industrial Mini-PCs

Cost Effective Embedded PC For Appliances

ITX-30G with NVIDIA® ION™ Graphics Barebone system **$199**

Full Height Riser or Low-Profile Add-on Slot up to 8 x 2.5" or 4 x3.5" Hard Drives

ITX-1000C with 4LAN and WiFi Option

Over 250 Mini-ITX Models Available:
- NVIDAI® GeForce 8200/8100 with AMD® Athlon/Phenom Processor
- NVIDIA® GeForce 9300/9100M/7050 with Intel® Core 2 Duo Processor
- PCI, PCIe, MiniPCIe Slot for TV Tuner or Industrial Add-on
- Custom Design Chassis for Small to Mid Size OEM Project

888.765.9686
linuxsales@polywell.com
polywell.com/us/Lx

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

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

NVIDIA, ION, nForce, GeForce and combinations thereof are trademarks of NVIDIA Corporation. Other names are for informational purposes only and may be trademarks of their respective owners.

POLY POLYWELL

Polywell Storage Servers

More Choices, Excellent Service, Great Prices!

Netdisk 9015N

Quiet Storage NAS/SAN/iSCSI

10TB $1,799	- Dual Gigabit LAN - RAID-5, 6, 0, 1, 10
20TB $2,999	- Hot Swap, Hot Spare - Linux, Windows, Mac
30TB $4,999	- E-mail Notification - Tower or Rackmount

Silent Eco Green PC
The Best Terminal PC
Intel® Low Voltage Processor
Energy efficient, Quiet Platform.
starts at $199

Mini-ITX LD-001

High Performance 5048A

5U-48Bay 96TB Storage Server

4U45JB / 4U36A / 4U24A
RAID-6, NAS/iSCSI/SAN Storage
Mix SAS / SATA, 4 Giga / 10Gbit LAN

4U-24Bay 48TB $8,999
4U-36Bay 72TB $12,500
4U-45Bay 90TB JBOD

2008A / 2012A / 2012B
RAID-6, NAS/iSCSI/SAN Storage
Mix SAS / SATA, 4 GigaLAN

2008A 16TB $2,699
2012A 24TB $3,999
2012B 24TB $4,899

1U945GCL2

Mini-1U Server $499
Intel® Dual-Core Processor, 2 x 500G RAID
Dual GigaLAN, 4GB DDR2 RAM

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

888.765.9686
linuxsales@polywell.com

www.polywell.com/us/Storage

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

Intel is trademarks of Intel Corporation. Other names are for informational purposes only and may be trademarks of their respective owners.

POLY POLYWELL

Coding in Pixels

Even if you're not a programmer, you still can make yourself useful.

DOC SEARLS

My parents were old-school. Literally. My mother (born 1913) started teaching at age 19 in a one-room schoolhouse in rural North Dakota, where she grew up. My father (born 1908) learned carpentry from his father (born 1863) and taught me as well. I also absorbed a value both my parents shared. "Make yourself useful", they said, as often as they saw my sister or me failing to do that.

Respect for usefulness gave me a special appreciation for Linux, as well as for free and open-source code. Usefulness is why most free and open-source code gets written in the first place. It's also what makes that code valuable. "The use value of a program is its economic value as a tool, a productivity multiplier", Eric Raymond writes (**catb.org/esr/writings/homesteading/magic-cauldron/ar01s03.htm**). Elsewhere (**catb.org/esr/writings/homesteading/cathedral-bazaar/ar01s02.html**), he adds, "Every good work of software starts by scratching a developer's personal itch." And, "Good programmers know what to write. Great ones know what to rewrite (and reuse)."

I've never made myself useful as a programmer. As fate has it, the only code I know is Morse. (Well, there's HTML, which I don't think counts much more than Morse does.) My skills with electronics, such as they are, were developed in my days as a ham radio operator, in the earliest 1960s. I was a kid then, building radio things and cultivating a lifelong obsession with devices that transmit, receive or both. For a while in my early adult life, that obsession got me work manning radio station transmitters and doing site studies for new FM stations. But I went on to other careers, and almost none of what I knew about broadcasting many decades ago is of practical use now, even in what's left of the broadcast field.

The other practical science I learned while young was darkroom chemistry.

Salt Ponds in San Francisco Bay

That came out of my work as a newspaper photographer, around the turn of the 1970s. Today, none of my old darkroom skills (pushing, burning, dodging and so on) have much leverage. My skills as a photographer, however, have only improved since I first picked up a digital camera about six years ago. At first, I mostly shot candid photos of people, since that was my specialty back when I did newspaper work. Then, after Flickr came along, I found I could do something much more useful. I could put up pictures that others could reuse.

That wasn't the main idea behind Flickr, but it was the idea behind Flickr's base code infrastructure, which was built from the ground up on Linux. It was also the idea behind giving users a choice of Creative Commons licenses for each of their pictures. I was familiar with Creative Commons before I started posting photos on Flickr, but I hadn't had personal experience with Creative Commons' effects. Flickr gave me that, in spades. What I discovered was that many of my pictures were proving useful, mostly as visuals in Wikipedia articles. That mattered far more to me than popularity or remuneration.

As of today (mid-August 2010), I have more than 36,000 photos on Flickr. All carry Creative Commons licenses that permit reuse and remixing. Most of them are captioned or tagged, making them easy to find when searching for one of their subjects. In last June's EOF, I told the story of how some of my winter shots ended up serving as wallpaper for NBC's 2010 Winter Olympics coverage. That was cool, but far cooler is seeing 153 of my photos show up in Wikimedia Commons (**commons.wikimedia.org/w/index.php?title=Special:Search&search=doc+searls**), through no additional effort of my own. Some shots are of people (including Eric. S. Raymond, Nat Friedman and Guido van Rossum). But most shots are of places I've seen out the windows of airplanes. These include a town in Nebraska, a lake in Norway, an island in Scotland's Outer Hebrides, a lava field in the Grand Canyon, salt ponds in San Francisco Bay and a glacier in Greenland. Nearly all of them illustrate at least one Wikipedia article. Some illustrate many. For example, a shot of the spiky white fabric roof of Denver International Airport shows up in 17 different articles across 13 language versions.

I think of each photo as a potential code patch to our body of common knowledge, not as a work of art. If they make themselves useful, then I've done the same.■

Doc Searls is Senior Editor of *Linux Journal*. He is also a fellow with the Berkman Center for Internet and Society at Harvard University and the Center for Information Technology and Society at UC Santa Barbara.

iX-Triton TwinBlade Servers: The Easy-to-Manage, Greener Way to Serve

› AFFORDABLE › ECONOMICAL › SAVINGS

The new **Triton TwinBlade Server** is the most technologically advanced blade server system in the industry, and the ideal solution for power-efficiency, density, and ease of management.

The Triton TwinBlade Server supports up to 120 DP servers with 240 Intel® Xeon® 5600/5500 series processors per 42U rack, achieving an unmatched 0.35U per DP node. Up to two 4x QDR (40 Gbps) Infiniband switches, 10GbE switches or pass-through modules give the TwinBlade the bandwidth to support the most demanding applications.

With N+1 redundant, high efficiency (94%) 2500W power supplies, the TwinBlade is the Greenest, most energy-efficient blade server in the industry. The energy saved by the iX-Triton TwinBlade Server will keep the environment cleaner and greener, while leaving the green in your bank account.

Server management is also simple with the Triton Twin Blade Server. Remote access is available through SOL (Serial Over Lan), KVM, and KVM over IP technologies. A separate controller processor allows all of the Triton's remote management and monitoring to function regardless of system failures, offering true Lights Out Management.

Using the Triton's management system, administrators can remotely control TwinBlades, power supplies, cooling fans, and networking switches. Users may control the power remotely to reboot and reset the Triton TwinBlade Center and individual Twin Blades, and may also monitor temperatures, power status, fan speeds, and voltage.

For more information on the **iX-Triton TwinBlade**, or to request a quote, visit: http://www.iXsystems.com/tritontwinblade

Key features:

- Up to 10 dual-node TwinBlades in a 7U Chassis, 6 Chassis per 42U rack
- Remotely manage and monitor TwinBlades, power supplies, cooling fans, and networking switches
- Virtual Media Over Lan (Virtual USB, Floppy/CD, and Drive Redirection)
- Integrated IPMI 2.0 w/ remote KVM over LAN/IP
- Supports one hot-plug management module providing remote KVM and IPMI 2.0 functionalities
- Up to four N+1 redundant, hot-swap 2500W power supplies
- Up to 16 cooling fans

Each of the TwinBlade's two nodes features:

- Intel® Xeon® processor 5600/5500 series, with QPI up to 6.4 GT/s
- Intel® 5500 Chipset
- Up to 128GB DDR3 1333/ 1066/ 800MHz ECC Registered DIMM / 32GB Unbuffered DIMM
- Intel® 82576 Dual-Port Gigabit Ethernet
- 2 x 2.5" Hot-Plug SATA Drive Trays
- Integrated Matrox G200eW Graphics
- Mellanox ConnectX QDR InfiniBand 40Gbps or 10GbE support (Optional)

iX systems®

intel® Xeon® inside™ Powerful. Intelligent.

Call iXsystems toll free or visit our website today!
+1-800-820-BSDi | www.iXsystems.com

Cool, Fast, Reliable

GPGPU computing for your office and data center

Designed from the ground up for ultimate customer satisfaction, Microway's WhisperStation integrates the latest CPUs with NVIDIA Tesla GPUs. Tesla's massively multi-threaded Fermi architecture, the CUDA™ C and FORTRAN language environments, and OpenCL™ provide the best performance for your application.

- Up to Four Tesla Fermi GPUs per WhisperStation, with 448 cores and 6 GB GDDR5, each delivering 1 TFLOP single and 515 GFLOP double precision performance
- Up to 24 cores with the newest Intel and AMD Processors, 128 GB memory, 80 PLUS® certified power supply, and eight hard drive
- Nvidia GeForce GTX 480 for state of the art graphics
- Ultra-quiet fans, strategically placed baffles, and internal sound-proofing

The Microway Advantage: Custom Integrations and HPC Expertise Since 1982

Put our years of expertise with Linux, Windows, CUDA and OpenCL to work for YOU!

Every Microway system is backed by pre and post sale techs who speak HPC. Whether it's graphics or GPGPU, FORTRAN or MPI, hardware problems or Linux kernel issues; you can talk to Microway's experts to design and support solutions for power hungry applications.

WhisperStation with 4 Tesla Fermi GPUs

1U Node with 2 Tesla Fermi GPUs

2U Twin² Node with 4 Hot-Swap Motherboards Each with 2 CPUs and 256 GB

Microway's Latest Servers for Dense Clustering

- 1U nodes with 48 CPU cores, 512 GB and QDR InfiniBand
- 1U nodes with 24 CPU cores, 2 Tesla GPUs and QDR InfiniBand
- 2U Twin² with 4 Hot-Swap MBs, each with 2 Processors + 256 GB
- 1U S2070 servers with 4 Tesla Fermi GPUs

The Fastest CPUs and GPUs Ever

- 12 Core AMD® Opterons with quad channel DDR3 memory
- 8 Core Intel® Xeons with quad channel DDR3 memory
- 448 Core NVIDIA® Tesla™ Fermi GPUs with 6 GB GDDR5 memory

Configure your next WhisperStation or Cluster today!
www.microway.com/quickquote or call 508-746-7341

NVIDIA TESLA with CUDA | intel Channel Partner Premier 2010 | AMD Fusion Partner Premier | GSA

GSA Schedule Contract Number: GS-35F-0431N

Microway®
Technology you can count on℠