

LINUX JOURNAL

Since 1994: The Original Magazine of the Linux Community

JANUARY 2008 | ISSUE 165

» Getting the Most Out of GCC

» Securing Open-Source Phones

» How to Store Passwords Safely

» New Column: Hack and /

SECURITY

» Learn the Path
to a More
Secure System

» Use Autopsy
and Sleuthkit
to Analyse
Security
Breaches

» Discover
Dangerous
Flaws in Your
DNS Infrastructure



+ Creating IPsec and SSL/TLS
Tunnels in Linux

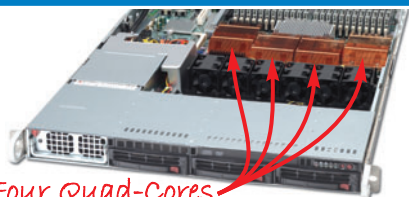
What's New with Eric Raymond?

www.linuxjournal.com



TRUE QUAD-CORE SERVERS. UP TO 16 INDIVIDUAL CORES.

ABERDEEN STONEHAVEN A135



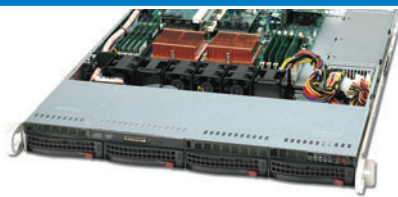
Four Quad-Cores

1U 3TB Quad Quad-Core HPC Server

- Up to four Quad-Core AMD Opteron™ 8000 Series processors
- nVIDIA nForce Pro Chipset with 64-Bit Support
- Up to 64GB 667MHz ECC Registered DDR2 SDRAM
- Up to 3 x 1TB (3TB Total) Hot-Swap SATA Hard Drives
- 1000W AC Power Supply w/PFC
- 5-Year Warranty

Starting at **\$3,789**

ABERDEEN STONEHAVEN A151



1U 4TB Dual Quad-Core Server

- Up to two Quad-Core AMD Opteron 2000 Series processors
- nVIDIA nForce Pro Chipset with 64-Bit Support
- Up to 32GB 667MHz ECC Registered DDR2 SDRAM
- Up to 4 x 1TB (4TB Total) Hot-Swap SATA Hard Drives
- 560W AC Power Supply w/PFC
- 5-Year Warranty

Starting at **\$1,599**

ABERDEEN STONEHAVEN A284



2U 8TB Dual Quad-Core Server

- Up to two Quad-Core AMD Opteron 2000 Series processors
- nVIDIA nForce Pro Chipset with 64-Bit Support
- Up to 64GB 667MHz ECC Registered DDR2 SDRAM
- Up to 8 x 1TB (8TB Total) Hot-Swap SATA Hard Drives
- 700W Redundant Hot-Swap Power Supply
- 5-Year Warranty

Starting at **\$2,059**

ABERDEEN STONEHAVEN X314



3U 12TB Dual Quad-Core Storage Server

- Up to two Quad-Core AMD Opteron 2000 Series processors
- nVIDIA nForce Pro Chipset with 64-Bit Support
- Up to 32GB 667MHz ECC Registered DDR2 SDRAM
- Up to 12 x 1TB (12TB Total) Hot-Swap SATA Hard Drives
- Areca ARC-1231ML PCI Express 800MB/sec RAID Controller
- 650W 2+1 Redundant Hot-Swap Power Supply
- 5-Year Warranty

Starting at **\$3,359**

ABERDEEN STONEHAVEN X418



4U 16TB Dual Quad-Core Storage Server

- Up to two Quad-Core AMD Opteron 2000 Series processors
- nVIDIA nForce Pro Chipset with 64-Bit Support
- Up to 32GB 667MHz ECC Registered DDR2 SDRAM
- Up to 16 x 1TB (16TB Total) Hot-Swap SATA Hard Drives
- Areca ARC-1261ML PCI Express 800MB/sec RAID Controller
- 650W 2+1 Redundant Hot-Swap Power Supply
- 5-Year Warranty

Starting at **\$3,819**

ABERDEEN STONEHAVEN X526



5U 24TB Dual Quad-Core Storage Server

- Up to two Quad-Core AMD Opteron 2000 Series processors
- Up to 32GB 667MHz ECC Registered DDR2 SDRAM
- Up to 24 x 1TB (24TB Total) Hot-Swap SATA Hard Drives
- Up to two Internal SATA Hard Drives for OS
- Areca ARC-1280ML PCI Express 800MB/sec RAID Controller
- 950W 3+1 Triple Redundant Hot-Swap Power Supply
- 5-Year Warranty

Starting at **\$4,959**

ABERDEEN STONEHAVEN X633

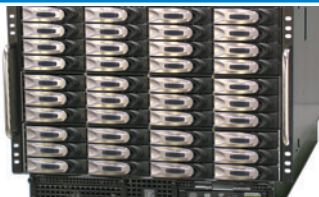


6U 32TB Dual Quad-Core Storage Server

- Up to two Quad-Core AMD Opteron 2000 Series processors
- Up to 32GB 667MHz ECC Registered DDR2 SDRAM
- Up to 32 x 1TB (32TB Total) Hot-Swap SATA Hard Drives
- Up to two Rear Hot Swap SATA Hard Drives for OS
- Dual Areca PCI Express 800MB/sec RAID Controllers
- 1350W 3+1 Triple Redundant Power Supply
- 5-Year Warranty

Starting at **\$6,399**

ABERDEEN STONEHAVEN X840



8U 40TB Dual Quad-Core Storage Server

- Up to two Quad-Core AMD Opteron 2000 Series processors
- Up to 32GB 667MHz ECC Registered DDR2 SDRAM
- Up to 40 x 1TB (40TB Total) Hot-Swap SATA Hard Drives
- Up to two Rear Hot Swap SATA Hard Drives for OS
- Dual Areca PCI Express 800MB/sec RAID Controllers
- 1350W 3+1 Triple Redundant Power Supply
- 5-Year Warranty

Starting at **\$7,999**

Enhanced AMD PowerNow!™ technology delivers optimum power and performance for each core depending on core workloads, without compromising performance.



Manage Any Data Center. Anytime. Anywhere.



SEE US AT ANY OF THESE SHOWS!

Infrastructure Mgt. World
Scottsdale, AZ - Booth TBD
Sept. 10 - 12

VM World
San Francisco, CA - Booth 1113
Sept. 11 - 13

AFCOM Data Center World
Dallas, TX - Booth 436
Sept. 17 - 18

High Perf. on Wall Street
New York, NY - Booth 216
Sept. 17

IDC Enterprise Infra. Forum
New York, NY - Booth TBD
Sept. 20

Interface Salt Lake City
Salt Lake City, UT - Booth 309
Oct. 4

GEOINT
San Antonio, TX - Booth 374
Oct. 22 - 24

Interop New York Fall
New York, NY - Booth 543
Oct. 24 - 25

AFCEA Asia-PAC TechNet
Honolulu, HI - Booth 516
Nov. 4 - 9

Super Computing
Reno, NV - Booth 164
Nov. 12 - 15

LISA
Dallas, TX - Booth 200
Nov. 14 - 15

DaCEV Awards
Atlanta, GA
Nov. 15

Gartner Data Center Conf.
Las Vegas, NV - Booth TBD
Nov. 27 - 30

Interface Seattle
Seattle, WA - Booth 206
Nov. 28

Avocent builds hardware and software to access, manage and control any IT asset in your data center, online or offline, keeping it, and your business, "always on".



Visit us on our Remote Control Tour. For locations near you, go to www.avocent.com/remotecontrol.


Avocent[®]
The Power of Being There[®]

Avocent, the Avocent logo and The Power of Being There, are registered trademarks of Avocent Corporation. ©2007 Avocent Corporation.

CONTENTS

JANUARY 2008

Issue 165



FEATURES

40 THE TAO OF LINUX SECURITY: FIVE LESSONS FOR A SECURE DEPLOYMENT

Tighten up your systems from the start using this simple plan.

Jeremiah Bowling

48 DIGGING UP DIRT IN THE DNS HIERARCHY, PART I

Even when your DNS system is functioning normally, all may not be well below the surface.

Ron Aitchison

54 INTRODUCTION TO FORENSICS

Hit the ground running on your first forensics project with Autopsy and Sleuthkit.

Kyle Rankin

60 PACKETFENCE REVISITED

PacketFence's extensive isolation mechanisms secure both your wired and wireless networks.

Regis Balzard and Dominik Gehl

ON THE COVER

- Getting the Most Out of GCC, p. 70
- Securing Open-Source Phones, p. 80
- How to Store Passwords Safely, p. 76
- New Column: Hack and /, p. 34
- Learn the Path to a More Secure System, p. 40
- Use Autopsy and Sleuthkit to Analyse Security Breaches, p. 54
- Discover Dangerous Flaws in Your DNS Infrastructure, p. 48
- Creating IPsec and SSL/TLS Tunnels in Linux, p. 90
- What's New with Eric Raymond?, p. 66

EmperorLinux

...where Linux & laptops converge



Portable

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

Toucan T61/T61ws

ThinkPad T61/T61ws by Lenovo

- Up to 15.4" WUXGA w/ X@1920x1200
- NVidia Quadro FX 570M graphics
- 1.8–2.4 GHz Core 2 Duo
- 512 MB–4 GB RAM
- 80–160 GB hard drive
- CDRW/DVD or DVD±RW
- 5.2–6.0 pounds
- 10/100/1000 Mbps ethernet
- 802.11a/b/g (54Mbps) WiFi
- **Starts at \$1530**



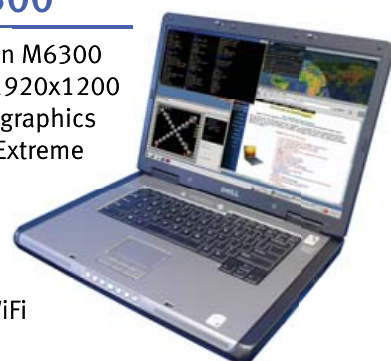
Powerful

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

Rhino D830/M6300

Dell Latitude D830/Precision M6300

- Up to 17" WUXGA w/ X@1920x1200
- NVidia Quadro FX 1600M graphics
- 1.8–2.8 GHz Core 2 Duo/Extreme
- 512 MB–4 GB RAM
- 60–200 GB hard drive
- DVD±RW or Blu-ray
- 6.3–8.6 pounds
- 802.11a/b/g (54Mbps) WiFi
- ExpressCard/EVDO
- **Starts at \$1360**



Unique

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

Raven X61 Tablet

ThinkPad X61 Tablet by Lenovo

- 12.1" SXGA+ w/ X@1400x1050
- 1.6 GHz Core 2 Duo
- 1–4 GB RAM
- 80–120 GB hard drive
- 3.8 pounds
- Pen/stylus input to screen
- Dynamic screen rotation
- Handwriting recognition
- X61s laptops available
- **Starts at \$2150**



LJ '07 Ultimate
Linux Laptop

www.EmperorLinux.com 1-888-651-6686

CONTENTS

JANUARY 2008

Issue 165

COLUMNS

16 REUVEN M. LERNER'S
AT THE FORGE
Working with Facebook

20 MARCEL GAGNÉ'S
COOKING WITH LINUX
Security's Front Door



24 MICK BAUER'S
PARANOID PENGUIN
Getting a Clue with WebGoat



30 DAVE TAYLOR'S
WORK THE SHELL
Numerology, or the Number 23

34 KYLE RANKIN'S
HACK AND /
Browse the Web without a Trace

96 DOC SEARLS'
EOF
Why to Build on FOSS in the
First Place

INDEPTH

66 INTERVIEW WITH
ERIC RAYMOND
Eric Raymond on open source.
Glyn Moody

70 GCC FOR EMBEDDED
ENGINEERS
A look at how GCC works and how
to get the most out of this marvel of
modern software engineering.
Gene Sally

76 GPG-BASED
PASSWORD WALLET
Forget your passwords.
Carl Welch

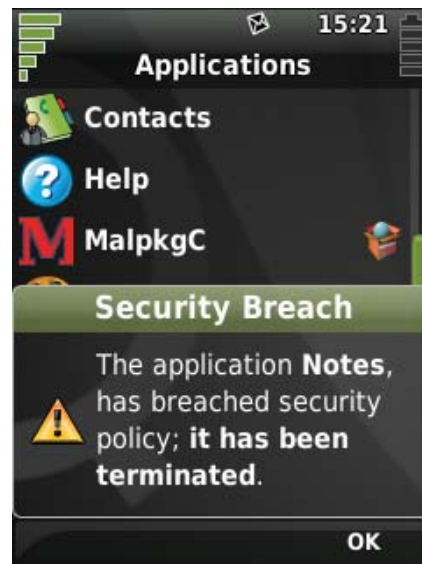
80 SECURITY IN QTOPIA
PHONES
Open source doesn't mean insecure.
Lorn Potter

84 SEPARATE THE STATIC FROM
THE DYNAMIC WITH TOMCAT
AND APACHE
Efficiency tricks with Apache
and Tomcat.
Alan Berg

90 CREATING VPNS WITH
IPSEC AND SSL/TLS
The two most common and current
techniques for creating VPNs.
Rami Rosen

IN EVERY ISSUE

8 LETTERS
12 UPFRONT
36 NEW PRODUCTS
81 ADVERTISERS INDEX



80 SECURITY IN QTOPIA PHONES

Next Month

VIRTUALIZATION

Next month, we'll take a look at using the GPL'd Open Source Edition of innotek's VirtualBox, with a focus on virtualization for home enthusiasts. And for enterprise users, we show you how to migrate a large number of servers to virtual machines with around only 30 minutes of downtime using VMware.

And, as always, there's much more. We review Zonbu, the Earth-friendly Gentoo-based desktop computer with on-line storage and automated backup and updates. We continue looking at techniques for diagnosing and auditing your DNS systems. And, we tackle the "How small can you make it?" question typically asked by embedded engineers.

USPS LINUX JOURNAL (ISSN 1075-3583) is published monthly by Belltown Media, Inc., 2211 Norfolk, Ste 514, Houston, TX 77098 USA. Periodicals postage paid at Houston, Texas and at additional mailing offices. Cover price is \$5.99 US. Subscription rate is \$25/year in the United States, \$32 in Canada and Mexico, \$62 elsewhere. POSTMASTER: Please send address changes to Linux Journal, PO Box 980985, Houston, TX 77098. Subscriptions start with the next issue. Canada Post: Publications Mail Agreement #41549519. Canada Returns to be sent to Bleuchip International, P.O. Box 25542, London, ON N6C 6B2

RouterBOARD 600

The High Performance Wireless Platform

It has four miniPCI slots, three **gigabit** ethernet ports, and it is the fastest wireless board that MikroTik has ever made.

The heart of this device is a new state of the art PowerPC networking processor. It makes the RB600 **faster** than any other MikroTik wireless router, introducing a whole **new** class to the RouterBOARD brand.

Two Compactflash slots for webproxy cache and configuration backups of the User Manager database or The Dude server.

RB600 includes RouterOS - the operating system which makes this the most **sophisticated** wireless router, firewall, bandwidth manager, or hotspot.

And all this **power** at a very affordable price:

\$245



routerboard.com

CPU	MPC8343E 266/400MHz network processor
Memory	64MB DDR SDRAM onboard memory
Boot loader	RouterBOOT, 1Mbit Flash chip
Data storage	64MB onboard NAND memory chip
Ethernet	Three 10/100/1000 Mbit/s Gigabit Ethernet with Auto-MDI/X
miniPCI	Four MiniPCI Type IIIA/IIIB slots
Expansion	Daughterboard support, including RB500 daughterboards
Compact Flash	Two independent CF slots (incl.TrueIDE Microdrive)
Serial port	One DB9 RS232C asynchronous serial port
Speaker	Mini PC-Speaker
Power options	IEEE802.3af PoE: 38..56V DC including over datalines. Power jack: 10..56V DC
Fan control	Two 5V DC fan power output headers with rotation sensor and automatic fan switching (maximum output current - 300mA total)
Dimensions	14 cm x 20 cm (5.51 in x 7.87 in), 227 g (8 oz)
Power consumption	~9W without extension cards, maximum - 35+ W
Operating System	MikroTik RouterOS v3, Level4 license

GO SOLID.
INCREASE RELIABILITY.



solid state systems
fully x86 compatible
fanless, quiet operation



Direct-Plug
IDE Flash Modules

Intel, VIA & AMD CPUs

95% Efficiency-Rated
PicoPSU Power Supplies

DISCOVER MINI-ITX.

LOGIC
SUPPLY

www.logicsupply.com

LINUX JOURNAL

Executive Editor Jill Franklin
jill@linuxjournal.com

Senior Editor Doc Searls
doc@linuxjournal.com

Art Director Garrick Antikajian
garrick@linuxjournal.com

Products Editor James Gray
newproducts@linuxjournal.com

Editor Emeritus Don Marti
dmarti@linuxjournal.com

Technical Editor Michael Baxter
mab@cruzio.com

Senior Columnist Reuven Lerner
reuven@lerner.co.il

Chef Français Marcel Gagné
mggagne@salmar.com

Security Editor Mick Bauer
mick@visi.com

Contributing Editors

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

Proofreader Geri Gale

Publisher Carlie Fairchild
publisher@linuxjournal.com

General Manager Rebecca Cassity
rebecca@linuxjournal.com

Director of Sales Laura Whiteman
laura@linuxjournal.com

Regional Sales Manager Joseph Krack
joseph@linuxjournal.com

Regional Sales Manager Kathleen Boyle
kathleen@linuxjournal.com

Circulation Director Mark Irgang
mark@linuxjournal.com

System Administrator Mitch Frazier
sysadm@linuxjournal.com

Webmaster Katherine Druckman
webmaster@linuxjournal.com

Accountant Candy Beauchamp
acct@linuxjournal.com

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

Reader Advisory Panel

Brad Abram Baillio • Nick Baronian • Hari Boukis • Caleb S. Cullen • Steve Case
Kalyana Krishna Chadalavada • Keir Davis • Adam M. Dutko • Michael Eager • Nick Faltys • Ken Firestone
Dennis Franklin Frey • Victor Gregorio • Kristian Erik • Hermansen • Philip Jacob • Jay Kruiuzenga
David A. Lane • Steve Marquez • Dave McAllister • Craig Oda • Rob Orsini • Jeffrey D. Parent
Wayne D. Povel • Shawn Powers • Mike Roberts • Draciron Smith • Chris D. Stark • Patrick Swartz

Editorial Advisory Board

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

Advertising

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

Subscriptions

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

LINUX is a registered trademark of Linus Torvalds.

4x4

to drive your business

AMD® Quad-Core Opteron™ servers from Polywell

www.Polywell.com/LJ

1U Multi-purpose Servers

Advanced 1U Servers: starting at \$3,299
Poly 2500A16 - 2x Quad Core Opteron 8347
16GB DDR2, 4x250GB HD, Dual Gigabit LAN
Entry Level 1U starts at \$499



Polywell has been specializing in building customized computer solutions for over 20 years.

Our system specialists will work with you to get you the best solution for any project.

Polywell has a tech team fully dedicated to supporting our Linux users. We also offer a wide selection of Linux distrOS.

All Opteron servers will support both Dual-Core and Quad-Core.

1U to 8U up to 32 way, 128G RAM

2U 16Way Server:
Poly 8425SS - 4x Opteron 8347, 64GB DDR2
8x500GB (RAID 5 Storage) 3xGigabit LAN



Storage up to 24TB

12TB Storage driven by Opterons starting at \$7,299
Other Options: 24TB Storage
SAN/NAS, SUMA Storage also available



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

Blades 10 Dual or Quad Processors

PolyBlade 2500A:
10x (Dual Opteron 2210, 4GB RAM, 80G HD)
Blades servers need not be fully populated.



Polywell OEM Services, Your Virtual Manufacturer

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

888.765.9686

- 20 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

Opteron and ATHLON are trademarks of Advanced Micro Devices, Inc. Image of truck is a Ford Expedition, trademark of Ford Motors Corporation. All other brands, names are trademarks of their respective companies.



letters



Date Tip

In the Tech Tips section of the November 2007 *Linux Journal*, there is a tip called “Show Date or Time, Past or Future” that requires you to download and compile showdate.c.

Another way to do these date calculations is to use the GNU date command that is included in most (all?) Linux distributions. See:

www.gnu.org/software/coreutils/manual/coreutils.html#Relative-items-in-date-strings.

--
David Blackman

Stealing

Mr Copeland’s argument for stealing music boiled down to its essence is “because it’s possible, it’s okay” [*LJ*, November 2007, Letters]. It’s a refreshingly candid argument.

While I certainly agree that by buying a song, I should be allowed and able to play it on any device I own (or will own), it does not follow that I now have the right to redistribute it to others, simply because I could!

The (possible) greed of the RIAA, et al., is a red herring. If you think an item is too expensive, don’t buy it. If enough people feel the same way, the price will come down (or the creator will go bankrupt).

If in fact the community (including *Linux Journal*) does indeed condone Mr Copeland’s (and other’s) argument, then I have a perfectly good CD containing the latest *Linux Journal* archive (which I recently bought) that I will be happy to make available on my Web site for immediate free worldwide distribution. I’ll just conveniently ignore that bothersome *LJ* “All Rights Reserved” copyright notice at the bottom.

The irony is apparent in Mr Copeland’s last paragraph where he states “If you fear people or begrudge them control over their own lives....”

Musicians, artists and others who create intellectual property or content are people too, and by your own arguments, they have the right to control their own lives, and that means THEY have the right to decide what happens to their intellectual property, how it should be distributed, what rights are allowed or reserved, etc. Why shouldn’t they be entitled to leverage new technology to make money for themselves? Ultimately, that’s what we all do.

--
David Jameson

Disabling Services Revisited

Nicholas Petreley’s command-line method to temporarily disable services, as described in the July 2007 *Linux Journal* Tech Tips section, works.

Granted, renaming the files the way he suggests is a far better alternative to simply removing them. However, there are very easy-to-use command-line utilities that properly deal with this problem:

- chkconfig (Red Hat or alike systems).
- update-rc.d (Debian or alike systems—consider looking into file-rc).

One more reason not to rename these files is that they will most likely be missed by the package manager when the package is removed.

--
Martijn

Waiting for Feedback

I read, with not inconsiderable interest, parts of the letter by Chuck Adams titled “Sold on Kubuntu”, October 2007 *LJ*, wherein appeared, “Now I can do a case in less than ten minutes on an AMD64.”

Well, now, that simple assertion—per se—seemed so interesting to me, for I did my M. Sc. N-Body work on an IBM System/360 Model 50 under OS/360 circa 1973, I just had to check into it a little more deeply! And, you see, my most expensive single run cost only ~ C\$7,000.00, I seem to recall!

So, assuming his runs took 3.5 hours (that is, 210 minutes) on a CDC 7600 versus ten minutes on an AMD64, the time ratio is only 21—I calculated!

Therefore, while I guess that’s possible—albeit somewhat seemingly [s]low—any feedback therefore from Dr Adams would be most gratefully appreciated, of course!

I now mostly employ Red Hat Linux Fedora 7, x86_64, on an AMD Athlon 64 X2 Dual-Core 5600+ processor-based PC built from components purchased from Fry’s and assembled at home, *avec quelque succes*.

--
Joseph Roy D. North

The Audacity...

After a lapse, it’s good to be receiving *Linux Journal* again. As a nontraditional (older) college student, money can be very tight at times. Money has been very tight over the past couple of years. Unfortunately, pleasures in life end up getting pushed aside in the name of the necessities.

I have always liked the Cooking with Linux column by Marcel Gagné. In the October

LJ pays \$100 for tech tips we publish. Send your tip and contact information to techtips@linuxjournal.com.

KEEP YOUR BUSINESS RUNNING SMOOTHLY

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

SDR-1105T 1U ENTRY LEVEL SERVER



\$1,129

EXCELLENT GENERAL PURPOSE SERVER FOR ORGANIZATIONS WITH THE NEED FOR A LOW, ENTRY LEVEL PRICE

- * 1U Rackmount Chassis with 520W power supply
- * Supermicro X7DVL-L Server Board with Intel® 5000V (Blackford VS) Chipset
- * Intel Quad-Core Xeon Processor E5310 1.6GHZ
- * Total 512MB, 2pcs x 256MB Kingston DDR2 533Mhz FB-DIMM ECC
- * Seagate SATAII 160GB 7200 RPM 8MB Cache SATA 3.0Gb/s Hard Drive
- * 4 x 1" Hot-swap SATA Drive Bays
- * Two Intel® 82563EB Dual-port Gigabit Ethernet Controller
- * Intel® ESB2 SATA 3.0Gbps Controller RAID 0, 1, 5, 10 support



STARTING PRICE \$1,259

SDR-2503T 2U APPLICATION SERVER

Highest performing with Dual Core/ Quad Core Xeon CPU based. Excellent with general purpose applications and provide the most power.

- * 2U Rackmount Chassis with 650W power supply
- * Supermicro X7DVL-E Server Board with Intel® 5000V (Blackford VS) Chipset
- * Intel Quad-Core Xeon Processor E5310 1.6GHZ
- * Total 512MB, 2pcs x 256MB Kingston DDR2 533Mhz FB-DIMM ECC
- * Seagate SATAII 250GB 7200 RPM 16MB Cache SATA 3.0Gb/s Hard Drive
- * 6 x 1" Hot-swap SATA Drive Bays
- * Intel® (ESB2/Gilgal) 82563EB Dual-port Gigabit Ethernet Controller
- * Intel® ESB2 SATA 3.0Gbps Controller RAID 0, 1, 5, 10 support



STARTING PRICE \$1,999

SDR-3500T 3U DATABASE SERVER

Easily Scalable storage solution with hot-swap functionality for growing businesses

- * 3U Rackmount chassis with Redundant 800W power supply
- * Supermicro X7DBE+ Server Board with Intel® 5000P (Blackford) Chipset
- * Intel Quad-Core Xeon Processor E5310 1.6GHZ
- * Total 1024MB, 2pcs x 512MB Kingston DDR2 533MHz FB-DIMM ECC
- * Seagate SATAII 500GB 7200 RPM 16MB Cache SATA 3.0Gb/s Hard Drive
- * 16 x 1" Hot-swap SATA Drive Bays
- * Dual-port Gigabit Ethernet Controller
- * Intel ESB2 SATA 3.0Gbps Controller RAID 0, 1, 10 support



STARTING PRICE \$1,199

SDR-7045B-TB 4U FILE SERVER

4U Quad-Core Xeon Server offers excellent value and expandability

- * 4U Rackmountable / Tower with 650W power supply
- * Supermicro Super X7DBE Server Board with Intel® 5000P (Blackford) Chipset
- * Intel Quad-Core Xeon Processor E5310 1.6GHZ
- * Total 1024MB, 2pcs x 512MB Kingston DDR2 667MHz FB-DIMM ECC
- * Seagate SATAII 750GB 7200 RPM 16MB Cache SATA 3.0Gb/s Hard Drive
- * 6 x 3.5" Hot-swap SAS/SATA Drive Bays
- * Dual-port Gigabit Ethernet Controller
- * Intel ESB2 SATA 3.0Gbps Controller RAID 0, 1, 5, 10 support



STARTING PRICE \$3,099

SDR-5111T 5U ADVANCED STORAGE SERVER

Quad Core dual Xeon CPU based, with 24 hot-swap hard disk bays suitable for 18TB of pure data Storage capacity

- * 5U Rackmount chassis with Redundant 1350W power supply
- * Supermicro X7DBE Server Board with Intel® 5000P (Blackford) Chipset
- * Intel Quad-Core Xeon Processor E5310 1.6GHZ
- * Total 1024MB, 2pcs x 512MB Kingston DDR2 667MHz FB-DIMM ECC
- * Seagate 750GB 7200 RPM 16MB Cache SATA 3.0Gb/s Hard Drive
- * 24 x 1" Hot-swap Drive Bays
- * Intel® (ESB2/Gilgal) 82563EB Dual-port Gigabit Ethernet Controller
- * Intel ESB2 SATA 3.0Gbps Controller RAID 0, 1, 5, 10 support

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

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

1.877.727.7887 | www.ServersDirect.com

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



TS-7800 High-End Performance with Embedded Ruggedness



\$269 qty 1 **\$229** qty 100

500 MHz ARM9

- New unbrickable design- 3x faster
- Backward compatible w/ TS-72xx
- Low power - 4W at 5V
- 128MB DDR RAM
- 512MB high-speed onboard Flash
- 12K LUT user-programmable FPGA
- Internal PCI Bus, PC/104 connector
- 2 USB 2.0 480 Mbps
- Gigabit ethernet
- 2 SD sockets
- 10 serial ports
- 110 GPIO
- 5 10-bit ADC
- 2 SATA ports
- Sleep mode uses 200 microamps
- Boots Linux in < 2 seconds
- Linux 2.6 and Debian by default

Design your solution with one of our engineers

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

See our website for options, peripherals and x86 SBCs

[LETTERS]

2007 issue, he wrote about audio editors. As an Audacity user, I can say Mr Gagné forgot one tiny little detail when it comes to writing out one's creations, and that is, Audacity does not come ready to write to the MP3 format. (At least this is true in SUSE.) It will write to the OGG and wav format right out of the box but not to the MP3 format. I always have had to, first, install the MP3 library file in the proper place and then tell Audacity where to find it before I could write MP3 files.

--

Walt L. Williams

Thanks Jack

I just wanted to say that I enjoyed Jack Xue's article on creating a Linux-based e-mail system that integrates with Active Directory in the November 2007 issue. One thing I want to correct or add to his article is that Windows Server 2003 R2 is not required for acquiring the POSIX schema modifications that are needed for authenticating UNIX clients. POSIX attributes and object classes are required, and Windows Server 2003 R2 includes those schema changes automatically; however, R1's Active Directory schema can still be modified to support POSIX accounts. It's just that the same third-party product that provides it to R2 has to be manually installed for R1: Services For Unix 3.5 (SFU). This includes the POSIX schema changes as well as other UNIX utilities and services (for example, NFS). This is useful for people who have Windows Server 2003 R1 who do not want to upgrade to R2.

I hope Jack has more articles to share with us in the future. Great article.

--

Brandon McCombs

Did you know Linux Journal maintains a mailing list where list members discuss all things Linux? Join LJ's linux-list today:
<http://lists2.linuxjournal.com/mailman/listinfo/linux-list>

LINUX JOURNAL

At Your Service

MAGAZINE

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

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

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

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

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

ON-LINE

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

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



We use our stuff.

visit our TS-7200 powered website at

www.embeddedARM.com

(480) 837-5200

Multi-core

www.pgroup.com

diff -u

WHAT'S NEW IN KERNEL DEVELOPMENT

recently posted a patch supporting that interface, and also added support for the Dreamcast keyboard. Users now can interact with Dreamcast after they have booted Linux on it successfully.

One consistent problem with Linux is an ever-burgeoning growth of **compile-time warnings**. Many of these don't indicate a serious problem, so developers tend to perform only the fixes that will make their code actually work, leaving any remaining warnings still clinging to the compiler output. Every once in a while, some brave soul goes through the great mass of the kernel source tree, scraping off as many warnings as possible. **Satyam Sharma** did it this time, spending a weekend slogging through the tree. The result is that now the truly significant warnings will be more visible in the compiler output, and developers will have an easier time debugging their code. Inevitably, warning cruft will build up again, until someone takes another weekend to play kernel dentist and scrape them off once more.

Marc Espie recently voiced the longtime concern among **BSD folks** that Linux people have been taking unfair advantage of the right to relicense BSD code or to choose only the GPL to cover dual-licensed code. He said this meant people were not "giving back" to the BSD community by making sure the code could not be reused in BSD- or ISC-licensed code. Marc's (and others') argument was that people should be "ethical" about how they utilize the terms of the BSD or ISC license. But, BSD people often tout this very difference between the BSD-type licenses and the GPL as showing that the BSD licenses are "freer". Saying it's unethical for people to make use of that freedom makes it seem as though the BSD people want to have it both ways. They want the "giving back" features of the GPL, and they want to retain the ability to criticize the GPL as being "less free". The solution is simple—use a license that does what you want, instead of hoping naively that everyone agrees that what you want is what they should do.

Zhang Rui noticed that some drivers were creating identically named files in a

Sega's Dreamcast serial bus provides a proprietary interface for mice, keyboards and other peripherals on that architecture.

Adrian McMenamin

single directory of the /proc filesystem. This was clearly not how things should be, so he posted a patch to return an error message if any driver attempted to create a file that already existed in that tree. But, as it turns out, you can't just do that. As **Andrew Morton** said, Zhang's change would cause a lot of currently working systems to break suddenly, as drivers would find their previously successful actions no longer to be successful. **Oliver Neukum** also told Zhang that merely preventing the files from being created—as Zhang's patch did—would simply hide the fact that all these file duplication bugs existed in the driver code. At Andrew's suggestion, therefore, Zhang submitted a new patch to detect the duplication and create a warning message in the system logs. This would make it easier for the driver maintainers to fix the broken code themselves over time.

IDE-CD and **IDE-SCSI** both have been orphaned. **Alan Cox** had been maintaining IDE-CD, but he gradually lost interest in that architecture and didn't really have the hardware anymore anyway. He posted a call for someone to step up as maintainer, and when no one did after a week and a half, submitted a patch to mark IDE-CD as orphaned. **Bartłomiej Zolnierkiewicz** made the same patch for IDE-SCSI. It's possible, as **Jens Axboe** suggested at one point in the discussion, that these drivers might be folded into the main IDE subsystem code for easier maintenance.

It's often the case, especially for relative newcomers to kernel development, that people try to submit a patch to the mailing list, only to discover that their e-mail client has wrapped lines, converted tabs to spaces or vice versa, or that they've submitted the patch as an attachment instead of inline and so on. Some of these problems can be avoided directly, but problems with the e-mail client can be a surprise to users who think they've been careful and done everything right. **Randy Dunlap** recently wrote up a document describing how to configure the various e-mail clients for as few unpleasant surprises as possible. **Jeff Garzik** had asked him to do it. Randy's initial draft contained a fair bit of advice and configuration information and was met with a barrage of additional information from users of various e-mail clients—clearly a welcome document whose time has long since arrived.

—ZACK BROWN

LJ Index, January 2008

1. Novell's Linux revenue in millions of dollars over the last reported nine months: **100**
2. Percentage increase of the above over the prior fiscal year: **243**
3. Number of square feet in the Microsoft-Novell joint development lab: **2,500**
4. Red Hat revenue in millions of dollars over the last reported quarter: **127.3**
5. Percentage growth rate represented by the above: **28**
6. Percentage of Russian school computers onto which Linux will be installed by 2009: **100**
7. Number of Linux computers at the Takoma Park, Maryland, public library: **28**
8. Thousands of times the above computers were logged in to over the last year: **40**
9. Linux's market percentage share of Internet-connected PCs in January 2006: **.29**
10. Linux's market percentage share of Internet-connected PCs in December 2006: **.37**
11. Linux's market percentage share of Internet-connected PCs in September 2007: **.81**
12. Percentage rate of growth for Linux's share of Internet-connected PCs in 2007, so far: **219**
13. Millions of dollars invested in Linux and open-source companies by venture capital firms in Q3 2007: **77.8**
14. Millions of dollars invested in Linux and open-source companies by venture capital firms in Q1 through Q3 of 2007: **226.7**
15. Millions of dollars paid for XenSource (open-source virtualization) by Citrix in August 2007: **500**
16. Millions of dollars paid for Zimbra (open-source e-mail collaboration) by Yahoo in September 2007: **350**
17. Position of "Make Ubuntu laptops cheaper than Windows laptops (in all countries)" among the most popular ideas at Dell IdeaStorm: **1**
18. Position of "Make Linux and no operating system standard options on all future products" among the most popular ideas at Dell IdeaStorm: **2**
19. Position of "Put Ubuntu on the list of operating systems when building a PC!" among the most popular ideas at Dell IdeaStorm: **3**
20. Number of results of a search on dell.com for "linux": **648**

Sources: 1–5: CNET.com | 6: CNews Russian IT Review | 7, 8: Phil Shapiro in PCWorld.com | 9–12: NetApplications.com, via Datamation | 13, 14: The 451 Group, via InternetNews.com | 15, 16: InternetNews.com | 17–19: DellIdeaStorm.com on October 11, 2007 | 20: Dell.com on October 11, 2007

Bug Labs Debugs the Hardware Business

Bug Labs is a startup with the laudable (and long overdue) ambition of making hardware building “just as easy as writing software or Web applications”. That’s from the index page. Dig down and find that “Bug Labs envisions a future where CE stands for Community Electronics, the term ‘mashups’ applies equally to hardware as it does to Web services, and entrepreneurs can appeal to numerous markets by inventing ‘The Long Tail’ of devices.” In an early blog entry, Bug Labs CEO Peter Semmelhack added, “It’s Legos meets Web services and APIs. Imagine being able to build any gadget you wanted by simply connecting simple, functional components together. Now imagine being able to easily program, share and connect these gadgets in interesting ways...”

At the heart of the system is the BUGbase, described as “a fully programmable and ‘hackable’ Linux computer, equipped with a fast CPU, 128MB RAM, built-in Wi-Fi, rechargeable battery, USB, Ethernet and small LCD with button controls. It also has a tripod mount because, well, why not?”

What matters most here, second only to Bug Labs’ ambitions, is its beliefs and practices around Linux and open source. Adds cofounder Peter Semmelhack, “In essence, we’re building an open-source-based platform for programmers to build not only the applications they want but the hardware to run it on.”

I asked Peter to provide some technical

background that would interest *Linux Journal* readers. Here’s his response:

Every piece of code on the BUG is GPL or GPL-compatible. The pieces from the OSS world that we’ve leveraged (or are leveraging) are on our Web site here:

www.buglabs.net/oss_list.

We are using Linux 2.6.19. In addition, the hardware schematics, BOMs and so forth will be made freely available to encourage users to hack, extend and enhance it.



For
our interface

between the hardware and software worlds, we chose to use RESTful Web services. This way, it’s easy to mash up the data coming from the modules with any other Web service. Think of all the cool things you could do if you could mash up the camera on your cell phone, the lat/lon from your

car’s GPS and Flickr, and so on.

Right now, that’s impossible, but with our architecture, it’s pretty straightforward. It’s like any other software mashup. Our SDK/IDE essentially is an Eclipse plugin and offers an integrated, visual way to configure and program your BUG. It also provides a window into BUGnet—our community site—where you can copy applications to/from, rate existing apps, participate in forums and so on.

There are three pieces to the BUG “experience”: the BUG hardware, the Eclipse SDK/IDE and BUGnet. They all work seamlessly together as a whole product out of the box. Lastly, I want to mention that while we chose Java as the language to launch with, there is nothing stopping engineers from using whatever other language they’re comfortable with: C, C++ and so forth. We intend to have explicit support for those in the SDK/IDE shortly.

The Bug Labs site and blog are also full of references to free software and open source (they’re especially fond of Ubuntu) and invitations to participate in both the FOSS value system and Bug Labs’ development community. Check it out at buglabs.net.

—DOC SEARLS

More from Less

We’ve come a long way since March 2003. That was when Paul Otellini of Intel (then COO, now CEO) was on stage at PC Forum, promoting the chip giant’s new Centrino brand. I was the first questioner, and I pressed him to stop dragging the company’s feet on Linux issues, especially device drivers (see www.linuxjournal.com/article/6794). His response was not encouraging.

Now, almost five years later, a Google search for Linux Intel device drivers brings

up more than six million results. Today, Intel pushes Linux as much as Linux pushes Intel. A good example of the former is LessWatts.org, a new community launched by Intel to improve power management on Linux.

Says the index page, “LessWatts.org is not about marketing, trying to sell you something or comparing one vendor to another. LessWatts.org is about how you can save real watts, however you use Linux on

your computer or computers.”

Among the projects featured are the PowerTOP toolkit, “tickless idle”, Power Policy Manager, Linux BLTK (Battery Life Toolkit), display and graphics power saving, device and bus power management and so on. It’s also part of Intel’s Climate Savers initiative (at climatesaverscomputing.org).

Involvement is invited. Visit lesswatts.org to find out more.

—DOC SEARLS

Linux Planned for a Majority of New Embedded Projects

“Linux in the Embedded Systems Market” is the latest report from VDC on developments in the embedded systems market, and it shows huge future advances for Linux in a category it already leads. Figures 1 and 2 (from VDC, vdc-corp.com) are two pie charts that tell an interesting story.

The report lists a number of reasons engineers increasingly favor Linux. These include, “royalty-free run-time costs, advanced networking capabilities and technical features, the large base of

engineers familiar with the Linux operating system, as well as many other factors”.

But the reasons hardly matter. These results are not only about share of market, or even share of mind. They’re also about minds that are already made up. For embedded development purposes, Linux is the practical choice. End of story. The next question is “How much more great embedded work will get done because the standard is now clear?”

—DOC SEARLS

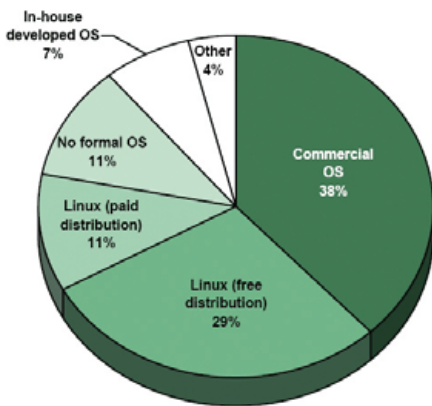


Figure 1. Operating System Used on the Previous Project

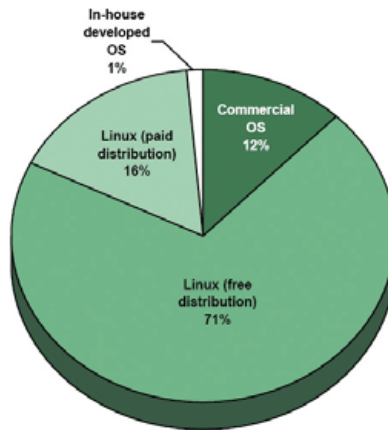


Figure 2. Operating System Planned for the Next Project

Resources

VDC Report: www.vdc-corp.com/

[PressCenter.asp?viewtype=detail&id=1394&pagesection=esw](http://www.vdc-corp.com/PressCenter.asp?viewtype=detail&id=1394&pagesection=esw)

Linux Devices Story: www.linuxdevices.com/news/NS2335393489.html

They Said It

What’s most fascinating to me is that members of the public have no clue that they’re not using Windows. They’re able to load up their Microsoft Word files using OpenOffice.org, and save them back to disk automatically in MS Word format. They surf the Web, check e-mail, do instant messaging, view YouTube videos, visit their Facebook pages, learn touch-typing skills and lots more. Our public library has been offering these Linux public stations for the past three years. People come up to me and ask, “What does Linux look like?” and I answer them with a straight face, “The computer you’ve been using for the past two hours is Linux.”
—Phil Shapiro, blogs.pcworld.com/communityvoices/archives/2007/10/linux_not_ready.html

Google is hot right now. Microsoft circa 1992 hot....So what’s the secret? I think it’s open-source management.

Google empowers its people to try things out, to put them “in beta”. And it leaves them there even when they’re not pulling their financial weight, because someone else may come along with a Clue, and the cost of leaving a server running is a rounding error. By letting people pound on its ideas in public, in other words, Google saved a ton of money...
—Dana Blankenhorn, blogs.zdnet.com/open-source/?p=1533

Regardless of its modest market share in absolute terms, the fact that Linux more than doubled suggests it is growing at a collision course with the other OSes. If it were to maintain its current growth rate, it would be the dominant OS by the year 2014.

—James Maguire in Datamation, itmanagement.earthweb.com/entdev/article.php/3704431

...trying Linux—especially if you boot it from a CD—is a great way to find out what a lot of open-source adherents are so excited about.

And with prices starting as low as free, you certainly cannot complain about the price.

—Larry Magid in the *New York Times*, www.nytimes.com/2007/10/04/technology/circuits/04basics.html



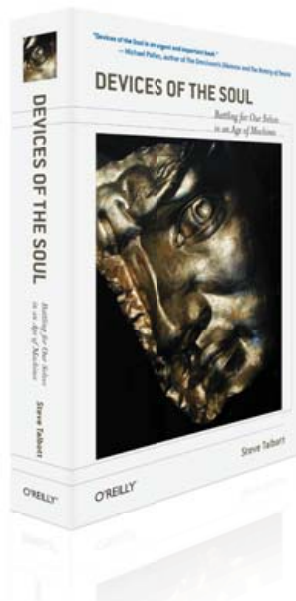
Probe the secrets of modern technology.

Discover some surprising insights with these three books from O'Reilly.



The Myths of Innovation

How do you discover the next big idea? According to bestselling author Scott Berkun, our beliefs about innovation are based on romantic ideas, like the “aha!” moment. In his look at innovation history, including the software and internet age, Berkun reveals powerful truths about how ideas really become successful innovations—truths that people can apply to present day challenges.



Devices of the Soul: Battling for Our Selves in an Age of Machines

With this collection of essays, Steve Talbott challenges us to take an objective look at the technology driving our lives. In an era when 65% of American consumers spend more time with PCs than with loved ones, Talbott insists that something vital is slipping away—our selves. Talbott isn't anti-technology, but he does raise vital questions about the way we conduct our lives.



Beautiful Code

In this book, 38 master coders explain how they solved difficult problems in software development, and why those solutions are so appealing. This isn't a design patterns book, or another software engineering treatise. Today's best programmers think aloud as they work through a project's architecture, the tradeoffs made in its construction, and those moments when it was important to break rules.

From the search for the ultimate solution, to the truth about innovation, to the effects that technology has on society and personal growth, these books will change the way you view technology. **Buy 2 books, get the 3rd FREE!** Use discount code OPC10. All orders over \$29.95 qualify for free shipping within the US.

O'REILLY®

Spreading the knowledge of innovators



oreilly.com

©2007 O'Reilly Media, Inc. O'Reilly logo is a registered trademark of O'Reilly Media, Inc. All other trademarks are the property of their respective owners. 70756



REUVEN M. LERNER

Working with Facebook

Writing a Facebook application? It's easy to retrieve information about Facebook users and their friends as well as display it, using the RFacebook plugin for Rails.

Web sites have become increasingly sophisticated during the past few years, providing a wide variety of applications to the public at large. Many popular sites now offer a variety of APIs, making it possible to interact with the sites, or just retrieve data, from within a program other than an interactive Web browser.

One of the most sophisticated and popular APIs to be unveiled in recent months is from Facebook. Facebook, as you probably have heard, was started by Mark Zuckerberg when he was a student at Harvard. He has since dropped out of college and has led Facebook to be one of the largest and best-known social-networking Web sites, offering people a chance to find and connect with friends and individuals with similar interests.

Facebook has become enormously popular in the last few years, particularly among US university students. But in mid-2007, Facebook unveiled an API that was far beyond what most other sites were doing. This API did not make it particularly easy to retrieve data from Facebook or to perform searches against its extremely large database. Rather, it was designed to let individual developers create new applications that could fit into Facebook's existing site.

If the first few months are any indication, Facebook's application platform has been a wild success. According to a report published by O'Reilly Radar in October 2007, more than 4,000 applications for Facebook have been released since the platform was first unveiled. Some applications have become staggeringly popular; the report estimates that these applications get more than 30 million page views per day, which works out to more than 2% of all Facebook page views.

Other social-networking sites have realized that they must respond in kind. Both LinkedIn and MySpace are (at the time of this writing) working on APIs of their own. But, it remains to be seen if their APIs will provide the deep integration that Facebook is offering. Granted, not every Facebook application is good, and many of them are getting far fewer than the millions of users enjoyed by the top tier.

Whether Facebook turns out to prevail in the social-networking wars is an interesting topic to debate, and it is being discussed at length by business reporters and those interested in what's known as Web 2.0. What's more interesting to us, as Web/database developers, is the fact that Facebook has provided programmers with an enormous opportunity, making it possible for us to add our own

applications to their site.

Last month, we created a simple "Hello, world" application that lived on our own server and was powered by Ruby on Rails. But, this application wasn't designed to be served up on its own. Rather, it is meant to be invoked via Facebook. When people go to the URL `http://apps.facebook.com/rmljfatf`, they will stay on Facebook, with the look and feel of the page remaining that of Facebook. But the contents of that page—currently, nothing more than "Hello from Facebook"—are generated dynamically by a Rails application sitting on my server, `atf.lerner.co.il`. Think of Facebook as a giant, smart proxy server, transparently passing certain HTTP requests to my server whenever someone tries my application.

This month, I explain how Facebook lets us do much more than display "Hello, world" messages. I show how we can retrieve and display information from Facebook and take an initial look at how we can use Facebook's FBML markup languages.

I also continue to develop the application I created last month—named `rmljfatf`—which I created using the Ruby on Rails framework in general and the RFacebook plugin for Rails in particular. See Resources for information on where to obtain this software.

Getting Information from `fbsession`

Last month, we saw how we could create a very simple "Hello, world" application using Ruby on Rails and RFacebook. However, it's not that exciting to produce such output. For example, how do we know that the person is really logged in to Facebook? (Beyond the fact that the page is rendered under the `apps.facebook.com` hostname and has the look and feel of the Facebook page, that is.) And, where are all the nifty, cool Facebook features we have come to expect, which we would expect to use from within a Facebook application?

If this were a normal Web/database application, we simply would create an SQL query, retrieve information about the current user from the database and display it. For example, if we were interested in retrieving a list of the current user's friends, we would write something like this:

```
SELECT F.friend_two_id, P.first_name, P.last_name
FROM Friends F, People P
WHERE F.friend_one_id = 123
AND F.friend_two_id = P.id
```

The above, of course, assumes that we have two tables. The first table is named `People`, in which each person has an ID, a first name and a last name. The second table is named `Friends`, and it indicates who is friends with whom; each friendship is indicated with the `friend_one_id` and `friend_two_id` columns, each of which is a foreign key to `People.id`. Modeling friends in this way requires two rows for each friendship. This might not be the best way to keep track of links, but it reduces the complexity of the logic in SQL queries.

If we were using a straight Rails application, we could eliminate the SQL altogether, relying on the automatic way in which Rails retrieves such data. For example, we could say:

```
@friends = @person.friends
```

This automatically would fire off an SQL query not unlike the one we saw above, albeit behind the scenes. The advantage is not only that we get to write (and read and debug) less code, but also that we can think at a higher level of abstraction, looking at our users in terms of people and links, rather than rows, columns and tables.

Either of these techniques would work fine with Facebook, except for one little problem: we don't have access to the database. Rather, we have to ask Facebook for data, authenticating ourselves as a particular user within a particular application. Only after we have told Facebook who we are can we gain access to the data. Moreover, Facebook makes it easy for users to share only particular pieces of information with third-party applications (and other users), so you cannot be sure you will have access to everything.

fbsession

Much of the Facebook developer documentation has to do with the ways in which you can retrieve information about current users and their friends. However, we will ignore that for now, because `RFacebook` pulls all of that together, as well as the authentication tokens that you need, into a single `fbsession` function. For example, you can write:

```
@friend_uids = fbsession.friends_get.uid_list
```

and `@friend_uids` will be populated with a list of the user IDs for the current user's friends. We even can display this:

```
@friend_uids = fbsession.friends_get.uid_list
render :text => "<p>#{@friend_uids.join(', ')}</p>"
return
```

To review, `fbsession` is our handle into the Facebook API. `fbsession.friends_get` is not merely an array of friends; rather, it is an object of type `Facepricot`. If this

seems like an odd name to you, consider that a popular XML-parsing tool for Ruby is called `Hpricot`. As you can imagine, `Facepricot` is a Facebook-specific extension of `Hpricot`, which allows you to navigate through the response as if it were an `Hpricot` document or use Facebook-specific shortcuts. One such shortcut is seen above, as the `uid_list` method. Although we also could have retrieved the list of friend uids using `Hpricot`, this is more natural, as well as more readable and terse.

Indeed, we also could have written the above code as:

```
@friend_uids =
fbsession.friends_get.search("//uid").map{|xmlnode|
xmlnode.inner_html}
render :text => @friend_uids.join(', ')
return
```

But, unless you're doing something particularly complicated, you probably don't want to that.

Displaying Friends

Once we have retrieved the user's friends' uids, we can ask Facebook to give us some information about each one, using `fbsession`'s `users_getInfo` method:

```
@friendsInfo =
fbsession.users_getInfo(:uids => @friend_uids,
:fields => ["first_name", "last_name"])
```

Notice that we're using instance variables (names starting with `@`) rather than plain-old variables. This ensures that the variables will be visible within our views. For example, we could render the above within our controller:

```
@friends_info =
fbsession.users_getInfo(:uids => @friend_uids,
:fields => ["first_name", "last_name"])

output = ""
@friends_info.user_list.each do |friend|
output << "<p>#{friend.first_name} #{friend.last_name}</p>\n"
end

render :text => output
return
```

In the first line, we use `fbsession.users_getInfo` to invoke the `getInfo` method from the Facebook API. (Indeed, `fbsession` provides us with an interface to the entire Facebook API, albeit with some character translation along the way.) `users_getInfo` takes two parameters: a list of user IDs about which to retrieve information and then the fields we want to retrieve about them.

For example, perhaps we want to find out whether each of our friends is male or female, as well as how many messages they have on their wall. We can do this by modifying our `users_getInfo` query, as well as by

changing our output:

```
@friends_info =
  fbsession.users_getInfo(:uids => @friend_uids,
    :fields => ["first_name", "last_name",
      "sex", "wall_count"])

output = ""
@friends_info.user_list.each do |friend|
  output << "<p>#{friend.first_name} #{friend.last_name}
    (#{friend.sex}), with #{friend.wall_count} hits on their wall.</p>\n"
end

render :text => output
return
```

Sure enough, this produces a list of our friends, along with their stated sex and the number of hits on their wall. Behind the scenes, our call to `users_getInfo` is sending a request to Facebook's servers. Facebook authenticates our request and then sends a response. Although the response is in XML, the Facepricot object provides us with some convenience functions that make it easy to work with what it provides.

A Nicer Display

The above code might work, but you would be hard-pressed to say that it was elegant. If nothing else, Rails programmers are consistent about their praise for the MVC paradigm in Web development. That is, you want to have a clear separation between the back-end data model, the controller that handles business logic and the way in which displayed items are rendered on the user's view or screen.

Luckily, it's easy to modify the way in which these things are displayed. Rather than collecting the textual

output in a variable (named `output` in our above examples), we can define our entire method as:

```
def facebook
  @friend_uids = fbsession.friends_get.uid_list

  @friends_info =
    fbsession.users_getInfo(:uids => @friend_uids,
      :fields => ["first_name", "last_name",
        "sex", "wall_count"])
end
```

We then create (or modify, if you still have your view from last time) `facebook.rhtml`, which looks like:

```
<% @friends_info.user_list.each do |userInfo| %>
<ul>
  <li><%= userInfo.first_name %> <%= userInfo.last_name %></li>
</ul>
<% end %>
```

In other words, we iterate through each element in our list of friends, pulling out their names. We can use all the information we have captured, not just the names:

```
<% @friends_info.user_list.each do |userInfo| %>
<ul>
  <li><%= userInfo.first_name %> <%= userInfo.last_name %> (<%=
  userInfo.sex %>), wall count <%= userInfo.wall_count %></li>
</ul>
<% end %>
```

But, wait one moment—we can do even better than this!

Resources

Facebook developer information is at developers.facebook.com. This includes documentation, a wiki and many code examples. One article on the wiki specifically addresses Ruby development: wiki.developers.facebook.com/index.php/Using_Ruby_on_Rails_with_Facebook_Platform.

Ruby on Rails can be downloaded from rubyonrails.com. Of course, Rails is written in the Ruby language, which is almost certainly included in your distribution, and it also can be downloaded from www.ruby-lang.org.

The RFacebook gem for Ruby and the companion RFacebook plugin for Rails developers can be retrieved from rfacebook.rubyforge.org.

Hpricot, written by the prolific Ruby programmer "why the lucky stiff", is at code.whiteluckystiff.net/hpricot. I have found it to be useful in many Ruby programs I've written, but it is especially useful in the context of RFacebook, given the central role of XML and the Facepricot extension.

Chad Fowler, a well-known Ruby developer, has developed a different Rails plugin (Facebooker) for working with Facebook. You can download the code, as well as learn more about the design principles behind his plugin, at www.chadfowler.com/2007/9/5/writing-apis-to-wrap-apis.

Finally, O'Reilly Media published a 30-page report in October 2007 describing the Facebook application platform. The report is meant for managers and marketing people, but even programmers can learn something from this (admittedly expensive) report, which describes the number of applications that have been deployed, as well as the types of things people are doing. Programmers won't learn enough from this for it to be worth buying, but it might well be worth finding and reading a copy that a more business-oriented friend has bought.

Because we are rendering things within Facebook, we can take advantage of FBML, the Facebook Markup Language. FBML is an extended subset of HTML, which is a fancy way of saying that it adds some Facebook-specific tags while removing some standard HTML tags. In any event, it allows us to create a variety of lists, interfaces and functionality that are common to Facebook applications and include them in our applications. For example, let's change our view to the following:

```
<% @friends_info.user_list.each do |userInfo| %>
<ul>
  <li><fb:name uid="<%= userInfo.uid -%>" target="_blank"
/><fb:profile-pic\
  uid="<%= userInfo.uid -%>" linked="true" /></li>
</ul>
<% end %>
```

Now we're iterating over the same list. But, instead of rendering things directly from Ruby, we're using Ruby to pass the friend's user ID to FBML tags. Each FBML tag takes one or more arguments, passed in the form of HTML/XML attributes. In this case, we have used two

FBML tags: `fb:name`, which displays a user's name, and `fb:profile-pic`, which displays the user's picture.

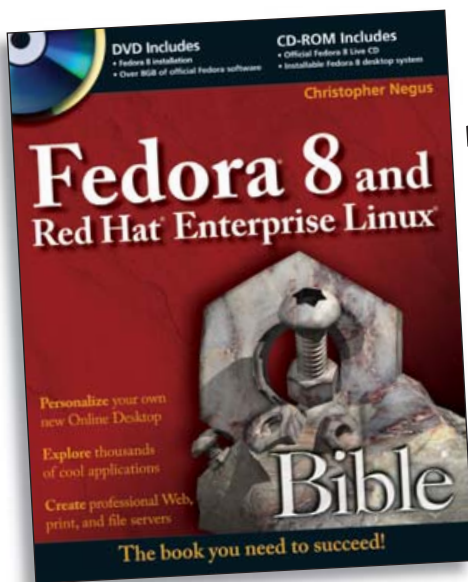
As you can see, we have passed each tag the `uid` attribute, then used some `rhtml` to bring in the user's ID. We also have passed the `linked` attribute to indicate that the picture should be a link to the user's profile. (The name is linked to the profile by default, so we don't need to say anything about that.) I have been quite impressed by the number and types of attributes that Facebook's developer API provides, going so far as to let us indicate whether we want to have the name rendered in the possessive form.

Conclusion

Facebook has provided application developers with a rich and interesting API that goes far beyond retrieving and storing data. It allows us to create applications that truly do sit within Facebook. Next month, we'll look at how we can have a Facebook application that stores its own data and integrates that data along with the user's Facebook profile. ■

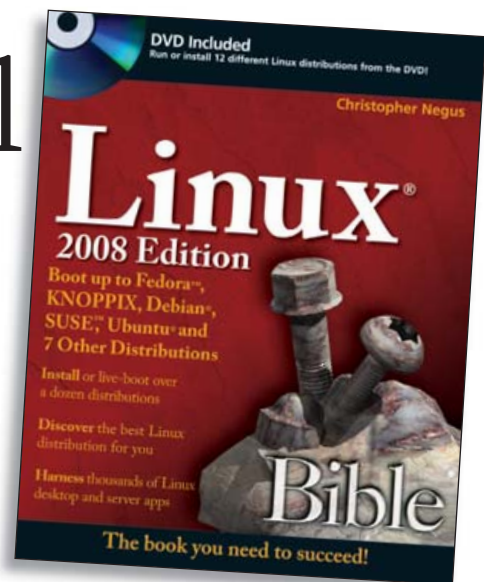
Reuven M. Lerner, a longtime Web/database developer and consultant, is a PhD candidate in learning sciences at Northwestern University, studying on-line learning communities. He recently returned (with his wife and three children) to their home in Modi'in, Israel, after four years in the Chicago area.

Negus knows Linux.®



978-0-470-23020-6

So will
you.



978-0-470-23019-0

Available wherever books are sold.

Wiley and the Wiley logo are registered trademarks of John Wiley & Sons, Inc. All other trademarks are the property of their respective owners.

 **WILEY**
Now you know.
wiley.com



MARCEL GAGNÉ

Security's Front Door

Words, words, words...whether spoken by a fool or a genius, they are still the first line of defense in system security.

What is this I see on our specials chalkboard, François? Mxyztplk? That is the root password for our main server! *Mon Dieu!* What do I see here? Those are all our administrative passwords! Why would you post secret information where everyone can see it? *Quoi?* So you would not forget? But François, neither will anyone else. I see you have posted your own login passwords as well. Please, erase those immediately and wash the chalkboard when you are done. *Merci.* Now, just to be safe, we will need to generate a whole new set of passwords for all our systems. What were you thinking, *mon ami?* Of course, I see. We'll discuss this later. Our guests are arriving now. Prepare yourself, François.

Welcome, everyone! How wonderful to see you here at *Chez Marcel*, home of superb Linux and open-source software and, of course, wine served from one of the world's finest wine cellars. Speaking of wine...François, please head down to the wine cellar, over in the East wing, and bring back the 2005 Sonoma County Kokomo Zinfandel. *Vite!*

Ah, *mes amis*, you missed a rare opportunity to see all of *Chez Marcel's* security, exposed on our Specials du Jour

Pierre's `makepasswd` program uses your computer's random number generator to create passwords of varying constraints.

board. Nevertheless, it does provide an excellent backdrop to our menu this evening, as all the items relate to password security. Passwords, *mes amis*, are still your first line of defense when it comes to computers. Biometric systems, like fingerprint readers, can make secure access more daunting and difficult to breach, but most systems, including countless Web sites, require a user name and password for access, and that's not changing anytime soon. In the end, it usually comes back to passwords, and passwords mean people need to remember them. And, that's where the problem starts.

I've been in offices where people will tell you (if you insist) that everyone pretty much knows everybody else's passwords—just in case. I've seen yellow sticky notes stuck to computer screens with passwords written down so the users don't forget. Even when that information is out of sight, people use simple passwords, like the word "password", because they're easy to remember.

One way to get secure passwords that aren't your pet's

name or your spouse's birthday is to pick a phrase that means something to you, and then play with it. For example, take the phrase "Believe in magic!" Now, take only the consonants of the first and last word, and you have `blvmgc`. Add an `l` at the beginning, but make that `l` a numeric `1` instead. Add an asterisk for the final character, and you have `1blvmgc*`—a great password if ever there was one.

Another, more secure way (particularly if you need many passwords), is to enlist the help of a random password generator. One such program is Pierre "khorben" Prochery's `makepasswd` program (inspired by Rob Levin's Perl script of the same name). Pierre's `makepasswd` program uses your computer's random number generator to create passwords of varying constraints. It also can generate encrypted passwords. You can get a single, random password by typing `makepasswd` at a shell prompt. The program also accepts different parameters on the command line, as shown here:

```
$ makepasswd --chars 8 --count=4
0dAU8BXM
suQt4CF2
5x0yGJ1S
6KTIinj58
```

So, what happened? The `--chars 8` parameter tells the program to use exactly eight characters in the resulting password. You also can specify `--minchars` and `--maxchars` to get different password lengths. The `--count=4` parameter tells the program to generate four passwords. The default is to provide only one password. Type `makepasswd --help` for a full list of parameters.

Shell users know this well, but those who take the time to learn the ins and outs of their Linux systems learn this too; many graphical programs are front ends to one or more text- or shell-based commands. The same is true for the next item on our menu, `KriptPass`, which wraps the `makepasswd` program in a nice, graphical interface. `KriptPass` is a Kommander script available from Kriptopolis.org. Kommander is a combination program editor and executor that can be used to create any number of graphical applications using the KDE framework. I mention Kommander, because you need it to use `KriptPass`. So, installing Kommander is the first step. Because it's a KDE application, check your system to see whether you already have Kommander installed. If you don't, check your distribution's repositories and install it. Because `KriptPass` is based on `makepasswd`,

Note:

More information on Kommander is available from the official Kommander Web site at kommander.kdewebdev.org.

you need that as well.

Assuming you have Kommander installed, installing and running KriptPass is as simple as downloading it from www.kde-apps.org/content/show.php/KriptPass?content=58800. Extract the script wherever you like, open up Konqueror and simply click on the `kriptpass.kmdr` file. That's all there's to it (if you like, you can add a shortcut icon on your desktop for future use). The KriptPass window appears (Figure 1), and you'll see three tabs labeled Passwords, Wireless Keys and About.

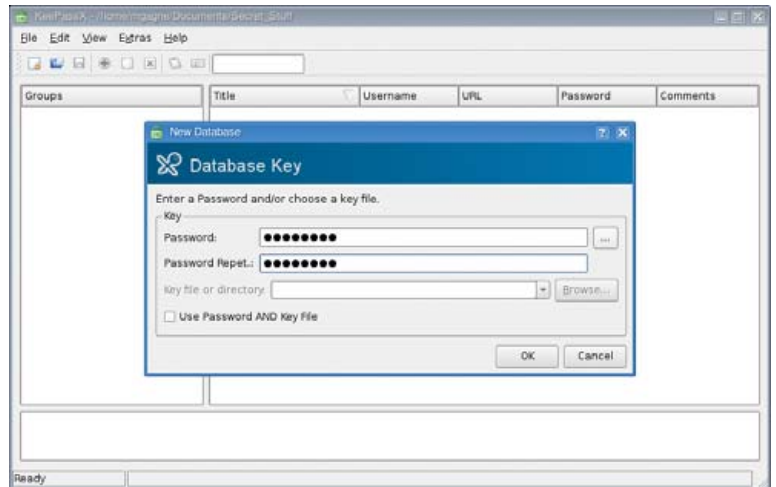


Figure 1. KriptPass is a Kommander script that provides a graphical front end to the text-based `makepasswd` command.

It's frightfully easy to use. Select the Password length and the number of passwords you want to generate, and then click Generate. You can cut and paste your new password into whatever application requires a password change. You also can save those passwords to a file by clicking the Save to file check box and selecting a name. If you want your password to use specific characters, check Modify Character Set, and enter your characters. The default uses the ten digits as well as the 26 letters in uppercase and lowercase—just like that, totally random passwords. Increase the password length, and your passwords will be even more secure.

The only catch—and this is the catch with any random, non-pronounceable password—is that the passwords are hard to remember, which, sadly, leads to people writing them down and potentially compromising security. How do we deal with this problem?

Tarek Saidi's KeePassX is a great place for this information.



This password manager and data safe provides a secure location for your vast collection of user names and passwords. It's also a cross-platform application that runs under Mac OS X and Windows too. If, like many people, you work on multiple systems and need access to your information, you can copy the database to a USB key and carry it with you. To get and start using KeePassX, visit keepassx.sourceforge.net, or check your distribution's repositories for prebuilt packages (some binary packages are available at the KeePassX Web site).

When you start KeePassX the first time, you'll see that it is divided into two main panes. The left pane is labeled Groups. To the right, in the larger section, are headings for Title, Username, URL and so on. To begin, you need to create a new password database. Click File on the menu bar, and select New Database. A dialog appears asking for a password, which you must enter twice (Figure 2).

The database itself is encrypted using 256-bit AES by default, but you also can select 256-bit Twofish. The number of rounds to encrypt is 6,000, making this a very safe place for your personal information. However, don't ever forget that master password. If you want to change the encryption format or the number of rounds, click File and select Database Properties from the menu bar.

The next step is to enter a group. Click Edit on the menu bar and select Add New Group. The Group Properties dialog appears (Figure 3). This is purely informational and serves as a folder for storing passwords. So, enter a title that means something to you,



Figure 3. When creating a group, you can select an icon to represent the type of information you are storing.

Figure 2. Before you can store anything in KeePassX, you need a database. You can create multiple databases if you want.

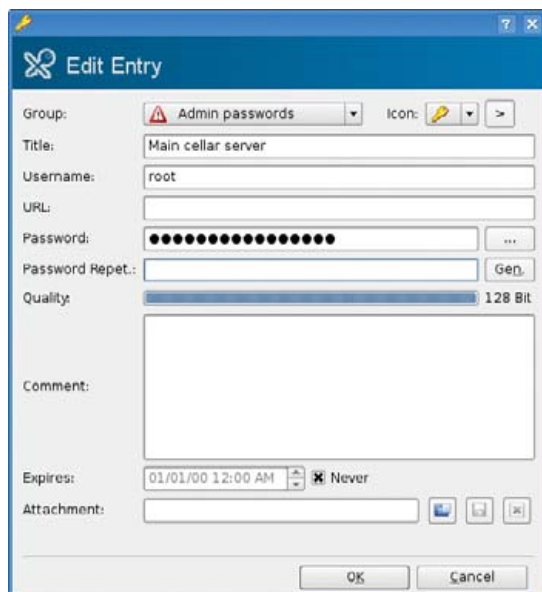


Figure 4. Add a password entry to your KeePassX safe.

then select an icon from the drop-down list. Click OK when you are done.

You can create as many of these as you like with names like System passwords, Family PCs (if you are doing the administration on your family's systems), Customer systems and so on. The groups will appear in the Groups column. Select a group, click Edit on the menu bar and then select Add New Entry (or click the plus sign on the icon bar). The Edit Entry window appears (Figure 4). The Group is selected automatically, but if you want, you can choose another at this point. Enter a title to identify the entry, then enter your user name and password information. As you enter your password, the quality of the password is analyzed and reported on the Quality bar. You can add a comment if you like, but it isn't necessary. Additionally, you can select an expiration date, attach a file or simply click OK if you are done.

Look closely to the right of the Password Repet. field, and you'll see a button labeled Gen. Given the earlier programs we've looked at, this might sound interesting, *non*? Click the button, and a password generator appears (Figure 5). KeePassX's password generator allows you to define what characters are included in your password, such as the use of special characters, spaces



Figure 5. If you prefer, KeePassX can generate a password for you.

and so on. You also can define the password length; the default is a difficult-to-crack 20 characters.

Click Generate and your password appears in the New Password field. If you like what you see, click Accept. In some ways, this brings us back to where we started, using a tool to generate secure passwords rather than relying on common words or phrases.

Later, when restarting the program, KeePassX challenges you with your master password before giving you access to the safe. If you are the sort of person who needs a tool like KeePassX, you also will have numerous passwords to look through when checking for a login you haven't used in ages. For that inevitable day, KeePassX provides a quick search feature, right on the main window at the far right of the icon bar

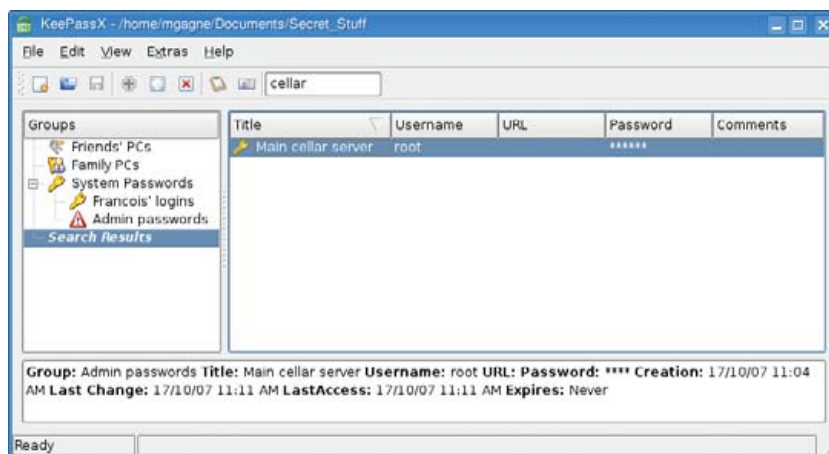


Figure 6. A quick search field is available at the top right of the icon bar. Simply enter part of your title, press Enter and your information is located quickly.

This password manager and data safe provides a secure location for your vast collection of user names and passwords.

(Figure 6). Enter one or more words in your title or comments, then press Enter. To see the actual password, double-click on the result, then click the ... button next to the hidden password.

I can see that closing time has arrived, *mes amis*. Given François' penchant for exposing sensitive information, I may do the locking up myself tonight. Even though I poke fun at François, he still is the best waiter I've ever employed and an artist when it comes to opening and pouring wine. In that, I trust him completely. Please, François, take a moment to refill our guests' glasses a final time. Raise your glasses, *mes amis*, and let us all drink to one another's health. *A votre santé! Bon appétit!* ■

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

Resources

KeePassX: keepassx.sourceforge.net

KriptPass: www.kde-apps.org/content/show.php/KriptPass?content=58800

makepasswd:
people.defora.org/~khorben/projects/makepasswd

Marcel's Web Site:
www.marcelgagne.com

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

Your World Runs Faster With c-tree® Database Technology

A small footprint c-tree database controls the traffic lights on your way to work.



Packages are scanned and delivered using a scalable c-tree database that can run on handheld devices and mainframes.



A high throughput c-tree database validates your credit card transactions.

Your financial transactions are secure because they are authenticated using a c-tree database.



Your digital pictures are well-organized thanks to the transparent deployment of a c-tree database within your photo album software.



Your world runs faster (and you sleep better!) with c-tree database technology.

FairCom provides high performance, low maintenance data management technology. Our customers – ranging from small startups to multinational corporations – are able to overcome application-specific performance dilemmas because c-tree gives them precise control over their database operations. Super-charge your application and simplify your deployment! Download an evaluation edition of c-tree today.



FairCom®

Download FairCom's c-tree Plus Today!

www.faircom.com/go/?usesDLD

High Performance Database Technology • 800-234-8180

Other company and product names are registered trademarks or trademarks of their respective owners. © 2007 FairCom Corporation



MICK BAUER

Get a Clue with WebGoat

Hack, analyze and learn from an intentionally insecure Web application.

As more and more critical applications have adopted Web browser front ends, Web security has become the most critical front in Internet security. And yet, year after year, the same types of Web application security mistakes keep cropping up in security bulletins: SQL injection, cross-site scripting, authentication mechanisms that “fail open” and so forth. How, as a Web developer or administrator, can you avoid making these mistakes with your own Web applications?

The WebGoat can help. Developed by the Open Web Application Security Project (OWASP), the WebGoat is an “intentionally insecure” Tomcat Web application that walks you through common Web security mistakes, exploits and solutions. In this article, I explain how to install WebGoat on your Linux system and use it to educate yourself on Web application security.

Getting and Installing WebGoat

The WebGoat on Linux has a critical dependency: the Java 1.5 Software Development Kit (JDK 1.5). Therefore, make sure you’ve installed your distribution’s package for the JDK 1.5. On SUSE and OpenSUSE systems, this package

install it separately.

To get the latest version of WebGoat, go to either WebGoat’s SourceForge Web site (sourceforge.net/project/showfiles.php?group_id=64424&package_id=61824) or its Google Code Downloads site (code.google.com/p/webgoat/downloads/list). Along with the Windows releases of WebGoat, you’ll find the standalone Web Application Archive file (WAR) version of WebGoat for UNIX/Linux and the “Release” version that includes Tomcat. You should opt for the latter, unless you’ve already got a working Tomcat installation on your system.

On my OpenSUSE system, I unzipped the Release version (Unix_WebGoat-5.0_Release.zip) in my home directory, which resulted in a new subdirectory, WebGoat-5.0 (/home/mick/WebGoat-5.0/). This directory contains a readme file (readme.txt), WebGoat’s startup script (webgoat.sh) and another directory, tomcat, that contains the Tomcat servlet engine plus, of course, the WebGoat WAR file.

Adjunct Tools

Now that you’ve installed WebGoat and the things on which it depends, but before plunging into Web-hacking mayhem, there’s one other thing you need: a good graphical local Web proxy. Not a proxy *server* like Squid; rather, a local proxy you can use to intercept, view and alter the data your Web browser sends to Web servers. This is a critical tool in the Web hacker’s bag of tricks—it’s also very useful for Web developers who need to troubleshoot their own Web applications—and you’ll need it to complete many lessons in WebGoat.

OWASP recommends the official OWASP proxy, WebScarab, which is available at www.owasp.org/index.php/OWASP_WebScarab_Project. WebScarab is a free, full-featured Web proxy and spider (a spider follows all links on a Web site, effectively cataloging it), all with a convenient GUI. It’s also written in Java, which means it’s cross-platform.

WebScarab can be downloaded as either a “self-contained” JAR file (Java archive) and run with the command `java -jar ./webscarab-selfcontained-20070504-1631.jar` or as an installer (currently `webscarab-installer-20070504-1631.jar`) that, when executed via the command `java -jar webscarab-installer-20070504-1631.jar`, unpacks WebScarab into the WebScarab directory in your home directory and installs a shortcut in your KDE or GNOME start menu. Start WebScarab either via this shortcut or by executing the self-contained version with the `java -jar` command.

Now that you’ve installed WebGoat and the things on which it depends, but before plunging into Web-hacking mayhem, there’s one other thing you need: a good graphical local Web proxy.

is called `java-1_5_0-sun-devel`. On Debian and Debian-derived distributions, like Ubuntu, it’s called `sun-java5-jdk`. Note that Red Hat and its derivatives don’t have their own JDK 1.5 packages; see Resources for links to two articles that may help.

Your JDK 1.5 package’s setup script should set your `JAVA_HOME` environment variable to the JDK’s root directory. (On my OpenSUSE system, this is `/usr/lib/jvm/java`, which is actually a series of links to `/usr/lib/jvm/java-1.5.0-sun-1.5.0_12/`.) You may need to log out and back in for this variable to “take”, but regardless, it must be set correctly for WebGoat to run. If in doubt, do an `echo $JAVA_HOME` from a bash session to check to see whether it’s set correctly.

Note that you do *not* need Apache installed to run WebGoat. In fact, if it is installed, I recommend you shut it down. WebGoat runs on its own bundled Tomcat installation, so although Tomcat *is* required, you don’t need to

Another good graphical, Java-based local Web proxy is Paros, available at www.parosproxy.org/index.shtml. Maybe I'm just more familiar with it, but I prefer Paros' interface. In my opinion, it's a little more friendly to non-expert users. You be the judge—both WebScarab and Paros are free, so there's no reason not to give each of them a spin.

Paros comes in the form of a zip file that decompresses to the directory `paros`. Inside, among other things, are a couple different versions of a startup script. The one you want is called `startserver.sh`. Start it with the command `sh ./startserver.sh`.

Note that you don't need to be root to install or start either proxy. In fact, there's no good reason for you to be root, because both proxies, by default, listen on the unprivileged port TCP 8008. You can change the listening port in WebScarab's Listener tab or in Paros' Tools→Options→Local proxy screen.

Configuring Your Browser

You're almost ready to start WebGoat, but there's one last thing to do: configure your Web browser to direct all traffic to your local proxy (for example, WebScarab or Paros). This is done in precisely the same way as specifying a proxy server; to a Web browser, a local proxy and a proxy server are the same thing. The only real difference is that instead of a proper fully qualified domain name, you need to give the name `localhost` or the IP address `127.0.0.1`. Be sure to specify the correct port too—the one on which your local proxy is listening (8008 unless you changed it).

Figure 1 shows the proxy configuration screen in Firefox. Access this screen from the Edit menu: Edit→Preferences→Advanced→Network→Connection→Settings. This dialog is very similar in other browsers. If your browser's proxy configuration dialog has a proxy-exceptions field, like Firefox's "No Proxy for:" box shown



Figure 1. Configuring Firefox for Proxy Use

in Figure 1, make sure this field is blank or that it at least does *not* contain the values `127.0.0.1` or `localhost`.

Starting WebGoat

Now, you've got JDK 1.5, you've got a local Web proxy running, and you've reconfigured your browser to use the proxy. It's time to milk the goat!

To start WebGoat, open a bash shell in your terminal window program of choice (I'm still partial to plain-old `xterm`), and change your working directory to the one WebGoat unzipped into—`/home/dartheim/WebGoat-5.0` on my system. You don't need to be logged in as root to start WebGoat, but you do need to start WebGoat with root privileges—for example, via `sudo`, so you will need root's password. Therefore, to start WebGoat listening on TCP port 80, issue this command:

```
sudo sh ./webgoat start80
```

If you're running Apache or some other process on TCP port 80 (though I don't recommend doing so), you can start WebGoat on TCP port 8080, with this command:

```
sudo sh ./webgoat start8080
```

In either case, you'll be prompted for root's password, and WebGoat will start up, logging startup messages and errors both to your shell and also to the file `WebGoat-5.0/tomcat/logs/catalina.out`. Note that you can run WebGoat safely in the background by appending an ampersand (&) to your startup command, but there's little point. Although you'll, thus, be able to issue other commands from the same shell, all those log messages still will make that

Local Web Proxies: Good or Evil?

You need to use a local Web proxy in order to carry out many of the practice attacks in WebGoat. But the only thing that makes them "practice" attacks is the fact that you're running them against an educational Web application running on your own system. The very same tools and techniques can be used for good or evil.

As with any penetration-testing tool, it's up to you to use your local Web proxy responsibly and ethically. If you run WebScarab or Paros when interacting with somebody else's Web server, you may be breaking the law (for example, by probing the site or sending intentionally malformed requests), even if all you're trying to do is learn.

Unless you're being paid to conduct a legal, fully authorized penetration test, you should run these tools only on and against *your own systems*.

So my other piece of advice, besides reading “OWASP Top 10 2007”, is to feel free to jump around between lessons.

particular shell nearly unusable. It's better simply to open another terminal window.

Using WebGoat

And, now, you can log in to WebGoat. Simply point the browser you just reconfigured to `http://127.0.0.1/WebGoat/` attack (or `http://127.0.0.1:8080/WebGoat/attack` if you started WebGoat on TCP 8080). You'll be prompted for a user name and password. Enter “guest” for both values. If you receive the page shown in Figure 2, you're ready to go!



Figure 2. WebGoat Welcome Screen

If you don't see the screen shown in Figure 2, double-check your browser's proxy setting, make sure your local Web proxy is running, and check the terminal in which you started WebGoat for error messages. Also, make sure you used `sudo` to start WebGoat—you might think that using the `start8080` command would make this unnecessary, as TCP 8080 is an unprivileged port, but I've never had any luck running WebGoat as an unprivileged process.

Once you get the welcome screen, click the Start WebGoat button to begin.

WebGoat Structure and Tips

The first lesson in WebGoat is an introduction to the WebGoat interface. It walks you through a simple HTTP transaction and gives you the opportunity to use WebGoat's various buttons: Hints, Show Params, Show Cookies, Show Java and Lesson Plans. The Hints and Lesson Plans buttons are particularly important (your local Web proxy is more useful for viewing HTTP parameters

and cookies), but I find that Lesson Plan pop-ups don't render properly under KDE's Konqueror browser (they render fine in Firefox).

After the first lesson, the lessons in WebGoat tend to assume that prior to running WebGoat, you've done *some* studying of Web application security, by at least reading the report “OWASP Top 10 2007” (available from www.owasp.org, in the “Top 10 Project” section). Therefore, I highly recommend you download the full PDF version of this report, read it carefully, and keep it available while you're running WebGoat.

Some of WebGoat's lessons are self-explanatory; the lesson text itself or the lesson plan tells you everything you need to know in order to complete the exercise successfully. Other lessons, however, are not as straightforward. The Hints button can help, but even then you may find yourself copying and pasting an attack string suggested in a hint, without really understanding how the attack works.

So my other piece of advice, besides reading “OWASP Top 10 2007”, is to feel free to jump around between lessons. They are *not* presented in order of difficulty, nor do later lessons build on earlier ones, as far as I can tell. If a given lesson is too hard for you, feel free to try a different one instead. You always can try the harder lesson again later.

This doesn't mean WebGoat is poorly organized; it just means that the lessons are nonsequential, being instead grouped by type (Code Quality, Invalidated Parameters, Buffer Overflows and so forth). In summary, you will have a much more positive WebGoat experience if you first read the “OWASP Top 10 2007”, attempt whichever lessons strike your fancy, paying attention to each lesson's Lesson Plan button, and click the Hints button as necessary. And, if all else fails, there's always Google!

Using WebGoat: a Sample Lesson

Because WebGoat is itself a tutorial, it would be redundant for me (and beyond the scope of a single article) to walk through every lesson. Similarly, Web proxies are too versatile and complex to explain fully here. (In the future, I may very well devote an entire article to them.) But, I can walk you through a sample WebGoat lesson, including correct proxy use, as an illustration.

One of my favorite WebGoat lessons is the one on fail-open authentication mechanisms. If it's possible to make a user-authentication transaction fail in a way that results in *successful* authentication, that authentication mechanism is said to “fail open”. Needless to say, this is not the way a secure access mechanism behaves! But, it's a common Web coding mistake.

WebGoat covers this under the “Improper Error Handling” section, in the “How to Bypass a Fail Open Authentication Scheme” lesson. To start this lesson, click the “Improper Error Handling” link on the left-hand frame of the WebGoat interface, and then click the “How to

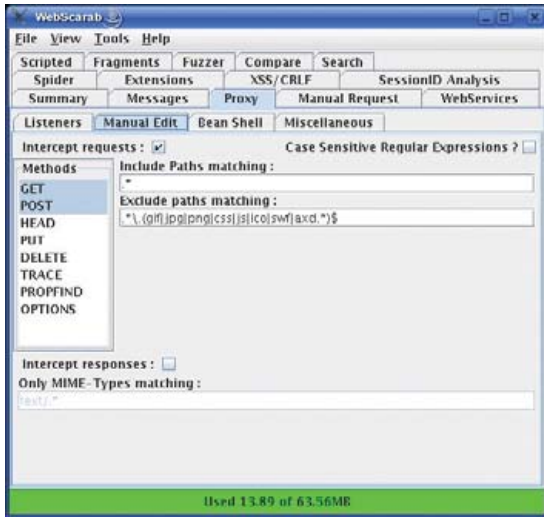


Figure 3. Putting WebScarab into Intercept Mode

Bypass a Fail Open Authentication Scheme” link that appears below it.

In this lesson, you’ll enter a user name and password,

use your local Web proxy to intercept the resulting HTTP request before it reaches the server (WebGoat), edit the request and then send it on its way.

Therefore, you need to put your Web proxy into intercept mode (trap mode on Paros). Figure 3 shows WebScarab set up for interception—all you need to do is check the box next to “Intercept requests”. Paros is similar: click the Trap tab, and check the box next to “Trap request”.

Now, whenever you navigate to a new Web page or submit a Web form, the resulting HTTP request will be intercepted and stopped by your local proxy, allowing you to alter the request before forwarding it on. (WebScarab will pop up an edit window in the same virtual desktop as your browser session; Paros won’t, so you’ll need to click on your Paros window manually.)

Continuing the sample lesson, you now can go back to your browser, and type in the user name webgoat and some arbitrary string for a password (Figure 4).

Be sure your local Web proxy is in intercept/trap mode before clicking the Login button. After you do, WebScarab will pop up an Edit Request window like the one shown in Figure 5.

Expert Included.

As product manager for HPC clusters, Kirtan helps to make the complicated process of sourcing, pricing, and acquiring a cluster as straightforward as possible for customers. Kirtan is pleased to be collaborating with Intel and other members of the Intel Cluster Ready program. He works to develop cluster solutions that will support multiple Intel Cluster Ready ISV applications.

He is proud that Silicon Mechanics has developed an industry-first cluster configurator, part of the RackScale Clusters product line, which allows users to dynamically configure complete cluster solutions – online, interactively.

The combination of Silicon Mechanics hardware, the RackScale Clusters online configurator, and the Intel Cluster Ready program offers customers tremendous value by reducing installation, deployment, and maintenance times for applications in their clusters.

When you partner with Silicon Mechanics, you get more than innovative Intel solutions — you get an expert like Kirtan.



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

Silicon Mechanics and the Silicon Mechanics logo are registered trademarks of Silicon Mechanics, Inc. Intel, the Intel logo, Xeon, Xeon Inside, Intel Cluster Ready and the Intel Cluster Ready logo are trademarks or registered trademarks of Intel Corporation in the US and other countries.

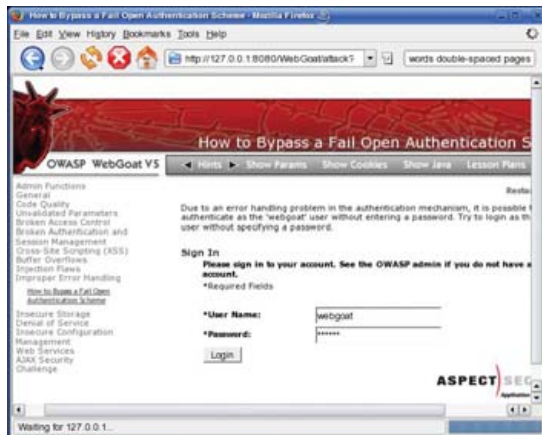


Figure 4. Entering User Name and Password

And now, we work our magic. In this Edit Request window, click the Text tab, select the string Password=bl0bb0&, and *delete* it. That's right, we're erasing the entire Password parameter from our authentication submission. This should result in a failed authentication, right?

But, when you click the Edit Request window's Accept button and switch back to your browser, you'll see the screen shown in Figure 6.

The attack succeeded: you just logged in without knowing, or even attempting to submit, a password! Before clicking on other links, you may want to turn off your Web proxy's intercept/trap mode; otherwise, you'll need to click the proxy's Accept button (Continue in Paros) repeatedly simply to navigate to and load the page.

The last thing you should do before leaving this lesson is, arguably, the most important: click WebGoat's

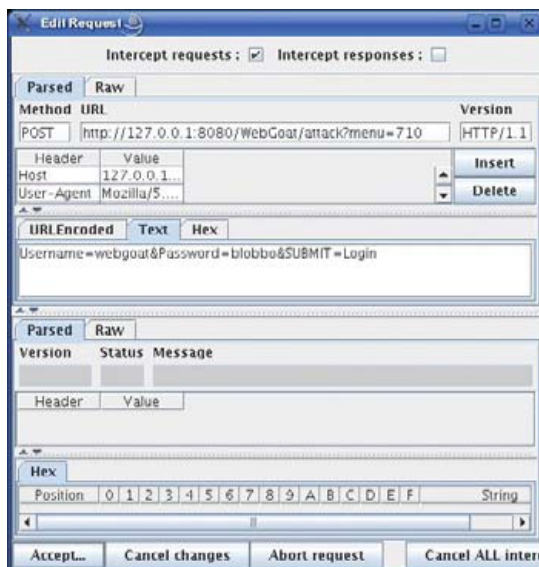


Figure 5. Editing an Intercepted HTTP Request



Figure 6. The Attack Succeeded!

Show Java button, and see exactly what coding errors led to this fail-open condition. You'll find WebGoat's code to be well-commented and easy to analyze, even if you're a novice programmer. Remember, the whole point of using WebGoat is not only to see what can go wrong, but also to learn how to prevent it from going wrong in the first place.

Conclusion

The OWASP Web site contains much more information about WebGoat, WebScarab and Web security in general. You may find the WebGoat User and Install Guide, located in the WebGoat section, especially useful. Be safe! ■

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

Resources

The Open Web Application Security Project home page, with links to its WebGoat, WebScarab, OWASP Top 10 and OWASP Guide Projects: www.owasp.org.

The Paros home page, where you can download the free Paros local Web proxy: www.parosproxy.org.

Russ Herrold's "HowTo Install Java on CentOS 4 and CentOS 5": wiki.centos.org/HowTos/JavaOnCentOS.

Jan K. Labanowski's "Sun Java 6 Development Kit on Fedora Core 7": ccl.net/cqa/software/SOURCES/JAVA/JSDK-1.6.



Are you Shocked

by the high cost of iSCSI & Fibre Channel SAN storage?

AoE is the answer!

ATA-over-Ethernet = **Fast, Reliable, Simple** storage.

www.coraid.com



EtherDrive® SRxxxx

- Fast & Flexible RAID appliances with slots for hot swap SATA disks
- Check out our full line of EtherDrive® Storage and VirtualStorage Appliances and NAS Gateways



1. Fast 10 Gigabit Ethernet Storage without the TCP/IP overhead!
2. Unlimited expandability, at the lowest possible price point!!
3. You want more storage...you just buy more disks – it's that simple!!!

Visit us at www.coraid.com



1.706.548.7200

The Linux Storage People

www.coraid.com



DAVE TAYLOR

Numerology, or the Number 23

Use a shell script to do basic numerology.

I admit it, I watch a lot of movies. In the decades I've been alive (a gentleman doesn't disclose his age!), I've watched tens of thousands of movies, and average about, oh, 6–8 movies/week. Truth be told, I prefer classic movies from the '40s and '50s, but my tastes range all over the map from cheesy horror films to the latest avant-garde foreign cinema.

When I realized that the deadline for this column was rushing up, I did what any self-respecting geek would do: I got sidetracked with something else. In this case, the something else was the surprisingly nuanced and interesting *The Number 23*, starring Jim Carrey and directed by Joel Schumacher.

In the movie, Carrey is obsessed with numerology and how so many of the things in his life add up to the number 23. He's "haunted by the number" and ultimately "attacked by the number" as the movie progresses through its twists and turns.

What I found interesting was the method by which he found 23 to be such a pervasive number, ranging from the character's birthday (February 3) to the time on a clock

Could it be that this very magazine is infused with that evil number?

(2:15 is 2/3 if you look at an analog clock face). Numerology is all about the ordinal value of letters though, where A is 1, B is 2, and so on. So much of the movie is about how words and names add up to 23 too.

Ah, I thought, could I write a shell script that would do basic numerology? Could it be that this very magazine is infused with that evil number? Let's find out!

Breaking Words into Characters

The first step in writing a basic numerology script is to learn how to break down a word or phrase into its component parts, scrubbing it of all punctuation and white space. We also want to convert all uppercase to lowercase, or vice versa, as A and a should have the same numeric value (1).

This can be done with a single line in a script, thanks to the ever-powerful `tr` command:

```
tr '[A-Z]' '[a-z]' | tr -Cd '[:alnum:]'
```

The first call to `tr` converts uppercase to lowercase, as required (though to be completely portable, I really should have written it as `'[:upper:]' '[:lower:]'`, but I wanted to have both common idioms demonstrated here for your reading pleasure).

The second call to `tr` is a bit more tricky: the `-d` option instructs the program to delete characters in the input stream that match the specified set, and `-C` reverses the logic of the match. By using `[:alnum:]`, I pull out only the letters and digits, stripping everything else.

Let's see this snippet at work:

```
$ echo "This Is A - 12,3 - Test" | \
tr '[A-Z]' '[a-z]' | tr -Cd '[:alnum:]'
thisisa123test
```

And, that's neatly and easily done. Now, the tougher part—how do you step through a word, letter by letter, in a shell script? That's a job for the `cut` command!

I'm going to use a stepping integer variable to make life easier too, called `ptr` (here's an example of where a Perl or C for loop with all its power is sorely missed):

```
ptr=1

while [ some condition ] ; do

letter="$( echo $cleanword | cut -c $ptr )"

ptr=$(( $ptr + 1 ))

done
```

The question is what condition should we be testing so that it'll get every character in the string, but nothing else? According to the `cut` man page, the program will produce a nonzero return code on failure, and it certainly seems to me that an invocation like this:

```
echo 123 | cut -c4
```

should be an error, because there is no fourth character, but experimentation demonstrates that it isn't the case. Here's how I tested it:


```
#!/bin/sh

echo 123 | cut -c4

if [ $? -ne 0 ] ; then

echo error condition

else

echo no error condition

fi
```

Alas, the result is "no error condition". On the positive side, cut does return a null string correctly, so we can test for that. But, because we're doing maximum paranoia coding, it's useful also to have the length of the word or phrase. After all, what if it's 23 characters long?

Given that the length is already computed (with a quick call to wc -c), the conditional simply can be to test ptr against the string length, calculated after the string is cleaned up. In other words, while [\$ptr -lt \$basislength].

Calculating Letter Value

The hardest part of this script unquestionably is mapping letters to numeric values. Perl, C, Awk and just about every scripting language has a solution, but within the shell itself? There's nothing I can imagine without extraordinary levels of effort.

Fortunately, there's a 15-character Perl solution that lets us write a command suitable for dropping into a command pipe:

```
perl -e '$a=getc(); print ord($a)-96'
```

Thus, we have a tool to calculate the ordinal value without too much difficulty, now that we know how to extract individual letters:

```
ordvalue="$(echo $letter | \
perl -e '$a=getc(); print ord($a)-96' )"
```

Let's put it all together and see where we are:

```
#!/bin/sh
```

Rugged Mobile Linux Ready for Everything

Introducing the TDS Nomad® 800 Series

The TDS Nomad begins where other handhelds end. With integrated GPS, 802.11, camera and bar code scanner, the Nomad packs in plenty of features.

Standard Features:

- Ultra-Rugged Design (IP67, MIL-STD-810F testing)
- Linux® 2.6 with Qtopia® or X11/GTK+
- Marvell® PXA320 at 806 MHz
- 128 MB RAM
- 1 GB NAND Flash
- Integrated Bluetooth®, 802.11g, GPS
- Sunlight-Visible VGA Screen (640x480)
- CompactFlash®, SDIO® Slots
- Available in Yellow, Olive Drab, and Gray

Optional Features:

- Integrated 2 Megapixel Camera
- Integrated Bar Code Scanning



TDS Recon Also Available

SDG
systems

Equipping the Mobile User
1.724.452.9366 www.sdgsystems.com/rugged

```
# Given a word or phrase, figure out its numeric equivalents

ptr=1

if [ -z "$1" ] ; then

    echo -n "Word or phrase: "

    read basis

else

    basis="$@"

fi

basis="$( echo $basis | \

    tr '[A-Z]' '[a-z]' | \

    tr -Cd '[:alnum:]' )"

basislength="$( echo $basis | wc -c )"
```

```
echo "(Working with $basis which has \

    $basislength characters)"

while [ $ptr -lt $basislength ] ; do

    letter="$( echo $basis | cut -c $ptr )"

    ordvalue="$(echo $letter | \

        perl -e '$a=getc(); print ord($a)-96' )"

    echo "... letter $letter has value $ordvalue"

    ptr="$(( $ptr + 1 ))"

done

exit 0
```

The conditional at the top lets this script be maximally flexible. If you specify a word or phrase when you invoke the script, it'll use that. If you forget, it'll prompt you to enter a word or phrase. Either way, that ends up as `basis`, which is then successively cleaned up to remove unwanted letters. `basislength` is the length of the resultant string, which is stepped through, letter by letter, in the while loop.

A quick test:

```
$ sh numerology.sh

Word or phrase: linux

(Working with linux which has 6 characters)

... letter l has value 12

... letter i has value 9

... letter n has value 14

... letter u has value 21

... letter x has value 24
```

Great. We have the basis of a numerology calculator with all the difficult work taken care of. All that's left is to do some summary values and push around possible combinations to see if we can ascertain whether that pesky 23 does indeed show up everywhere!

Acknowledgement

Thanks to Dave Sifry for his help with that spiffy little Perl code snippet. ■

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

COMPACT EMBEDDED SERVER



- x86 200MHz CPU
- 128MB SDRAM On Board
- 512MB CompactFlash™
- 10/100 Base-T Ethernet
- Reliable (No CPU Fan or Disk Drive)
- Two RS-232 & Two USB 1.1 Ports
- Optional Wireless LAN & Hard Drive
- Optional On Board Audio
- Dimensions: 4.5 x 4.5 x 1.375" (115 x 115 x 35mm)



2.6 KERNEL

**Compact SIB
(Server-In-a-Box)
Starting at \$198.00
Quantity 1.**

- EMAC Linux 2.6 Kernel
- Menu Drive Configuration Utility
- Eclipse Development Environment
- HTTP and FTP Servers
- PPP Dial In/Out Server & Client
- Telnet Server

Since 1985
OVER
22
YEARS OF
SINGLE BOARD
SOLUTIONS

EMAC, inc.

EQUIPMENT MONITOR AND CONTROL

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



February 11, 2008
8 AM - 5 PM

University of North Florida University Center
Jacksonville, Florida

<http://www.floridalinuxshow.com/>

Show Highlights

- "Green" (energy saving) PCs
- Computer Security
- Making the Move to IPv6
- Linux Careers
- Linux at Work and at Home
- Linux Certification by Linux Professional Institute (LPI)

The **Windows Petting Zoo** will feature Windows demonstrations, software, hardware, and local vendor advice.

The **Apple Orchard** will give Macintosh users a chance to see and try Apple's latest hardware and software innovations.

Official Media Sponsors



<http://www.linuxjournal.com/>



<http://www.thinkgeek.com/>



<http://www.linux.com/>



<http://slashdot.org/>



<http://www.itmanagersjournal.com/>



<http://web.sourceforge.com/>



<http://www.freshmeat.org/>



<http://sourceforge.net/>

SourceForge.net, Slashdot, freshmeat, and ThinkGeek are registered trademarks of SourceForge, Inc., in the United States and other countries.

Charitable Outreach



We believe it's important to support worthwhile charitable causes. This year, Florida Linux Show will donate \$1.00 of every \$10.00 to the Children's Miracle Network. The Children's Miracle Network of hospitals provides specialized care for sick and injured children, training for pediatricians and pediatric specialists, and education for families. Donation jars will also be available during the show for anyone who wishes to make an additional contribution.



Join us for the hottest Linux show in Florida this coming February!

Key Personalities

Hear from some of the most relevant names in the Linux community:

Donald L. "TheLinuxGuy®" Corbet, founder of D. L. Corbet & Associates, LLC, the 2008 show chairman.

Jon "Maddog" Hall, Executive Director of Linux International and well known Linux advocate, key presenter.

Robin "Roblimo" Miller, Editor in Chief of Open Source Technology Group, featured speaker.



KYLE RANKIN

Browse the Web without a Trace

Concerned about your privacy? It takes only a Knoppix disc and a few simple steps to browse the Web anonymously.

Is privacy dead? When I think about how much information my computer and my gadgets output about me on a daily basis, it might as well be. My cellphone broadcasts my general whereabouts, and my Web browser is worse—every site I visit knows I was there, what I looked at, what browser and OS I use, and if I have an account on the site, it could know much more.

Even if you aren't paranoid (yet), you might want to browse the Web anonymously for many reasons. For one, your information, almost all of it, has value, and you might like to have some control over who has that information and who doesn't. Maybe you just want to post a comment to a blog without the owner knowing who you are. You even could have more serious reasons, such as whistle-blowing, political speech or research about sensitive issues such as rape, abuse or personal illness.

Whatever reason you have for anonymity, a piece of software called Tor provides a secure, easy-to-setup and easy-to-use Web anonymizer. If you are curious about how exactly Tor works, you can visit the official site at tor.eff.org, but in a nutshell, Tor installs and runs on your local machine. Once combined with a Web proxy, all of your traffic passes through an encrypted tunnel between three different Tor servers before it reaches the remote server. All that the remote site will know about you is that you came from a Tor node.

Tor works well on its own for anonymity, but anyone who has access to your machine can see that you have it installed. In some situations, even possession of anonymizing software might implicate you if you work in a company or live in a country where it is frowned upon. However, even in these cases, if you have a Knoppix disc, you quickly and easily can set up an anonymous Web browsing environment that will disappear once you reboot your computer. Because Knoppix boots and runs completely from the disc, any changes you make to it are stored in RAM and are erased once you reboot the machine.

First, you need a Knoppix disc. If you have a fast Internet connection, download a CD or DVD image from the official Knoppix site and then burn it to disc. Otherwise, look on the official Knoppix page for links to retailers who will ship you a Knoppix disc for a fee.

Next, boot your Knoppix disc. Knoppix should attempt to get on the network automatically, but if it doesn't, click K→Knoppix→Network/Internet for network configuration options. Knoppix has a sophisticated system that allows you to write to all areas of the filesystem as though it were installed on a hard drive. Because of this, you actually can install Tor on Knoppix

according to the official directions on the Tor site. First, click K→Knoppix→Utilities→Manager software in Knoppix to start Knoppix's package manager. Then, click Reload to get the latest list of packages, search for the privoxy and tor packages, and select them for installation (or, if you want a shortcut, simply open a terminal and type `sudo apt-get update && sudo apt-get install tor privoxy`).

Now Tor will be set up and running, but Privoxy still needs a bit of extra configuration to use Tor. Open `/etc/privoxy/config` in a text editor, and add this line to the top of the file (don't forget the trailing dot):

```
forward-socks4a / 127.0.0.1:9050 .
```

After that, find any lines that look like the following and comment them out with a #:

```
logfile logfile
jarfile jarfile
```

Finally, open a terminal and type:

```
sudo /etc/init.d/privoxy restart
```

And, Privoxy will be ready to use.

Although you could configure Iceweasel (Firefox's name on Knoppix) by hand to use Privoxy, there is a nice plugin created just for Tor. Open Iceweasel and go to <https://addons.mozilla.org/firefox/2275> to install the Torbutton plugin. Once you install the plugin and restart Iceweasel, a button at the bottom right of Iceweasel will appear that says either "Tor Disabled" or "Tor Enabled". Simply click the button to toggle the state, and the plugin will take care of the rest.

From here, you can browse the Web anonymously. If you have never used Tor before, it's worth noting that you might see a slowdown in performance, as your traffic does need to be encrypted and pass through three extra servers. Also, certain sites, such as Google, may appear in their German or Japanese versions, depending on which Tor node you exit through. Once you are finished, shut down the machine, and all traces of Tor, Privoxy and your browsing history will be erased. ■

Resources

Official Tor Site: tor.eff.org

Official Knoppix Site: www.knoppix.org

Kyle Rankin is a Senior Systems Administrator in the San Francisco Bay Area and the author of a number of books, including *Knoppix Hacks* and *Ubuntu Hacks* for O'Reilly Media. He is currently the president of the North Bay Linux Users' Group.

8th Annual

2008 Web Services/SOA on Wall Street

Show and Conference

Web Services, Service Oriented Architecture, SaaS and Web 2.0

February 11, 2008, Mon Roosevelt Hotel, NYC

Madison Ave and 45th St, next to Grand Central Station

"An intimate show and conference – I could talk to the speakers and see the exhibits without the usual big-show hassle."

Get on board for High Performance SOA and Web Services, SaaS and Web 2.0 for Wall Street.

Case studies for Service Oriented Architecture by leading Wall Street players.

2008 Web Services/SOA on Wall Street expands to include Web 2.0 for the Wall Street enterprise and Software as a Service (SaaS) offerings for the financial markets. Underlying technologies including ESBs, Java, .Net, XML and AJAX are also covered.

Wall Street IT directors are under the gun to reduce Total Cost of Ownership while increasing Agility. Our event will offer a unique, person-to-person networking opportunity for Wall Street IT executives and project directors.

The 8th Annual will bring together major Wall Street speakers together with leading vendor experts. See who is changing the way the financial markets employ new innovations while maintaining 24/7 operations.

This is your opportunity to attend this Live Search Engine covering:

- Web Services and Integration Technologies
- High Performance Service Oriented Architectures
- SOA Management, Security and Performance
- Software as a Service (SaaS) Delivery of Financial Applications
- SaaS Infrastructure, Virtualization and On Demand Technologies
- Web 2.0 Collaboration suites - From RSS to Wikis
- Rich User Interface delivery - AJAX, Flex, JavaFX
- Web 2.0 and SOA Mashups - A New Middleware for Wall Street
- XML, Java and .Net infrastructure

Show: Mon, Feb 11 8–4:00
Conference: 9–4:50

The Show is free, but you must register in advance. Full conference only \$295. Save \$100. \$395. on site. Register today.



2007 Sponsors



Microsoft®

webMethods®



2008 Media Sponsor



For more information on the event, to be a sponsor or an exhibitor, contact:

Flagg Management Inc
353 Lexington Ave, NY, NY 10016
(212) 286 0333 flaggmgmt@msn.com

Register online: www.webservicesonwallstreet.com

Wolfram Research's Interactive Mathematica

The Linux-friendly Wolfram Research has taken a page from Adobe Acrobat's playbook by creating the new and free Mathematica Player runtime application, which is available now for download on Linux, Windows and Mac OS platforms. People with a licensed copy of Mathematica 6 can upload their Mathematica notebook files for processing to the new Publish for Player Web service, after which the notebook files will run in the Player. The end result is that you do not need a full version of Mathematica just to view documents as in the past. Also in the pipeline is the non-gratis Mathematica Player Pro for viewing interactive Mathematica documents and other functionality.

www.wolfram.com/player



Nuxeo's Enterprise Platform

The world of enterprise content management (ECM) has gotten more interesting with veteran Nuxeo's recent release of its Enterprise Platform 5.1 ECM application. Nuxeo says that this release is "distinguished by its service-oriented architecture (SOA), scalability and flexibility." The firm touts its infrastructure being built on plugins and extension points that are based on the OSGi standard, giving developers and integrators the ability to create custom configurations and extensions quickly and easily. New features in 5.1 include an advanced search service based on the NXQL query language (SQL-based), data import/export service, enhanced horizontal scalability and electronic and physical records management.

www.nuxeo.com

Grameen Foundation's Mifos

Open source is being used to battle against global poverty. One excellent example is the Grameen Foundation's Mifos, an application for nonprofits to manage microfinance operations efficiently. Microfinance is a form of economic development whereby poor people, typically in developing countries, receive small loans to start small enterprises and get out of poverty. You may recall that Grameen's director, Dr Muhammad Yunus, won the 2006 Nobel Peace Prize for his microfinance work in Bangladesh. Although Grameen created Mifos, it has generously made the software available to everyone and is leveraging the open-source model industry-wide. Although Mifos has been around for a year, the latest news is that IBM will apply its expertise in finance and open source to improve the application.

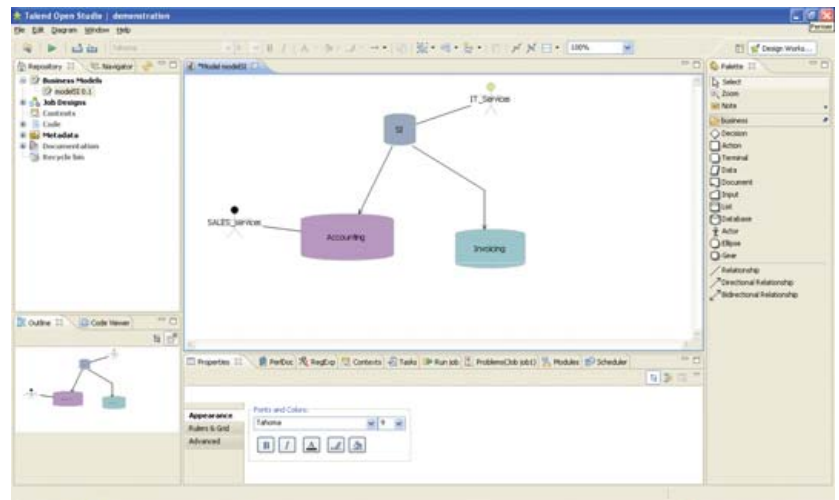
www.mifos.org

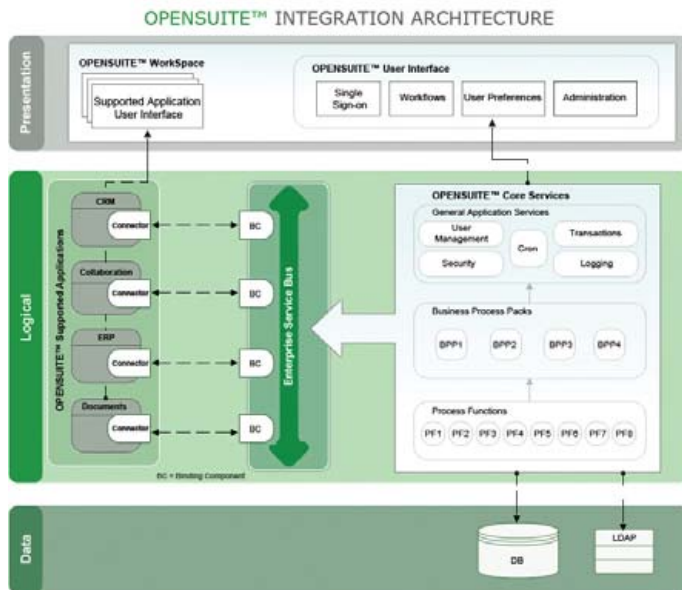


Talend's Open Studio

Talend has two new developments this month: its Open Studio open-source data-integration application was upgraded to version 2.2; and it released the Activity Monitoring Console/Personal Edition. First, Open Studio Version 2.2, which has more than 150 connectors available, now offers a number of new specialized connectors, as well as event-based action triggering and SOA functionality that enables exposure of data-integration processes as Web services. Furthermore, Open Studio takes advantage of recent improvements in Eclipse v3.3. Second, Monitoring Console/Personal Edition is a new centralized tool for monitoring the distributed execution of all data-integration jobs. It provides notifications upon failure or error as well as the ability to analyze statistics and trends and detect potential execution bottlenecks before they occur.

www.talend.com





CorraTech's OPENSUITE

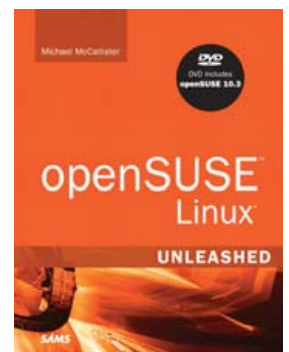
In other integration news, CorraTech announced OPENSUITE, a Java-based, open-source application that will enable business process and data integration across a range of open-source applications. OPENSUITE is currently in pre-beta. The aim is to integrate CRM, ERP, content/document management, messaging and project management. Organizations can implement cross-application business processes, preserve intrasession context while working with multiple applications, create single sign-on access for multiple applications and reduce redundancy introduced by the complexity of integration across applications. Using an SOA approach and supplying middleware layer functionality, OPENSUITE is distributed with a number of packaged business processes, called Business Process Packs. The first Pack will support CentricCRM, KnowledgeTree, Openbravo and Zimbra.

www.opensuite.com

openSUSE Linux Unleashed by Michael McCallister (Sams Publishing)

Despite the crush of Linux information out in Internetlandia, having an organized, distro-specific book on hand when trouble or confusion strikes is sanity insurance. The upgraded *openSUSE Unleashed* by Michael McCallister and Sams Publishing is the latest in the wide-ranging Unleashed series of comprehensive technology reference guides. Based on openSUSE 10.3, *Unleashed* covers just about everything you'd like to do with your OS, from installing and administering to working with standard desktop applications and setting up networks and servers. The companion DVD includes openSUSE 10.3 with five CDs worth of goodies, and on-line updates are available. *Unleashed* is recommended for intermediate to advanced users.

www.sampublishing.com



Wrox's Professional SlickEdit and Beginning Linux Programming

Wrox had a couple of particularly interesting November releases, such as *Professional SlickEdit* by John Hurst and *Beginning Linux Programming*, 4th edition, by Neil Matthew and Richard Stones. *Professional SlickEdit*, likely the first guide dedicated to the SlickEdit tools, is an example-heavy, hands-on guide to getting the most out of this popular development environment. The CD-ROM offers an exclusive extended trial version of SlickEdit. Meanwhile, in its 4th debut, *Beginning Linux Programming* takes a similar, learn-by-doing approach to teaching UNIX programming and application development in C on the Linux platform. The book also introduces toolkits and libraries for working with UIs of all sorts. Advanced topics include processes socket programming, MySQL, writing apps for GNOME/KDE desktop, writing device drivers, POSIX threads and kernel programming. Wrox also offers *Professional Linux Programming*, a recent book for more-advanced developers.

www.wrox.com

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

Pure64

Power, speed,
and performance in one

www.infi-tech.com/pure64



Experience pure 64-bit computing.
Workstation with pure performance, pure speed.

INFITECH

1-800-560-6550

For maximum speed and performance, we recommend PureOS

Linux reborn in pure 64-bit form -
free from legacy,
free to fly.

Let the new age of
Linux performance begin.

www.pure-linux.com

The One

PureOS

For optimized performance, PureOS recommends Infitech's Pure64 system.



THE TAO OF LINUX SECURITY

the Five Phases of a Secure Deployment

A tao, meaning “the path” or “the way”, is a system of guidelines or rules meant to achieve a desired end. Like any tao, security requires a structured, systematic approach. It also should be holistic, encompassing every part of a system’s life span from planning to retirement. The tao in this article consists of five simple phases that every Linux system should pass through to establish a secure baseline. The phases are not a be-all, end-all formula for security. They merely provide a foundation on which to build deeper levels of security later.

As I go through each phase, I will deploy two sample systems, a Red Hat Fedora desktop and a Debian server system, to show how security is implemented. I chose Fedora because it is arguably the most popular distribution out there. It works well for any purpose, has many desktop enhancements and is one of the easiest distributions for non-command-line Linux users to pick up. I chose Debian as our server platform because it is lightweight, has a long history of reliability, a great support community and extensive documentation. Both are excellent platforms and have their own unique security measures built in. If you have a different distribution preference, the phases presented here can be applied to any system.

Jeramiah Bowling

**START ON "THE PATH"
TO A MORE SECURE SYSTEM.**



Phase 1: Planning

The first phase of the tao is also the most important, as it is where you make most of the decisions that affect your overall security. The first step before all others is to define the purpose of the system being deployed. Will it be a mail server? A desktop? An intrusion-detection system? Once you have a purpose, you can use it as your guide throughout this process. The focus then becomes fulfilling that purpose while providing the most secure environment possible. After all, why deploy a system that no one can use? Security never should handicap functionality.

Next, you need to decide what your security goals are for the system. The primary goal of every build always should be the principle of least access or least privilege. This means providing only the minimum permissions to users and programs necessary for the system to operate. You may have other security goals, such as scanning every file with antivirus software or authenticating every user with LDAP, but least privilege always should be paramount.

Armed with these goals, you need to figure out how to accomplish them before actually building the system. Answering some simple questions will help determine the appropriate steps to meet the goals in phases 2 and 4. Will this system be server or a desktop? The answer to this question will dictate a great deal of the configuration on your new system. Will users access the system locally or remotely? This is another important configuration question with security implications. Will the system need a desktop environment?

If you or the people who will be administering the system are comfortable with the command line, deploy your system as "headless", or without a GUI interface. By eliminating the X Window System from your install, you shrink your attack surface (your exposed area) dramatically. On the other hand, if you or your staff needs a GUI, install only one and understand that additional steps are required to lock down your system properly. It is a good rule of thumb to install a server without X and use a desktop environment for user desktop systems.

Finally, plan the applications that will operate on this system. Determine which dependencies and libraries are essential to operation. Too often, unnecessary libraries are used after a system has been compromised to run remote commands, mask an intruder's presence or probe networks. If you do not need a package do not install it.

When finished working out the answers to these questions, write them down in a build log or notebook and keep them up to date even after the initial build.

Phase 2: Building

Armed with written plans, now comes the building phase. Here is where you begin the technical application of the security goals you set in the planning phase. Due to space constraints, the following section is not a detailed checklist of each of the example installs. I have highlighted only those options that are relevant to the focus on security. As in phase 1, write down your selections during installation in a build log in case you ever need to rebuild the system.

Fedora Core 7 Start by booting a fresh system from a Fedora 7 ISO available from any of the distribution's site mirrors. After entering your keyboard and language settings, you come to the disk partitioning utility. For most desktops, the installer does a fine job of configuring the drives appropriately. But, if the system will be used by someone whose work is sensitive, place the /home folder on its own partition.

After partitioning comes the bootloader options. Select GRUB and set a password. Using a password with your bootloader is good practice

and helps prevent data loss if your drive or system ever is stolen. Like any good password, it should have complexity, avoid dictionary words and use numbers, letters and non-alphanumeric characters.

On the next screen, select DHCP, as client machines usually don't require static IPs. If you require one, make sure to use Network Address Translation (NAT) somewhere on your network. Next, set a hostname and domain, and set a complex root password. At the package selection screen, always review what you are installing by using the Customize Now radio button. On the Custom Selection screen, select only one desktop environment. Installing more than one environment leads to more vulnerabilities (and patches to update) later. Leave the default option on GNOME. Go through each of the other checked options, and you will discover a lot of packages will be installed (Figure 1). On my build, I have 843 packages using these options. Your number may vary. Use the Optional Packages button to eliminate any extraneous packages you do not need. When you're finished with package selection, the system will reboot.

Upon reboot, you are prompted to enable the firewall. Enable it, and add only the ports/programs you need to operate this system (Figure 2). At the Security Enhanced Linux screen, if you are able to, set the policy to enforcing (Figure 3). To those unfamiliar with SELinux, it's a policy-based protection scheme, originally developed by the



Figure 1. Install Package Selections in Fedora 7



Figure 2. Fedora Firewall Configuration



Figure 3. SELinux dramatically enhances security on your machine.

National Security Agency, that adds a layer of security on top of the OS and several popular Linux applications, such as Apache. The downside to using SELinux is it sometimes can break the programs you are trying to protect (and other programs too). If you run into problems, you always can change the policy later, using the `setenforce` command or by editing the policy with a text editor. You then will be prompted to create a user account, and after a few other selections, the first login screen comes up.

Debian Start your blank server up with a Debian ISO (4.0r1 used here). Because I'm using this system as a server for this example, let's take a more Spartan approach toward the install. Servers usually are a juicier target than desktops for nefarious people, so you need to take extra precaution, especially if they will be deployed on a public network or the Internet.

After setting the usual options for time zone, country and so forth, you need to set a hostname and domain for your server. Next comes the partitioning options. Rather than putting many of the system directories on their own partitions, stick with putting the `/home` directory on a separate partition, and go one step further by encrypting your drive. This may sound difficult, but the Debian partitioning utility makes this easy. Select Guided partitioning from the options, and then select Guided - Use entire disk and set up encrypted LVM. Use the entire disk and select the option to use a separate home partition. Debian recommends further segregating your directories to keep the root, programs and user data separate, but configurations like this can be difficult to administer. Write the changes to disk and, when prompted (Figure 4), enter a complex passphrase to encrypt the volume. Next, set your time zone and root password, and then the base install will begin.

After the base install, the next few prompts are inconsequential until you get to the software selection screen. For our scenario, install an Apache Web server only. Finally, install the GRUB loader, and your new system is ready to go. Upon reboot, lock down the GRUB loader with a password, by adding the following lines to the `/boot/grub/menu.lst` file:

```
timeout 30
password yourpasswordhere
```

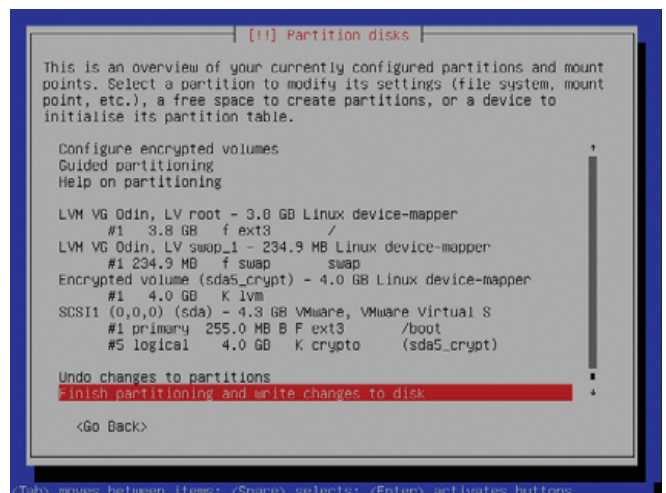


Figure 4. Debian makes complex partitioning schemes easy to configure.

Phase 3: Patching

The next phase in this deployment is to patch the systems. Although sometimes tedious, patching is a necessary evil. After all, 843 packages are a lot to protect. You also need to keep your systems updated to mediate the threat of new vulnerabilities. Thankfully, the example distros covered in this article make the process very easy. On the Fedora client, you already have this ability. Upon the first login, the system checks for updates automatically (Figure 5). Fedora uses the Yellow Dog Update Manager, better known as yum, and a new update GUI, pup, to automate the update process. However, the pop-up style reminder seems to work only in the GNOME desktop environment. If you want to update your system manually, you also can use the commands `yum list updates` or `yum info updates` to see which packages need an update. You also simply can run `yum` without any options to apply all available updates to all installed packages.

Moving on to the example server, Debian uses a utility called aptitude (apt) for updating packages. apt traditionally is used as a package manager, like Red Hat's RPM, but it also has the ability to check for updates like yum. It uses predefined and custom source lists to check for updates against your installed packages. If you do not have the following line in your `/etc/apt/sources.list` file, add it so you can check for

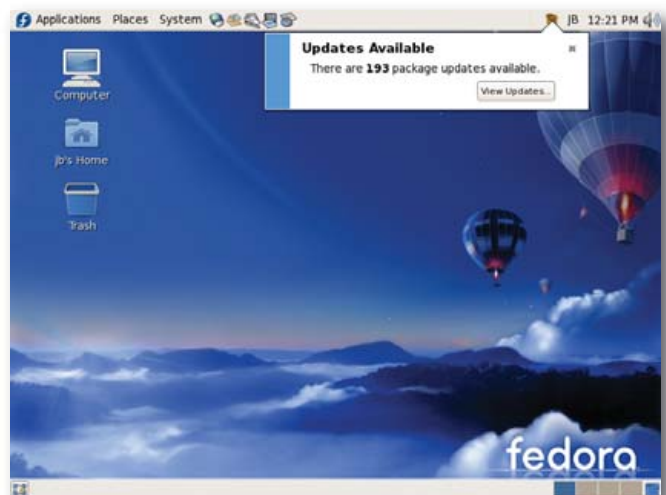


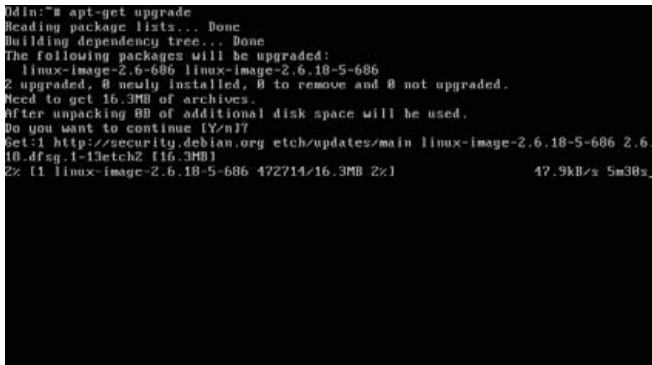
Figure 5. Fedora checks for updates immediately after install.

updates against the main stable US archive:

```
deb http://http.us.debian.org/debian stable main contrib non-free
```

Run the `apt-get update` command after adding the line. To update all the packages on your system, type `apt-get upgrade` (Figure 6) from a command prompt, and the system will begin checking and, with your approval, downloading and applying the updates. If you want to see what packages you have before running `apt-get`, use the command `dpkg --get-updates`. To check for updates once a week, use the commands below or write your own script and use `crontab` to schedule it:

```
echo /usr/bin/apt-get upgrade > /etc/check4updates
chmod 750 /etc/check4updates
crontab --e
```



```
admin:~# apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
The following packages will be upgraded:
  linux-image-2.6.686 linux-image-2.6.18-5-686
2 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 16.3MB of archives.
After unpacking 88 of additional disk space will be used.
Do you want to continue [Y/n]?
get:1 http://security.debian.org etch/updates/main linux-image-2.6.18-5-686 2.6.
18.dfsg.1-1etch2 [16.3MB]
2% [1 linux-image-2.6.18-5-686 472714/16.3MB 2%] 47.9kB/s 5m38s_
```

Figure 6. Use `apt-get` to look for Debian security updates.

Add the following lines to your `crontab` file to make the script run every Wednesday at 3:30am:

```
30 3 * * 3 /etc/check4updates
```

Beware—if you compile your own packages or use packages from another source, they may not be updated automatically using `yum` or `apt-get`.

Phase 4: Hardening

After patching your new system, you may need to take additional steps to secure it. This is where having your security goals noted in the planning phase helps. You can use these goals to determine what additional steps are appropriate, as you more than likely will have more steps than the few illustrated here. The more goals you have, the more steps you are likely to have as well. Try to keep simplicity in mind. Complex settings actually can make a system less secure, because they often can lead to misconfiguration. Also, remember to note these steps in your build log.

The Fedora example already has shown two important steps to enhance security: enabling SELinux and installing a firewall. In most typical desktop-use scenarios, when combined with an antivirus application, this is enough. For the Debian box, I have selected three common steps that should be used on any server system: using `sudo`, locking down SSH and using a restrictive `iptables` firewall. These simple items should be considered the bare minimum on any server system, and if desired, they can be applied to a desktop as well.

sudo

`sudo` is a great application for limiting root access, which should be guarded closely on any server. Adding users to the `/etc/sudoers` file, limits their ability to use `su` to specific commands, specific directories or by network host. Any users in the `sudoers` file simply need to type `sudo` before the commands they want to execute to run under root credentials. This is much easier and much safer than giving the root password to everyone.

SSH

SSH is the standard remote access protocol in use on Linux systems today. In its default configuration, it has some settings that you definitely need to lock down. Add the following lines to `/etc/ssh/sshd.config`:

```
PermitRootLogin no
X11DisplayForwarding no
```

The first line prevents root from logging in to the server via SSH, which never should be done. The second line disables X forwarding, which would allow users to launch an X session from your server. In the example case, X isn't installed, so this should not be a problem. You could lock down SSH further by chrooting it or using TCP Wrappers; however, due to space constraints, I have omitted those configuration steps.

iptables Firewall

Rather than go into a long discussion on the proper configuration of a firewall, I have created the following script with comments to secure the Debian system. It restricts traffic (statefully) only to new SSH, HTTP and SSL connections. Change the IP address in this example to your server's address. For more details on the options available in `iptables`, consult the man page. When building your own firewalls, keep in mind the goal of shrinking the attack surface by opening only necessary ports in `iptables`:

```
#!/bin/sh

PATH=/usr/sbin:/sbin:/bin:/usr/bin

#FLUSH PREVIOUS TABLE ENTRIES
iptables --flush

#CHANGE DEFAULT POLICIES FROM
#ACCEPT TO DROP
iptables -P INPUT DROP
iptables -P FORWARD DROP
iptables -P OUTPUT DROP

#ALLOW LOCAL LOOPBACK TRAFFIC
iptables -A INPUT -i lo -j ACCEPT

#ALLOW ESTABLISHED CONNECTIONS
iptables -A INPUT -m state --state \
ESTABLISHED,RELATED -j ACCEPT
iptables -A OUTPUT -m state --state \
ESTABLISHED,RELATED -j ACCEPT

#ALLOW DEFINED TRAFFIC
#
```

Expert Included.



Figure 7. Bastille in Debian

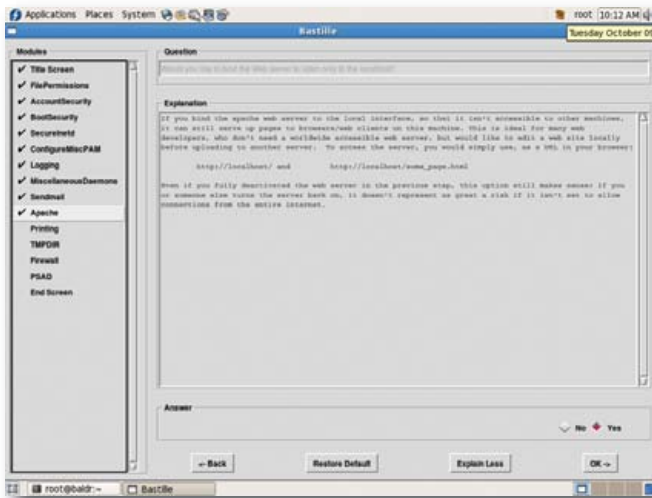


Figure 8. Bastille in an X Window System Fedora Environment

```
#SSH - 22
iptables -A INPUT -d 192.168.1.2 -p tcp \
--dport 22 --sport 1024:65535 -m state \
--state NEW -j ACCEPT

#HTTP - APACHE -80
iptables -A INPUT -d 192.168.1.2 -p tcp \
--dport 80 --sport 1024:65535 -m state \
--state NEW -j ACCEPT

#SSL - 443
iptables -A INPUT -d 192.168.1.2 -p tcp \
--dport 443 --sport 1024:65535 -m state \
--state NEW -j ACCEPT
```

Save this script locally, and copy or move it to the /etc/network/if-up.d directory so that it runs when the network comes up after boot. If you want to apply this configuration on a Red Hat-based system, you simply could run the above script and use the iptables-save command to keep the rule set across reboots.

Although you could take these steps and many more, there is a tool that makes this process much easier, Bastille (Figures 7 and 8). Bastille uses question/answer responses to script your preferred security settings and apply them to the actual system. There also are a multitude of manual security checklists available for most distributions and

As president of Silicon Mechanics, Bob stays on top of advances in the technology industry. He makes it his business to ensure that the technology experts of Silicon Mechanics are positioned to provide best-fit solutions for every customer. That's why Bob is all over AMD's recent innovation, the Quad-Core Opteron™ processor. The Quad-Core Opteron processor is an industry-defining native quad-core microprocessor that continues AMD's tradition of energy efficiency, performance and virtualization leadership. Silicon Mechanics is proud to offer the AMD Quad-Core Opteron processor with many of its rackmount servers, including the ones pictured here: the Rackform nServ A236 (1U), A266 (2U), A276 (3U), and K501 (5U).

When you partner with Silicon Mechanics, you get more than a finely tuned AMD solution—you get Bob and a team of technology experts.



SILICON MECHANICS

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

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



Bastille uses question/answer responses to script your preferred security settings and apply them to the actual system.



applications that can be found on the Internet. Some of the best checklists are the benchmarks put out by the Center for Internet Security. These benchmarks contain detailed settings and descriptions of “best practices” relating to specific operating systems and popular applications. They are excellent companions to Bastille.

Phase 5: Monitoring/Auditing

The last phase of the tao is an ongoing process. Regular monitoring of your system will verify that your security goals are being met over time. The most useful tool for this is built right in to the system,

syslog. From the /var/log/messages file, you can view a variety of security-related information for both the system and some applications. Many applications use their own log files. Be sure to look through those as well. If you have multiple systems, you should use a central syslog server to collect the logs. This easily can be configured in the syslog.conf file.

A newer alternative to syslog is called Splunk (Figure 9). Splunk has both free (limited to 500MB daily) and enterprise versions. The nice thing about Splunk is its super-easy install, and you can search through logs using Google-like commands through a streamlined Web-based interface.

As useful as logs are, they do not provide a complete picture of how well your security is working. Only regular auditing can accomplish this. Doing so tells you if your security is still in place and functioning. I am not suggesting penetration testing for every system, but active testing of your settings is good insurance. Create checklists or scripts to test those settings that are important to maintaining your security goals. In lieu of a checklist, you could run Bastille using the --assess switch to get a security report of your current configuration. You also can use the CIS benchmarks (which rely on Bastille) as baseline checklists for an audit. If you can afford it, have an outside consultant come in and verify your security with his or her own tests to give you peace of mind, especially if you work in a heavily regulated industry.

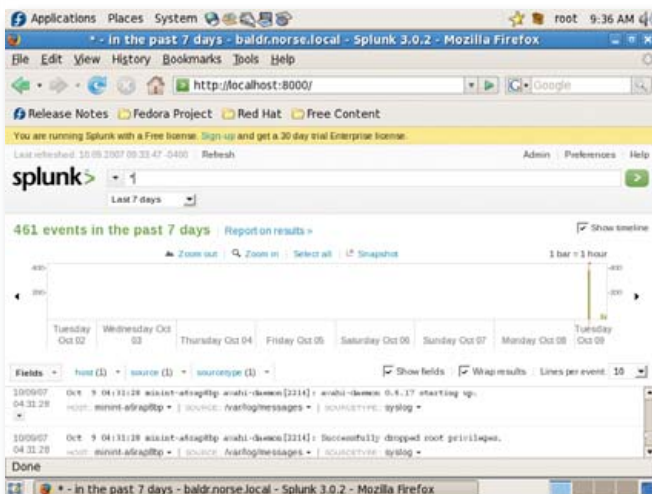


Figure 9. Splunk is one of the best and most useful open-source projects available.

On the Path

Using the phases of the tao in your builds is the first step. This simple, ordered strategy will get you to a more secure place with your systems; however, each system is unique. Make sure your security goals match your system. Security is not difficult. Use simple and repeatable processes, stay informed of best practices and common vulnerabilities, exercise least privilege wherever possible, regularly review your logs, and you will find yourself on the Path. ■

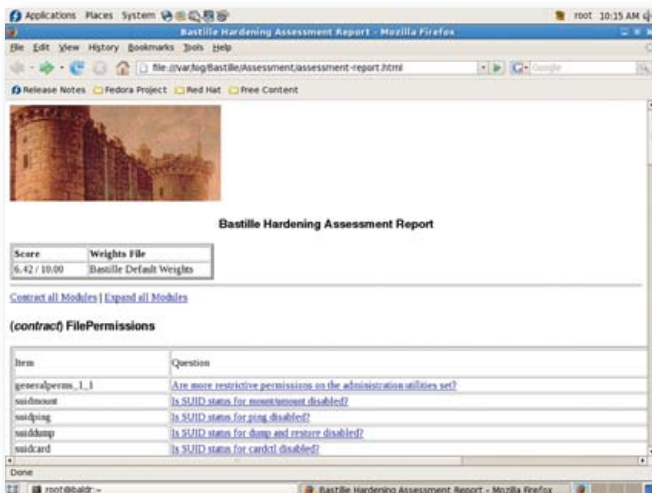


Figure 10. The Bastille assessment report gives you a detailed overview of your current security configuration.

Jeremiah Bowling has been a systems administrator and network engineer for more than ten years. He works for an accounting and auditing firm in Hunt Valley, Maryland, and holds numerous industry certifications, including CISSP and Linux+. Your comments are welcome at jb50c@yahoo.com.

Resources

Debian Security Manual: www.debian.org/doc/manuals/securing-debian-howto

Fedora 7 Security Wiki: fedoraproject.org/wiki/Security

Bastille-Linux: bastille-linux.sourceforge.net

Center for Internet Security: www.cisecurity.org

Splunk: www.splunk.com

Continuous Data Protection

The Future of Data Centers

Can your backup software do this?	R1Soft CDP Server	Acronis® True Image	EMC Retrospect®
Daily Backups	✓	✓	✓
Hourly Backups	✓	Not Supported	Not Supported
Open File Backups	✓	✓	Not Supported
Bare-Metal Restore	✓	✓	Not Supported
Continuous Data Protection	✓	Not Supported	Not Supported
Restore Linux LVM	✓	Not Supported	Not Supported
Restore Linux Software RAID	✓	Not Supported	Not Supported
Easy To Use Web Interface	✓	Not Supported	Not Supported
Manage Thousands of Servers	✓	Not Supported	Not Supported
Control Panel Integration	✓	Not Supported	Not Supported
	\$80 - \$100 /server	\$699 /server	You Can't Afford It

Data Centers serious about uptime and performance use R1Soft.

For more information visit: www.r1soft.com or call us at **800-956-6198**



R1Soft
 Continuous Data Protection
 For **LINUX & WINDOWS**

A shovel with a wooden handle and a metal blade is stuck upright in a large pile of dark brown soil. The shovel is the central focus, with its handle extending towards the top of the frame and its blade buried in the dirt. The background is plain white.

DIGGING UP DIRT IN THE DNS HIERARCHY, PART I

Explore hidden secrets of the DNS hierarchy with a benign and systematic diagnostic and audit methodology using readily available tools.

Our DNS is working. Our clients can access our Web site, our secure e-commerce service, our FTP servers and our LDAP services. Our people can e-mail and browse the Web, and our VoIP system is fully functional. In short, all the services that, without a functional DNS, would not be working, appear to be fully operational. All is well with our DNS—perhaps.

When researching for my first book, I was constantly surprised, and on more than one occasion stunned, at how often even large, technically proficient organizations had potentially dangerous flaws in their DNS infrastructure. This article describes a simple set of techniques using two readily available tools, `dig` and `fpdns` (see *Obtaining Tools* sidebar), to explore the DNS hierarchy methodically. The techniques used are benign; they essentially emulate the functions of a caching resolver and can be used both for diagnostic and auditing purposes. There are tools available that cover some of the techniques shown here, but it is vital that administrators understand the entire process to avoid any shortcomings in the tools.

Although this article uses the inevitable `example.com` and private IP addresses in the interest of being a good Netizen, I encourage you to pull up a shell and substitute your domain, one that interests you, is important to you or about which you are just plain curious. The diagnostic examples show BIND because it represents approximately 80% of the estimated nine million name servers.

RON AITCHISON

Scope the Target

To animate the techniques, let's simply look up the IP address(es) of the Web site or sites of our target domain example.com. First, using a browser, go to the site and navigate around, taking interesting links, especially the secure ones. Our objective is to build a list of all the host and domain names being used by our target. In our illustrative case, say we discover that our target uses www.example.com as its public portal and a second host, online.example.com, for secure (HTTPS) transactions. From here on, we simply emulate the behavior a caching DNS would use to find its way to the target Web sites with a few deviations thrown in to spice up the action.

The Root of All

The principal job of a caching name server is to resolve a domain name, such as www.example.com, into an IP address. If it has no information about the domain name in its cache, it must track the delegation route starting from the root servers through the DNS name hierarchy to the authoritative name server for the target domain. A delegation is defined by two or more Name Server (NS) Resource Records (RRs) in a parent zone. The NS RRs at the parent zone point to the authoritative name server of the child zone. This parent-child process is repeated for each level in the name hierarchy until we reach our target. Thus, for www.example.com, we will obtain a delegation from the root to .com's authoritative name servers, and then a delegation from .com to example.com's authoritative name servers.

A caching server gets its initial list of root servers from the root or hints zone, which uses a zone file typically called named.ca or named.root. To get the names of the root servers being used by your local DNS (defined in /etc/resolv.conf), simply enter: dig.

You should get a response that looks something like this:

```
>>> dig 9.4.1-P1 <<<>
;; -->HEADER<<- opcode: QUERY, status: NOERROR, id: 16298
;; flags: qr rd ra; QUERY: 1, ANSWER: 13, AUTHORITY: 0, ADDITIONAL: 7

;; QUESTION SECTION:
; . IN NS

;; ANSWER SECTION:
. 5058 IN NS A.ROOT-SERVERS.NET.
...
. 5058 IN NS M.ROOT-SERVERS.NET.

;; ADDITIONAL SECTION:
A.ROOT-SERVERS.NET. 798 IN A 198.41.0.4
...
M.ROOT-SERVERS.NET. 6957 IN A 202.12.27.33
```

```
;; Query time: 36 msec
;; SERVER: 192.168.2.1#53(192.168.2.1)
...
```

I've removed a lot of material above, indicated by the ... sequence in the interests of brevity.

The format of a dig response has five sections. The HEADER contains summary and status information, which we look at in more detail later. The next four sections contain information in standard Resource Record (RR) format as they may appear in a zone file. The QUESTION SECTION reflects the question or query received by the responding server. In the above case, the dig command was interpreted to be "get me the NS RRs for the root". The ANSWER SECTION may be empty if our question was not answered or may contain one or more RRs, which are the answer to our query. In the example above, it contains the NS RRs for the root servers (a.root-servers.net to m.root-servers.net). Note especially the infamous dot on the left hand side of each result line in this section, which is the short form for the root. The AUTHORITY SECTION normally contains one or more NS RRs for servers that are authoritative for the domain in question. In the above case, it is not present simply because the ANSWER SECTION already contains this information. The ADDITIONAL SECTION contains any information the responding server thinks may be useful and has available. In this example, and in most cases, it contains the A (Address) RRs of the authoritative name servers that our local name server has used.

The really interesting stuff is in the HEADER. The first thing to check is the status. In this case, NOERR means the command was successful (see the Dig Header Values sidebar for a complete list). The flags in this case are qr, indicating we received a query response that seems pretty reasonable; rd, indicating our dig message requested recursive services; and ra, signifying that this server supports recursive service (again, see the Dig Header Values sidebar for a complete list of possible flags). The HEADER also contains the id, which uniquely identifies this request/response pair and finally summarizes how many RRs we have in each section.

Obtaining Tools

Dig is one of the many utilities made available with a normal distribution of BIND, which may be obtained from www.isc.org in source form, and is widely available as a package for most Linux distributions and is in the ports system for BSD. It also can be installed on Windows 2000, XP and Server 2003 for those administrators who work in heterogeneous environments. For casual use on Windows, there is no need to install BIND fully; simply unpack the Windows distribution zip file and copy dig.exe, libbind9.dll, libdns.dll, libisc.dll, libiscfg.dll and liblwres.dll onto suitable portable media.

fpdns is a Perl script, developed by two of the smartest guys in the DNS world (Roy Arends and Jakob Shlyter), and it can be obtained from www.rfc.se/fpdns, the ports collection in FreeBSD and by using `get-apt install fpdns` for Debian/Ubuntu users.

FEATURE Digging Up Dirt in the DNS Hierarchy, Part I

The last few lines of the dig response yield useful performance information. The SERVER line particularly confirms the address and name of the server from which the results were obtained.

Now, our toolkit is in place, and we have some idea about what it is telling us. So, next, let's follow the DNS hierarchy from the root to our targets `www.example.com` and `online.example.com`.

The first command on our journey is:

```
dig @a.root-servers.net www.example.com
```

Dig Header Values

dig response HEADER values:

id: the 16-bit message ID supplied by the requester (the questioner) and reflected back unchanged by the responder (answerer). Identifies the transaction. Range 0 to 65535.

Flags may be one or more of the following values:

- AA (Authoritative Answer): set if the response was received from a zone master or slave.
- TC: (TrunCation): length greater than permitted, set on all truncated messages except the last one.
- RD (Recursion Desired): set in a query and copied into the response if recursion is supported.
- RA (Recursion Available): valid in a response and, if set, denotes recursive query support is available.
- AD (Authenticated Data), DNSSEC only: indicates that the data was reliably authenticated.
- CD (Checking Disabled), DNSSEC only: disables checking at the receiving server.

Status field response code:

- 0 = NOERR: no error.
- 1 = FORMERR: format error—the server was unable to interpret the query.
- 2 = SERVFAIL: name server problem or lack of information. Often also returned with the same meaning as REFUSED.
- 3= NXDOMAIN Name does not exist: meaningful only from an authoritative name server.
- 4 = NOTIMPL: not implemented.
- 5 = REFUSED: typically for policy reasons, for example, a zone transfer request.

This command uses one of the root servers (`a.root-servers.net`) to get the A RR of `www.example.com`. A purist would say that to avoid possible sources of corruption, we always should use `@x.x.x.x` (an IP address) rather than a name, especially in a diagnostic mode, and in general, this method is safer. However, names are more understandable (the reason for the DNS), and we always can check that the IP address on the SERVER line of the response is in the expected set. The dig response will look like this:

```
>>> DiG 9.4.1-P1 <<>> dig @a.root-servers.net www.example.com
...
;; -->HEADER<<-- opcode: QUERY, status: NOERROR, id: 15570
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 13, ADDITIONAL: 14
;; WARNING: recursion requested but not available
;; QUESTION SECTION:
www.example.com. IN A
;; AUTHORITY SECTION:
com. 172800 IN NS A.GTLD-SERVERS.NET.
....
com 172800 IN NS M.GTLD-SERVERS.NET.
;; ADDITIONAL SECTION:
A.GTLD-SERVERS.NET 172800 IN A 192.5.6.30
A.GTLD-SERVERS.NET. 172800 IN AAAA 2001:503:a83e::2:30
....
;; Query time: 38 msec
;; SERVER: 198.41.0.4#53(198.41.0.4)
...
```

Again, I've removed a lot of material indicated by the ... sequence in the interest of brevity. The first thing to note in the header is that the `ra` flag is not set, meaning recursion is not available—normal in the root and TLD servers. Second, the `aa` flag is not set, which means this is not an authoritative response. At first, this may seem strange; this is, after all, a root server. The root is the parent of `.com` (the next name in the hierarchy), but a parent's NS RRs (the point of delegation) are never authoritative; only the child can give an authoritative response for its NS RRs. This has important implications as we will see later. In summary, we have no answer (ANSWER 0) and no error (status NOERR), but there are AUTHORITY (13) entries. This identifies the response as a referral. The root cannot supply the answer but has

helpfully referred us to the next level in the hierarchy—in this case, the .com gTLD servers, whose names are given in the AUTHORITY SECTION and some IP addresses in the ADDITIONAL SECTION, including an IPv6 address, which is becoming increasingly common.

Continuing our journey, we issue:

```
dig @a.gtld-servers.net www.example.com
```

This command says, using one of the .com gTLD servers (a.gtld-servers.net), get me the IP address of www.example.com. This may begin to look a little tedious. After all, the root and gTLD servers are operated by professional organizations, so why bother checking them? Consider, however, that we may not be using the real root and gTLD servers. We obtained our list from our local caching server. If its hints zone is poisoned, this process will help us identify deviations from normality, and besides, some of the ccTLD zones have really interesting structures. The golden rule is make no assumptions. The dig response will look something like this:

```
>>> DiG 9.4.1-P1 <<<> @a.gtld-servers.net www.example.com
...
;; -->HEADER<<-- opcode: QUERY, status: NOERROR, id: 20018

;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 1

;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;www.example.com. IN A

;; AUTHORITY SECTION:
example.com. 86400 IN NS ns2.example.com.
example.com. 86400 IN NS ns1.example.net.

;; ADDITIONAL SECTION:
ns2.example.com 172800 IN A 10.10.0.2

;; Query time: 80 msec

;; SERVER: 192.228.28.9#53(192.228.28.9)
```

This response is almost identical to the previous one. It's a referral to the authoritative name servers for the domain example.com, which are ns2.example.com (in the same domain or zone) and ns1.example.net (not in the same domain)—termed an out-of-zone or an out-of-bailiwick name server. If we choose to use ns2.example.com, we can use the supplied IP address; this is a normal "glue" A record and is required to make delegation work. However, if we want to use ns1.example.net, we must obtain its IP address by restarting the process from the root servers through to the .net gTLD servers. We must

do this even if the IP address had been supplied, as it frequently is, in the ADDITIONAL SECTION of the response.

What's the reason for not trusting an out-of-zone IP address? We could pollute our authoritative name server space and allow a domain hijack. Let's illustrate the dangers. Suppose your local friendly caching name server comes to my unrelated domain, say www.example.org, and in this domain's zone file, I have the following zone records:

```
$ORIGIN example.org.
...
; NS RR for the domain -- perfectly correct

IN NS ns3.example.org.
IN NS ns1.example.net.
...
; normal name server A RR

ns3 IN A 192.168.2.4
...
```

Quick & Easy 2-Factor Authentication

We've taken the cost & complexity out of One-Time Passwords without sacrificing the security.

Now you can increase security with 2-Factor strong authentication today.

- Open Source
- Install and Configure in Minutes
- Runs on Linux, Solaris, or any Unix variant
- Integrates directly with FreeRADIUS
- Supports OATH (RFC 4226 HOTP) Standard

TRID Systems, Inc.
Simply Strong.

www.tri-dsystems.com

FEATURE Digging Up Dirt in the DNS Hierarchy, Part I

```
; out-of-zone A RR . potentially naughty
```

```
ns1.example.net. IN A 192.168.2.4 ;a malicious server
```

If the caching server trusts the A record for ns1.example.net—the out-of-zone server—and caches it without reading from its authoritative source, any subsequent resolution of example.com names using ns1.example.net would use a name server at 192.168.2.4, where a bogus zone file would likely redirect all example.com traffic to a malicious site. So, if we have an out-of-zone (or out-of-bailiwick) name server, any server that is not in the same domain name space as the target domain, its IP address must not be trusted until obtained from an authoritative source. This means tracking from the root. Modern versions of BIND discard all out-of-zone A RRs with an appropriate log message when the zone is loaded.

In our manual audit process, we could ignore ns1.example.net, but a real caching DNS will not. It will use it for approximately 50% of queries. In order to audit the route to our Web addresses thoroughly, we must check all the DNS servers that appear in the AUTHORITY SECTION. Any weak DNS may allow a compromise of some proportion of user traffic. The percentage of queries received by a name server should not be confused with the number of users. A single query from a large ISP may affect a disproportionate number of users, especially in regionally focused sites.

It can get a lot worse than this. Let's assume we have tracked ns1.example.net through the root servers and the gTLD servers for .net, and we get this response to our last dig:

```
; <<>> DiG 9.4.1-P1 <<>> @a.gtld-servers.net ns1.example.net
...
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 49319
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 2, ADDITIONAL: 2
;; WARNING: recursion requested but not available
;; QUESTION SECTION:
;ns1.example.net. IN A
;; AUTHORITY SECTION:
example.net. 172800 IN NS ns1.example.net.
example.net. 172800 IN NS ns4.example.org.
;; ADDITIONAL SECTION:
ns1.example.net. 172800 IN A 192.168.2.2
;; Query time: 61 msec
;; SERVER: 192.5.6.30#53(192.5.6.30)
...
```

We have another out-of-bailiwick name server (ns4.example.org), which we have to track down using exactly the same principles as before, through the root to the .org TLD servers, which in turn also could respond with another out-of-bailiwick server and so on. Every one of these name servers plays a role in name resolution, and any one of them could be a weak link. Research into what is sometimes called “transitive trust” has shown that in extreme cases, more than 400 name servers can be involved in resolving a name. If you give other people even a small proportion of your DNS traffic, make sure that they, and all their DNS servers, are in good shape.

We even can create DNS loops by erroneous delegation. Here's a trivial zone file delegation loop:

```
$ORIGIN example.org.
...
NS ns1.example.net.
NS ns2.example.net.
...
$ORIGIN example.net.
...
NS ns1.example.org.
NS ns2.example.org.
...
```

Although even a superficial glance will show that these two domains names are unresolvable, now assume that they are buried in multiple layers of indirection and transitive trust to perhaps two, three, five or more levels. Debugging can become extremely complex. Further, assume that we add one in-zone name server to example.net's zone file. Our problem now is intermittent errors and timeouts (or slow access) rather than simple non-availability—always a much tougher problem to diagnose.

So far, we have navigated the perils of the DNS hierarchy, but we have not even touched our domain's authoritative name servers. In part two of this article, we will look at what can happen when we move into the user's territory. Now, that can get really scary.■

Ron Aitchison is the author of *Pro DNS and BIND* and loves nothing better than using dig to uncover bizarre DNS configurations. One day, real soon now, he is going to get a real life.

Resources

DNS Statistics: dns.measurement-factory.com

BIND: www.isc.org

BIND Configuration: www.zytrax.com/books/dns

fpdns: www.rfc.se/fpdns

ETech™ Emerging Technology Conference

March 3 – 6, 2008
San Diego Marriott & Marina
San Diego, CA

ETech hones in on the ideas, projects, and technologies that the alpha geeks are thinking about, hacking on, and inventing right now, creating a space for all participants to connect and be inspired.

- Plenary sessions alternate between rapid-fire high-order bits and blue-sky views from visionaries
- Keynotes illuminate the big picture behind emerging technology stories
- Focused breakout sessions translate technology into real-world ideas
- In depth tutorials explore the underlying technology that powers innovations
- Events, extra-curricular activities, and the lively “hallway track” foster unexpected encounters
- The Exhibit Hall brings you up close with the people and products shaping the future of technology

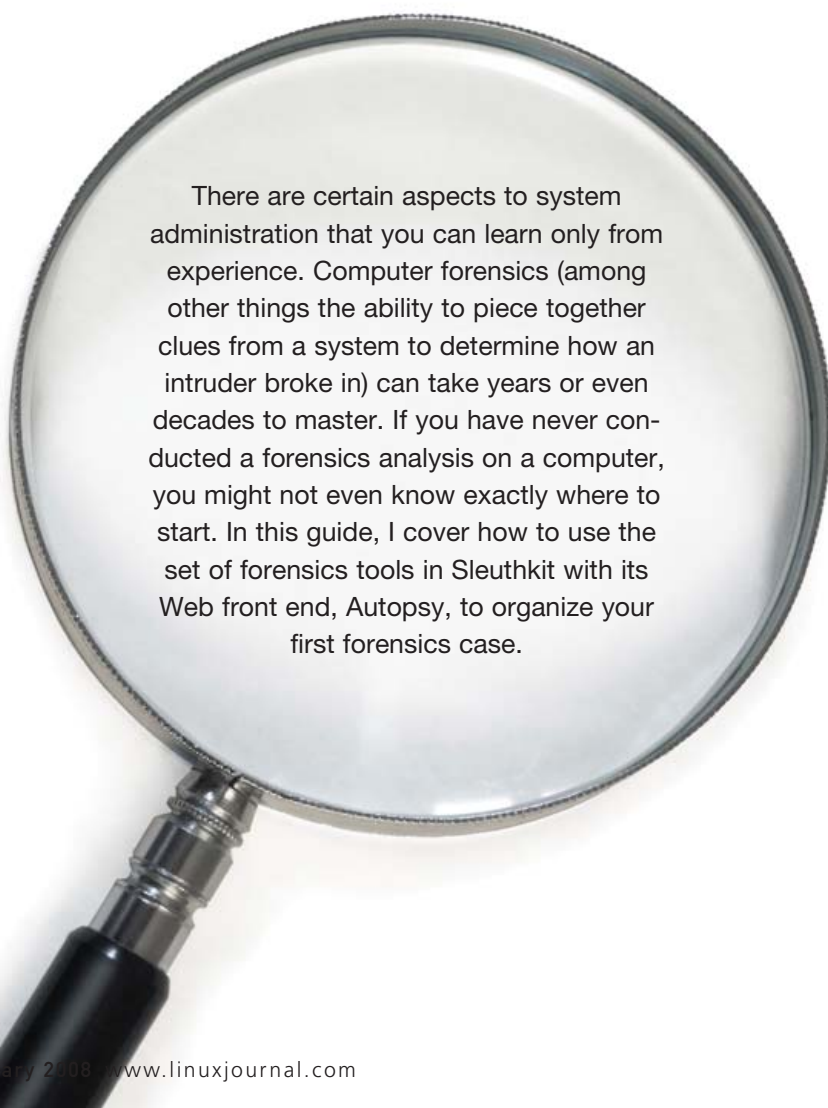
Register and
Save 10% when
you use discount
code **et08ljr**



INTRODUCTION TO FORENSICS

A BREAK-IN CAN HAPPEN TO ANY SYSTEM ADMINISTRATOR ●●●●●

KYLE RANKIN



There are certain aspects to system administration that you can learn only from experience. Computer forensics (among other things the ability to piece together clues from a system to determine how an intruder broke in) can take years or even decades to master. If you have never conducted a forensics analysis on a computer, you might not even know exactly where to start. In this guide, I cover how to use the set of forensics tools in Sleuthkit with its Web front end, Autopsy, to organize your first forensics case.



..... FIND OUT HOW TO USE **AUTOPSY** AND **SLEUTHKIT** TO HIT THE GROUND RUNNING ON YOUR FIRST FORENSICS PROJECT.

Before You Start

One of the most common scenarios in which you might want to use forensics tools on a system is the case of a break-in. If your system has been compromised, you must figure out how the attacker broke in so you can patch that security hole. Before you do anything, you need to make an important decision—do you plan to involve law enforcement and prosecute the attacker? If the answer is yes, you should leave the compromised system alone and make no changes to it. Any changes you make post-attack could complicate and taint the evidence, and because of that, many people have a policy of unplugging a system once they detect an attack and leaving it off until law enforcement arrives. Investigators likely will want the complete system, or at least the drives, so they can store it safely; thus, your forensics analysis might end here until your system is returned.

If you do not plan to prosecute the attacker, you still need to set up some policies beforehand on how to respond to an attack. The first policy you should create concerns whether to pull the power from a compromised server immediately. Two main schools of thought exist on this. One school of thought says that because a live server contains valuable data in RAM, such as running processes, logged in users and so forth, that you should try to collect all of that live data first and then power off the server. The opposing school says that once a system is compromised, all parts of the system are potentially compromised and cannot be trusted, including any tools you might use to grab live system data, so you should pull the power from the server immediately. Otherwise, attackers also could have

examine elsewhere, and then re-install your operating system on the original drives. Remember, once a system has been compromised, you can no longer trust the system. There could very well be a back door that you missed. It's worth saying again that if you plan to prosecute, you will not be able to bring the system back into service, at least not with the original drives as investigators will need them. If you have the extra space, I recommend creating images of your drives to work from and leaving the originals alone. If you accidentally write to the images, you always can create a fresh image from the original drives. Autopsy can manage raw disk or partition images, so any imaging tool, from dd to Ghost, will work.

Install Sleuthkit and Autopsy

For the purposes of this guide, I assume you have created an image of any drives on the system and have stored them on a separate machine that you will use for the forensics analysis. This new machine needs to have both Sleuthkit and Autopsy installed. Some distributions have both Sleuthkit and Autopsy available as precompiled packages, so you can use your distribution's package manager to install them. Otherwise, you can download and compile the tools from the tarballs available on the main project site, sleuthkit.org.

Autopsy works as a Web-based front end to all of the Sleuthkit tools and makes it easy to examine a filesystem without learning each of the different command-line tools. Autopsy also makes it easy to organize multiple forensics analyses into different cases, so you can reference them later. Once Autopsy is installed, get root privileges, and type `autopsy` into a terminal to start the program. Instructions on Autopsy's settings appear in the terminal, including the



Figure 1. Default Autopsy Page

describe your case, and you also can provide a list of investigators who will work on the case. Once your case is named and created, you will see the case gallery—a page that simply lists all the cases you have created. If this is your first case, simply click OK to proceed to the Host Gallery. The Host Gallery lists all the servers you are investigating for this case. In our example, only one host was compromised, but often an attacker will move from one compromised host to another, so include as many hosts as you need to investigate in this gallery. As with the Case Gallery, click Add Host to fill out information about the host you are adding.

You will see some interesting fields on the Add Host page relating to time. If the host was set to a time zone different from your local time zone, be sure to put its time zone in the Time Zone field. When you piece together a chain of events, especially across multiple hosts, having correctly synced time is valuable. The Timeskew Adjustment field lets you account for a server with out-of-sync time, and Autopsy automatically adjusts the times to reflect any skew you put in this field.

When you add the host and go back to the Host Gallery, select the host to analyze and click OK to go to the Host Manager page. If this is a new host, the first thing you should do is click Add Image File to add the image you created previously. The image page has only three fields: Location, Type and Import Method. Autopsy expects that the image is available somewhere on the local computer—either actually on the local disk or via an NFS or SMB mount. Type the full file path to the image file in the Location field. The Type field lets you inform Autopsy of the type of image you created. If you imaged the entire drive, select Disk; otherwise, select Partition. If you select Disk, Autopsy scans the partition table for you and lists all the image's partitions.

Autopsy needs the image file to be in its evidence locker in some form, and the Import Method field lets you choose how to put the

REMEMBER, ONCE A SYSTEM HAS BEEN COMPROMISED, YOU CAN NO LONGER TRUST THE SYSTEM.

compromised shutdown scripts to remove their tracks. I personally lean more toward the second school of thought and believe that no commands should be run and no changes made to a system once a break-in is discovered.

The second policy you should create beforehand concerns how and whether to image the hard drives on the system and how and when to bring the system back into service. If you cannot tolerate much downtime on the system, you probably will want to create an exact image of the drives to

default location for evidence (`/var/lib/autopsy`) and the default port on which it listens (9999). Open a Web browser and type in `http://localhost:9999/autopsy` to view the default Autopsy page and start your investigation.

From the main Autopsy page, click Open Case to open a case you already have created, or for this example, click New Case. In the New Case page, you can name and

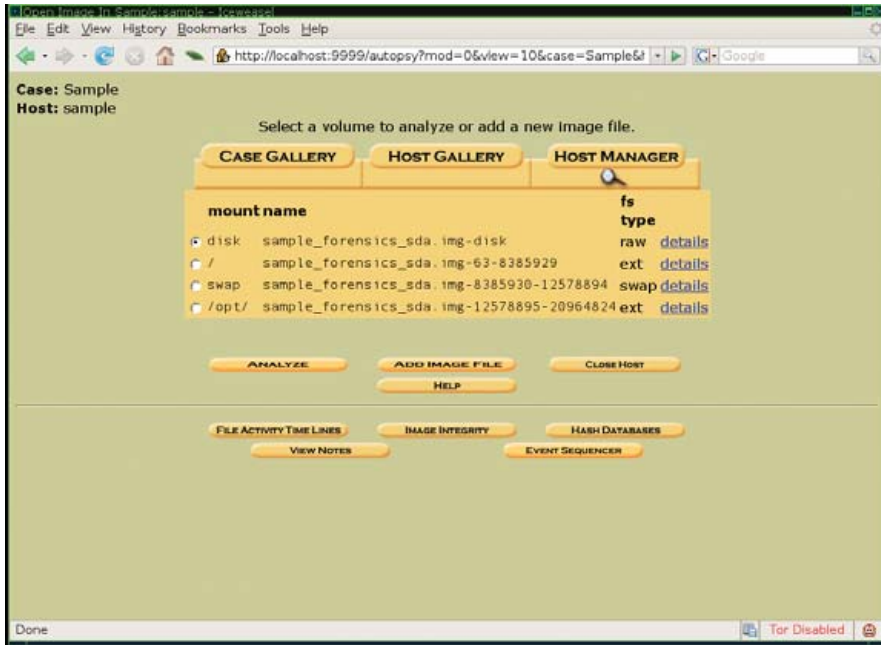


Figure 2. Host Manager Page

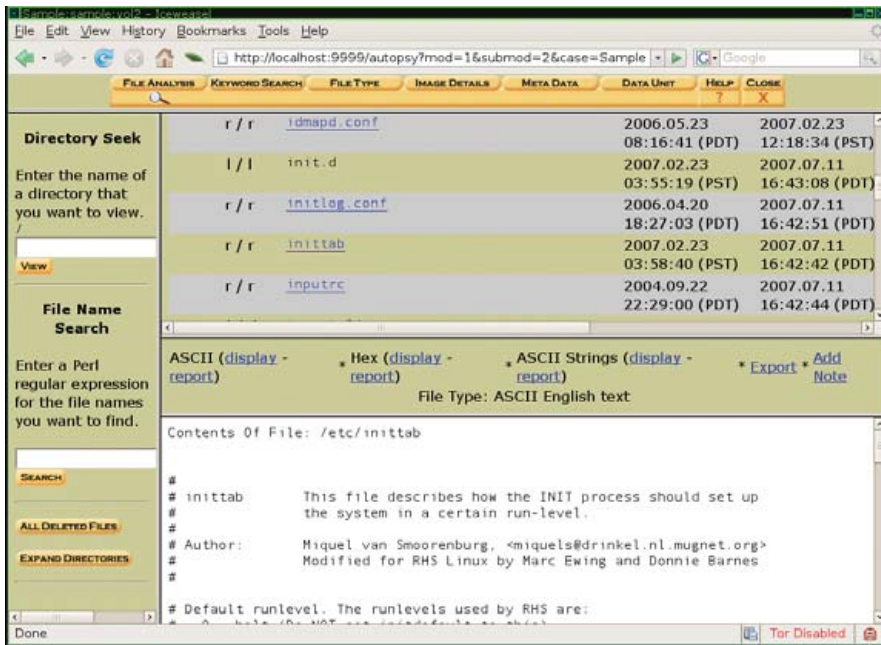


Figure 3. File Analysis

image file there. If you store all your Autopsy evidence on a separate USB drive, you may want to select Copy, so that a copy of the image stays with the rest of the evidence. If your evidence locker is on your local disk along with the image (which is likely under the default settings), select Symlink or Move, depending on whether you want the image to stay in its original location. Repeat these steps to add any additional images for your host.

Start the Investigation

Now that you have created the case, added a host and selected any disk images, you are ready to start the analysis. On the Host Manager page, you will see all the partitions available to analyze. The root (/) partition is a good place to start, so select it, and click Analyze. The Analyze page lists a number of different ways to investigate the filesystem, but click the File Analysis button along the

top of the screen to enter one of the main pages you will use for this analysis.

The File Analysis page gives you a complete view of the filesystem, starting at its root. The top-right frame lists all the files in the current directory, along with additional information each in its own field, including MAC times, permissions and file size. MAC (Modified, Accessed and Changed times), refers to three different changes the filesystem keeps track of for each file. The modified time is the last time the file or directory actually was written to. For instance, if you open a text file, edit it and save the changes, this updates the modified time. The access time is the last time the file or directory was accessed at all. Reading a file updates its access time, and listing the contents of a directory also updates its access time. The changed time keeps track of the last time the file's metadata (such as file permissions and owner) were changed. It's possible, in some cases, for some or all of these times to match.

Each of the files or directories in the File Analysis page are hyperlinked. If you click a directory, the page changes to list the contents of that directory. If you click a file, the bottom-right frame changes to list the contents of the file (even if it's binary) along with a number of functions you can perform on that file. You can display the ASCII or Hex versions of a file or have Autopsy scan the file and display only the ASCII strings inside. This feature is particularly handy to try on suspected trojan files. Often the ASCII strings inside a trojan binary list strange IRC channels or other remote servers or passwords the attacker is using. You also can export a copy of the file to your local disk for further examination.

Attackers often like to delete files to cover their tracks, but Autopsy can attempt to recover them from the free space on the filesystem. Go to the File Analysis page, click the All Deleted Files button on the bottom of the left-hand frame, and Autopsy lists all the deleted files it finds on the system. If Autopsy can recover that much information, you also can see the MAC times and may even be able to click on the file and recover its original contents!

All of these features are handy, but one of the most useful is the Add Note feature. If, for instance, you notice a system binary in your /bin directory that has a strange recent modified date and you notice some suspicious ASCII strings inside, you could click Add Note and list your findings. On the Add Note page, you also can add a sequencer event based on MAC time. If you thought the modified time was suspicious, you might select M-Time on the Add Note page. When

you add notes like this for a number of files or directories, you end up with a large series of notes on what you have found along with interesting times. From the Host Manager window (the window that lists the host's partitions), click View Notes to see the list. This is an invaluable feature when you are trying to piece together the sequence of events from an attacker—particularly when you want to share your findings with others.

If you find a piece of information, such as an IP address or a particular server name as you scan files, you also can click Keyword Search at the top of the Analysis page to scan the entire filesystem for that keyword. You might find log entries or additional files the attacker uploaded that reference that keyword in unlikely places with this tool.

One thing you will discover is that the sequence of events is very important when figuring out an attacker's steps. The File Analysis window lets you sort by any of the headers, including the MAC times. An attacker often will replace a system binary under /bin or /sbin with a trojan, and because that will update the modified time for a file, if you sort the /bin and /sbin directories by modified time in the File Analysis window, you quickly can see suspicious file changes, such as a series of core programs, like ls, vi and echo, all modified a few days ago at a time when you know you didn't update any programs.

Where to Search

If you are new to forensics, you might not be sure of exactly where to start looking in your filesystem. A few directories often contain evidence of an attack that will at least give you a starting point. I've already mentioned the /bin and /sbin directories, as attackers often replace core system binaries in these directories with trojans. The /tmp and /var/tmp directories also are favorite locations, as any user on the system can write to them, so attackers often start their attacks in these directories and download rootkits and other tools here. Pay particular attention for hidden directories (directories that start with a .) in /var/tmp, as that's one way for attackers to cover their tracks from a casual observer. Finally, scan under /home and /root for suspicious files or strange commands in each users' .bash_history file.

What you hope to find is some idea of when attackers were active on your system. Once you have an idea of when the attackers were there, you can check file access and modify times during that period to track down where the attackers were on your system and which files they touched. Although you certainly could browse through the File Analysis

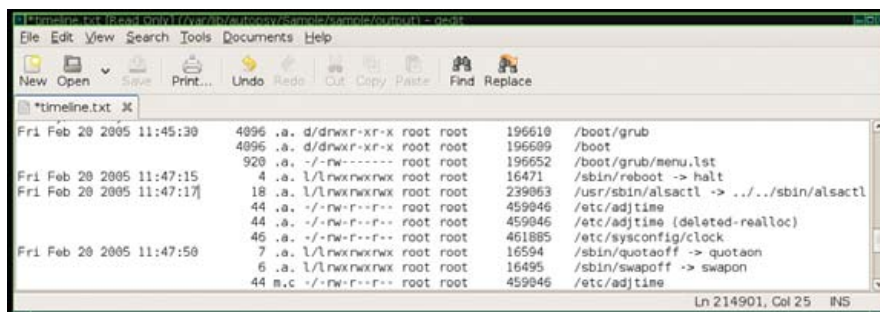


Figure 4. Sample timeline.txt File

window directory by directory, Autopsy provides an easier way via its File Activity Time Line. If you are currently in the File Analysis window, click Close to return to the main Host Manager window that lists the images you have added for your host. From there, click the File Activity Time Line button. Next, click Create Data File, click the check box next to all of the images it lists, and then click OK. This job will take some time, depending on the size and speed of your disk and your CPU.

Once the data file is created, click OK to proceed to the Create Timeline window. In this window, you can narrow down your timeline so that it lists only a particular time period; however, just leave all the options as they are for now and click OK. As you never exactly know where an investigation will lead, you don't want to rule out periods of time that might have valuable clues. When the timeline has been created, click OK to view the Web-based timeline viewer, but a note on that page gives a valuable tip—the timeline is easier to view via a text editor than from the Web interface. Find the raw timeline text file under /var/lib/autopsy/case/host/output/timeline.txt. If you named your case Investigation1 and your host Gonzo, you can find the file under /var/lib/autopsy/Investigation1/Gonzo/output/timeline.txt.

The timeline.txt file lists every file on your image sorted by MAC time. This file contains a lot of information, but once you figure out what each field stands for, it's easier to decipher. The first column lists the time in question for a file followed by the file size. The next field denotes whether this time was a time the file was modified, accessed, changed or any combination of the three. If a file was both modified and accessed at this time, but its metadata was not changed, you would see "ma." in this field. The next field lists the file permissions, followed by the user and group that owned the file. The final two fields list the filesystem inode and the full path to the file or directory. Note that if a group of files has the same time, only the first time field is filled.

If you have found one of the attackers' files, try to locate it in the timeline and see what other files were accessed and especially modified during that time period. With this method, you often can see a list of accessed files that show someone compiling or executing a program. If you notice that the attackers used a particular account on the system, use the File Analysis window to check the /home/username/.bash_history for that user and see any other commands the attackers might have run.

In addition, look at the login history, which often is found under /var/log/messages, for other times that user has logged in and try to correlate those times with any other file activity on the system inside the timeline.txt file. Remember to add notes for each clue you find—as you dig further and further into the filesystem, it can be difficult to keep track of all the different files and how they correlate, but the notes page makes it easy to see. The ultimate goal is to try to locate the earliest time attackers left tracks on the system and use that information to figure out how they got in.

As you might gather, thorough forensics analysis can be a time-consuming process. Even with a tool like Autopsy, it still takes time and experience to make sense of all of the data it presents so you can piece together an attack. One easy way to gain experience is to image your personal system and view it through Autopsy. Create a timeline and see whether you can track down some of the commands you last ran or files you last edited. You might possibly even want to attack your own machine and see if you can use Autopsy to retrace your steps. Although nothing can replace real data, this sort of practice goes a long way toward understanding forensics so you're prepared when a real attack occurs. ■

Kyle Rankin is a Senior Systems Administrator in the San Francisco Bay Area and the author of a number of books, including *Knoppix Hacks* and *Ubuntu Hacks* for O'Reilly Media. He is currently the president of the North Bay Linux Users' Group.

Hear Yourself Think Again!



WhisperStation™ **Cool... Fast... Silent!**

For 64-bit HPC, Gaming and Graphic Design Applications

Originally designed for a group of power hungry, demanding engineers in the automotive industry, WhisperStation™ incorporates two dual core AMD Opteron™ or Intel® EM64T™ processors, ultra-quiet fans and power supplies, plus internal sound-proofing that produce a powerful, but silent, computational platform. The WhisperStation™ comes standard with 2 GB high speed memory, an NVIDIA e-GeForce or Quadro PCI Express graphics adapter, and 20" LCD display. It can be configured to your exact hardware specification with any Linux distribution. RAID is also available. WhisperStation™ will also make a system administrator very happy, when used as a master node for a Microway cluster! Visit www.microway.com for more technical information.

Experience the "Sound of Silence".

Call our technical sales team at 508-746-7341 and design your personalized WhisperStation™ today.



Microway
Technology you can count on™

**THE PERFECT NAC SOLUTION FOR BOTH WIRED
AND WIRELESS NETWORKS**

PacketFence Revisited

In our initial PacketFence article in the April 2007 issue of *LJ*, we introduced the great network access control (NAC) solution that rivals the best ones on the market. On that occasion, we covered ARP-based isolation, which works relatively well for small networks.

Unfortunately, ARP-based isolation can't really scale to many thousands of nodes and is relatively easy to bypass with a simple static ARP table. Thus, we, at Inverse, decided to improve PacketFence by adding a VLAN-based isolation mode. This addition, combined with other enhancements, makes the solution suitable for large-scale networks.

REGIS BALZARD AND DOMINIK GEHL

Introduction to VLAN Isolation

The purpose of PacketFence's VLAN isolation is to assign any device connected to the network to an appropriate VLAN based on its MAC, registration and violation status. A simple scenario would be that every new device belongs to a registration VLAN, a registered and violation-free device belongs to a "normal" VLAN and a device with open violations belongs to an isolation VLAN. If the isolation and registration VLANs are not routed to the "normal" VLAN, PacketFence fully isolates the new device, and any device that violates network policy, from the regular network, efficiently preventing any attack or virus propagation. Of course, real networks are a bit more complicated, and VoIP phones may not end up in the same VLAN as regular computers or servers. But, whatever your network's VLAN design, PacketFence is up to the task.

In order for the VLAN isolation code to work properly, you must use manageable switches. In particular, the switches must provide a means to change a port's VLAN remotely and must be able to send SNMP

(Simple Network Management Protocol) traps to PacketFence. SNMP traps include the switch's IP address; the port number and, depending on the trap type, could include the port status (for example, "up" when a device has been connected and "down" when disconnected); the MAC address of the device (mostly for MAC notification and security violation traps); the number of MACs connected to the switch port and so on.

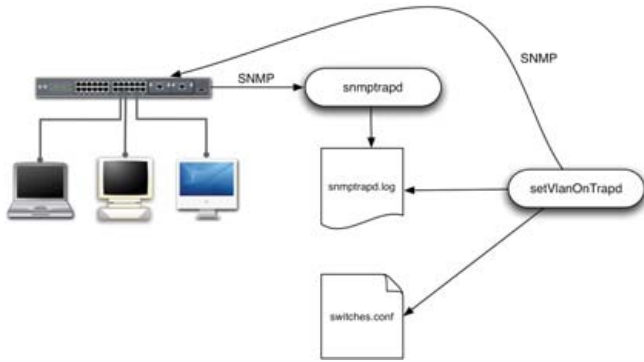


Figure 1. PacketFence's Handling of SNMP Traps

When a switch sends an SNMP trap to PacketFence, the snmptrapd daemon receives it and writes it to the log file `/usr/local/pf/contrib/log/snmptrapd.log`. PacketFence's setVlanOnTrapd daemon continuously reads this log file and, for every new trap, determines whether it needs to change a port's VLAN. If this is the case, it sends the appropriate SNMP commands to the switch.

A crucial part of VLAN isolation is knowing when a device connects to or disconnects from the network. In early 2006, we started the development of the VLAN isolation code by supporting two very basic SNMP traps: linkup and linkdown traps. The vast majority of switch vendors support these two traps, which made our implementation immediately usable on a wide variety of networking hardware. Unfortunately, simply relying on linkup and linkdown traps has its downsides, from both a performance and a functional perspective, including:

- Because a switch needs to see some network traffic on a port to determine the MAC address of the connecting device, linkup traps cannot include any MAC address. PacketFence's setVlanOnTrapd must, therefore, repeatedly query the switch after every linkup trap in order to determine the MAC address of the newly connected device, which introduces some overhead.
- Most VoIP phones contain a built-in switch to connect a PC. The switch sends the linkup trap when you connect the phone; when you connect the PC to the phone, the switch won't send a second linkup trap. Therefore, in this deployment scenario, relying solely on linkup and linkdown traps does not provide enough information to setVlanOnTrapd to work correctly.

One possible solution to address these issues is MAC notification traps. Every time a switch learns a MAC address on a port, it sends a "MAC learned" trap that includes the MAC address. And, of course, PacketFence now also supports MAC notification traps.

In addition to assigning an appropriate VLAN to devices when they connect to the network, PacketFence also isolates devices already connected to the network when they violate the network

policy. Two different options are available:

- PacketFence can briefly disconnect a device from the network by administratively shutting down the switch port and re-opening it soon after. In this case, the switch sends a linkdown, followed by a linkup trap. When PacketFence receives the linkup trap, it determines that the device has an open violation and switches the port to the isolation VLAN. On the computer side, the network adapter notices that the network link went down and automatically renews its IP address—this time in the isolation VLAN. PacketFence's captive portal then informs the user that he or she has been isolated.
- Administratively shutting down a switch port can be problematic when using VoIP phones, as doing so might end a call. If PacketFence has access to the isolation VLAN, you don't actually need to shut down the port. Changing the port's VLAN and doing some ARP spoofing generally are sufficient to make the captive portal available to the user.

So far, we've mentioned only the registration and isolation VLANs, but PacketFence uses a third VLAN, the MAC detection VLAN. This VLAN, which is the default one of every port, must not contain access to any DHCP server and could be seen as an "empty" VLAN. It exists to allow the switch to learn the MAC address of a newly connected device before it obtains an answer from a DHCP server.

Example Installation

Install PacketFence 1.7 from www.packetfence.org. If you are using Red Hat EL5 or CentOS5, the easiest way to do so is to install the official RPM.

In this example, we set up VLAN isolation on a Cisco Catalyst 2960 switch (IP address 192.168.0.3). The PacketFence server's IP address is 192.168.0.10. Let's further assume that you are using the following VLANs and that these already have been created on your switch:

1. "normal" VLAN
2. isolation VLAN
3. registration VLAN
4. MAC detection VLAN

Enable the SNMP traps globally on the switch with the following commands:

```

snmp-server enable traps snmp authentication linkdown linkup
snmp-server enable traps mac-notification
snmp-server host 192.168.0.10 version 2c public mac-notification snmp
mac-address-table notification interval 0
mac-address-table notification
mac-address-table aging-time 300
  
```

Then, configure every access port PacketFence should be handling with the following:

```

switchport access vlan 4
  
```

FEATURE PacketFence Revisited

```
switchport mode access
snmp trap mac-notification added
spanning-tree portfast
```

Edit the VLAN isolation configuration file `/usr/local/pf/conf/switches.conf`, so that it contains the correct SNMP community strings. Then, adjust the VLAN definition as follows:

```
vlan = 1,2,3,4
normalVlan = 1
isolationVlan = 2
registrationVlan = 3
macDetectionVlan = 4
```

And, finally, add a new section for your switch:

```
[192.168.0.3]
ip = 192.168.0.3
type = Cisco::Catalyst_2960
mode = production
uplink = 23,24
```

The purpose of PacketFence's VLAN isolation is to assign any device connected to the network to an appropriate VLAN based on its MAC, registration and violation status.

Next, you can do a communication test between PacketFence and the switch:

```
/usr/local/pf/bin/pfcmd_vlan -getVlan -switch 192.168.0.3 -ifIndex 10001
```

The next test is to determine whether the switch can send SNMP traps to PacketFence. Start `snmptrapd`:

```
service snmptrapd start
```

and observe the log file:

```
tail -f /usr/local/pf/logs/snmptrapd.log
```

Every time a device connects to and disconnects from the network, you should see a new line in the log file.

Now, configure PacketFence's access to VLAN 1, 2 and 3. Set the configuration of the switch port that PacketFence plugs into to "trunk mode", and allow packets in VLAN 1 to pass through the switch without tagging. On the PacketFence server, add two new NICs that read and write 802.1q tagged packets for VLAN 2 and 3. Don't forget to add these new NICs to your configuration file `/usr/local/pf/conf/pf.conf`.

To simplify the installation, configure a DHCP service on the PacketFence box for VLANs 2 and 3. The DHCP server should return its own (VLAN-specific) IP address as both the gateway and the DNS server. Last but not least, set up a "fake" DNS service for VLANs 2 and 3 that responds to all queries with its own IP address. Now, verify that a host connected to the registration VLAN is able to obtain an IP address and that whatever DNS query it sends, PacketFence answers with its own IP.

If all these tests work fine, you can start `setVlanOnTrapd`:

```
service setVlanOnTrapd start
```

and look at the log file to verify that your devices, upon connection to the switch, are assigned to the correct VLAN:

```
tail -f /usr/local/pf/logs/setvlanontrapd.log
```

This setup should be transparent for already-registered devices, because they end up in the "normal" VLAN; unregistered devices will be assigned to the registration VLAN where all they can access is the PacketFence server that will show the captive portal with the registration screen.

Introduction to Isolation in Wireless Networks

PacketFence also integrates very well with wireless networks. As for its wired counterpart, the switch, a wireless Access Point (AP) needs to implement some specific features in order for the integration to work perfectly. In particular, the AP needs to support the following:

- Several SSIDs with several VLANs inside each SSID.
- Authentication against a RADIUS server.
- Dynamic VLAN assignment (through RADIUS attributes).

- SNMP deauthentication traps.

- The deauthentication of an associated station.

We then can configure two SSIDs on the AP, the first one reserved for visitors and unregistered clients. In this SSID, communications will not be encrypted, and users will connect either to the registration VLAN or the visitor's VLAN (depending on their registration status). The second SSID will allow encrypted communications for registered users. The VLANs here are the "normal" VLAN and the isolation VLAN (if ever there are open violations for the MAC).

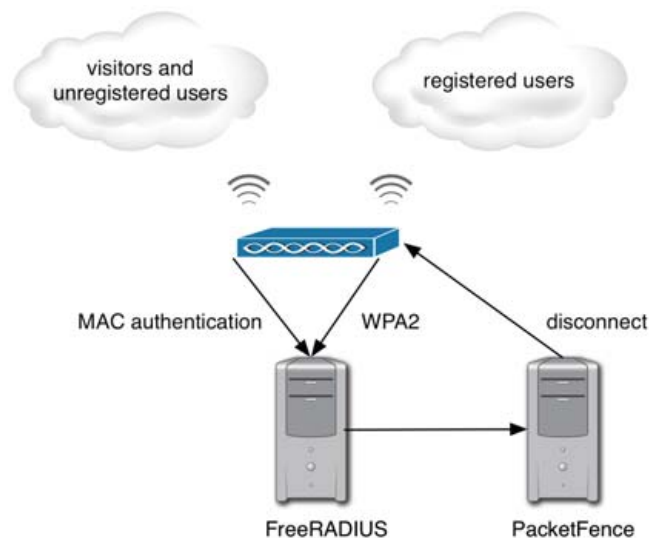


Figure 2. How PacketFence Integrates with Wireless Networks

Example Installation

In this example, we configure a Cisco 1242 AP (IP address 192.168.0.4). Configuration of other vendors' APs is similar. First, define the normal, isolation, registration and visitor VLANs on the AP, together with the appropriate wired and wireless interfaces as shown for the isolation VLAN:

```
dot11 vlan-name isolation vlan 2
```

```
interface FastEthernet0.2
 encapsulation dot1Q 2
 no ip route-cache
 bridge-group 253
 no bridge-group 253 source-learning
 bridge-group 253 spanning-disabled
```

```
interface Dot11Radio0.2
 encapsulation dot1Q 2
 no ip route-cache
 bridge-group 253
 bridge-group 253 subscriber-loop-control
 bridge-group 253 block-unknown-source
 no bridge-group 253 source-learning
 no bridge-group 253 unicast-flooding
 bridge-group 253 spanning-disabled
```

Then, create the two SSIDs:

```
dot11 ssid WPA2
 vlan 2 backup normal
 authentication open eap eap_methods
 authentication key-management wpa
 accounting acct-methods
 mbssid guest-mode
```

```
dot11 ssid MACauth
 vlan 3 backup visitor
 authentication open mac-address mac_methods
 accounting acct_methods
 mbssid guest-mode
```

Configure the RADIUS server (we assume here that the FreeRADIUS server and the PacketFence server are located on the same box):

```
radius-server host 192.168.0.10 auth-port 1812
 >acct-port 1813 key secretKey
```

```
aaa group server radius rad_eap
 server 192.168.0.10 auth-port 1812 acct-port 1813
 aaa authentication login eap_methods group rad_eap
```

```
aaa group server radius rad_mac
 server 192.168.0.10 auth-port 1812 acct-port 1813
 aaa authentication login mac_methods group rad_mac
```

Enable the SNMP deauthentication traps:

```
snmp-server enable traps deauthenticate
 snmp-server host 192.168.0.10 public deauthenticate
```



Want your business to be more productive?
The ASA Servers powered by the Intel Xeon Processor provide the quality and dependability to keep up with your growing business.

Hardware Systems for the Open Source Community Since 1989
(Linux, FreeBSD, NetBSD, OpenBSD, Solaris, MS, etc.)

1U Woodcrest/Clovertown Storage Server Starts at - \$1,741



- 1TB Storage installed. Max - 3TB.
- 1U Dual core 5030 CPU (Qty-1). Max - 2 CPUs.
- 1GB 667MGZ FBDIMMs Installed.
- Supports 16GB FBDIMM.
- 4X250GB htswap SATA-II Drives installed.
- 4 port SATA-II RAID controller.
- 2X10/100/1000 LAN onboard.

2U Woodcrest/Clovertown Storage Server Starts at - \$3,791

- 4TB Storage installed. Max - 12TB.
- 3U Dual core 5050 CPU.
- 1GB 667MGZ FBDIMMs Installed.
- Supports 16GB FBDIMM.
- 16 port SATA-II RAID controller.
- 16X250GB htswap SATA-II Drives installed.
- 2X10/100/1000 LAN onboard.
- 800w Red PS.



3U Woodcrest/Clovertown Storage Server Starts at - \$3,991



- 4TB Storage installed. Max - 12TB.
- 3U Dual core 5050 CPU.
- 1GB 667MGZ FBDIMMs Installed.
- Supports 16GB FBDIMM.
- 16X250GB htswap SATA-II Drives installed.
- 16 port SATA-II RAID controller.
- 2X10/100/1000 LAN onboard.
- 800w Red PS.

5U Woodcrest/Clovertown Storage Server Starts at - \$6,691

- 6TB Storage installed. Max - 18TB.
- 5U Dual core 5050 CPU.
- 4GB 667MGZ FBDIMMs Installed.
- Supports 16GB FBDIMM.
- 24X250GB htswap SATA-II Drives installed.
- 24 port SATA-II RAID. CARD/BBU.
- 2X10/100/1000 LAN onboard.
- 930w Red PS.



8U Woodcrest/Clovertown Storage server Starts at - \$11,191



- 10TB Storage installed. Max - 30TB.
- 8U Dual core 5050 CPU.
- 2X5050 installed.
- 1GB 667MGZ FBDIMMs.
- Supports 32GB FBDIMM.
- 40X250GB htswap SATA-II Drives installed.
- 2X12 Port SATA-II Multiline RAID controller.
- 1X16 Port SATA-II Multiline RAID controller.
- 2X10/100/1000 LAN onboard.
- 1850 W Red Ps.

All systems installed and tested with user's choice of Linux distribution (free). ASA Collocation—\$75 per month



2354 Calle Del Mundo,
Santa Clara, CA 95054
www.asacomputers.com
Email: sales@asacomputers.com
P: 1-800-REAL-PCS | FAX: 408-654-2910

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

Prices and availability subject to change without notice. Not responsible for typographical errors.



FEATURE PacketFence Revisited

Finally, activate the SSIDs on the radio:

```
interface Dot11Radio0
  encryption vlan 1 mode ciphers aes-ccm
  encryption vlan 2 mode ciphers aes-ccm
  ssid WPA2
  ssid MACauth
```

In addition to assigning an appropriate VLAN to devices when they connect to the network, PacketFence also isolates devices already connected to the network when they violate the network policy.

Now, check with a Wi-Fi card that you actually can see the two new SSIDs. You can't connect to them yet because the RADIUS server is not up and running.

Start configuring the FreeRADIUS server by adding the following lines at the end of `/etc/raddb/clients.conf`:

```
client 192.168.0.3 {
  secret = secretKey
  shortname = AP1242
}
```

In `/etc/raddb/eap.conf`, set the default eap type to peap at the beginning of the eap {} section:

```
default_eap_type = peap
```

And, set up your cryptographic keys in the tls {} section.

Then, update `/etc/raddb/radiusd.conf`, first by adding the following lines to the modules {} section:

```
perl {
  module = ${confdir}/rlm_perl_packetfence.pl
}
```

Then, add "perl" at the end of the authorize {} section. The script `/etc/raddb/rlm_perl_packetfence.pl` uses the Calling-Station-Id RADIUS request attribute, containing the MAC of the wireless station to determine its registration and violation status. Based on this information, it sets the Tunnel-Medium-Type, Tunnel-Type and Tunnel-Private-Group-ID RADIUS reply attributes. The AP, upon reception of these three attributes, then confines the wireless station into the specified VLAN.

The last file to edit is `/etc/raddb/users` to define that non-EAP messages should, by default, lead to an authentication acceptance:

```
DEFAULT EAP-Message !* "", Auth-Type := Accept
```

Then, add a local test user with:

```
testUser User-Password == "testPwd"
```

Now, start FreeRADIUS in debug mode:

```
radiusd -x
```

Try to connect to one of the two new SSIDs with your Wi-Fi card, and you'll see the packets received by FreeRADIUS with the generated responses.

It also is noteworthy that the concepts you've learned here on using PacketFence with wireless networks are identical to using 802.1x on a wired network, which of course, is supported by PacketFence.

Extending to New Switches and APs

We designed the VLAN isolation feature from the beginning with extensibility in mind. All supported switches are represented through Perl objects, and we make extensive use of inheritance. For example:

- At the highest level, you have the `pf::SNMP` object that defines general functions, such as SNMP session creation and deletion, database connections and some standardized SNMP queries.
- At the next level are the vendor-specific objects, such as `pf::SNMP::Cisco` and `pf::SNMP::Nortel`. They include the necessary functions to parse SNMP traps and, most of the time, to read and write a port's VLAN assignment.
- Finally, at the lowest level, are the model-specific objects, containing only the model-specific code.

This architecture simplifies adding the support for a new product from an already-supported vendor; it comes down to redefining only a very limited number of methods and can be done in a matter of hours.

Conclusion

As you've seen in this article, PacketFence secures both wired and wireless networks in an efficient way. Offering the same level of security and using the same NAC solution on both networks make PacketFence one of the most essential security tools to have. ■

Regis Balzard (rbalzard@inverse.ca) holds a Computer Engineering degree from the Ecole Supérieure d'Ingenieurs en Genie Electrique (ESIGELEC) in Rouen, France. He is currently a systems architect for Inverse, Inc., an IT consulting company located in downtown Montréal that specializes in the deployment of infrastructures based on free and open-source components like PacketFence and SOGo.

Dominik Gehl (dgehl@inverse.ca) holds a Master's degree in Computer Science from the University of Montréal. He is currently a systems architect for Inverse, Inc., an IT consulting company located in downtown Montréal that specializes in the deployment of infrastructures based on free and open-source components like PacketFence and SOGo.

Resources

"PacketFence" by Ludovic Marcotte and Dominik Gehl, *LJ*, April 2007: www.linuxjournal.com/article/9551

PacketFence: www.packetfence.org

Inverse—PacketFence VLAN-Based Isolation Mode: www.inverse.ca/contributions/packetfence.html

Net-SNMP—snmptrapd Daemon: www.net-snmp.org

The FreeRADIUS Project: www.freeradius.org

Power of **32** Processors



256GB of Memory

Based on industry leading AMD Opteron™ microprocessors, the **HPC A5808-32** server provides industry leading scalable x64 processing with the latest I/O, network, memory, and power efficient technologies.

With features such as **AMD quad-core Opteron™** microprocessors and up to 256GB of DDR2 for memory-intensive performance, multithreaded applications like CRM, ERP, e-commerce, and virtualization will see significant performance improvements.

And **HPC MasterSight™** delivers features to help manage the system with inclusive diagnostic tools.

Specializing in High Performance Computers, HPC systems, Inc. is a solution provider serving the financial, health, educational, and government sector. We appreciate that you have a choice of dozens of vendors, but not all of them have our uncompromising dedication and experience in producing the best solutions.



HPC Systems, Inc.
48009 Fremont Blvd
Fremont, CA 94538
Tel: 510-656-8282
Fax: 510-656-8341
E-mail: info@hpcsystems.com



Interview with Eric S. Raymond

Eric Raymond on the history and future of open source. GLYN MOODY

Eric Raymond has played a key role in the history of open source. In 1997, he published *The Cathedral and the Bazaar (CatB)*, his seminal analysis of why the open-source development approach works so well. He was one of a group who came up with the term open source and, until 2005, was President of the Open Source Initiative (OSI), which he co-founded. After years as one of computing's most vocal and colorful characters, Raymond has been conspicuous by his absence recently. He tells Glyn Moody why, looks back over the ten years since *CatB* was published, and forward to the future of open source and its ideas.

GM: I can't quite remember the point when you were there and then you weren't, but things have been rather quiet from you recently on the open-source front. What have you been up to?

ESR: I've been keeping a low profile for a number of reasons, some of which I can't talk about but expect to go public with within the next few months. Among the reasons I *can* talk about it is that I made a strategic decision ten years ago not to try to be the indispensable man. That meant at some point I was going to have to step away from various leadership roles deliberately, and I did that (notably by resigning from the presidency of OSI).

I was a historian before I was an activist, and one of the things I've paid attention to are patterns of success and failure in reform movements. One of the things I've observed is movements that remain dependent on the talents and charisma of key individuals don't survive those individuals. In fact, if your charismatic founder doesn't step offstage, healthy reform movements often end up having to ritually execute, banish or disgrace him/her in order to build themselves properly into sustainable institutions.

So one of the things I've been up to is doing nothing, letting myself slip out of the public eye and watching the movement mature. I still do a lot of programming, as I always did. Two of my recent projects have been *gpsd* (a GPS monitoring daemon) and *Battle For Wesnoth* (a rather spiffy fantasy combat game). I still occasionally talk to reporters, and I've done a fair amount of behind-the-scenes work that's important to the community, including some things I still have to be mysterious about for a bit.

GM: Do you have any plans to add your voice to the debate more frequently in the future?

ESR: I've un-stealthed a little in the last few months. I haven't yet decided how much more visible I want to get. In truth, I never really *liked* being famous, which is why I find it more entertaining than infuriating to be accused (as I sometimes am) of being a monster of ego or an attention junkie. Fame was never more than an instrument for me, and it's ridiculously easy to get; if I decide it's tactically necessary again, I'll just fire up the same techniques I used a decade ago and take it.

GM: Looking back over the ten years since *CatB* was published, what surprises you most about your analysis and its effects?

ESR: There was a day in early 1998 that Netscape open-sourced Mozilla, and I thought out the strategy I've been following ever since. Nothing since then has surprised me much. Details, yes, like IBM being the major Fortune 100 ally that stepped up, or the fact that our best shot at the end-user desktop came out of the brain of a former space tourist from South Africa.

But the broad trends, not. For example, I wrote in 1999 that I expected us to have basically won our technical argument for the superior quality of open source by about 2003, and that happened pretty much on schedule. Since then, the arguments you hear against it are mostly about transition costs and lack of a road map. That's a heck of a victory.

The *reason* I haven't been surprised much is that I understand the rise of open source as being driven by long-term trends that go much deeper than hacker ideology or intercorporate rivalries or any of the other colorful stuff that tends to preoccupy people. Even the dot-com boom and the following bust didn't perturb those trend-lines noticeably.

The deep drivers come from things like scaling laws, the way that various different costs and payoff functions in software design shift in response to hardware getting ever less expensive. Linus Torvalds and RMS and I are chips on that wave; we didn't create it, we rode it.

Here's an example of the sort of thing I mean: the Vista flop. Completely predictable, didn't surprise me for a nanosecond, and not because I think Microsoft is staffed by incompetents either. It's not; it hires some of the brightest programmers in the world. But, as I've been explaining for ten years, there's a scale regime above which closed-source development is unsustainable as the ratio between productive work and complexity-management overhead rises. Microsoft was bound to reach it; the only question was when.

GM: What do you think are the most important new lessons we have learned in the last ten years about how open source works?

ESR: I'm not sure we've learned anything fundamental, which actually disappoints me somewhat—I expected my original theoretical models and language to have been thoroughly superseded by now with better ones, and they haven't been. Actually, *that* might count as my biggest surprise since 1998.

That being said, there have been a lot of evolutionary developments that really matter. Our technical toolkits have improved markedly, and so has individual and social consciousness about how to use them. One good example is the rise of project workbenches like SourceForge and Gna and Berlios; another is the shift now underway from centralized to decentralized version-control

systems; and a third is the clever ways development groups are finding to use IRC as a complement to and replacement of traditional e-mail channels.

GM: How do you think the open-source tribe differs today from the one you wrote about ten years ago, or the one that existed even further back? Do those differences matter?

ESR: Oh yes, they certainly do. The two big ones are that we're *conscious* and *unified* now in a way we weren't before. That sounds like a fuzzy feel-goodism, like political rhetoric, but it's actually a sharp fact with hard, measurable consequences, and it may actually be the most interesting thing I can talk about in this interview.

Before *The Cathedral and the Bazaar*, open-source development was a folk practice, a set of working methods evolved unconsciously by hackers who had no theory about why the things they were doing actually worked. It didn't have a name—and no, "free software" wasn't it, because that label was about ideology and goals rather than working methods and communications structures.

After *The Cathedral and the Bazaar*, we got conscious. We woke up. We started *reflecting on what we were doing* and deliberately trying to improve our process efficiency. One measurable consequence of that is the toolkit changes I just listed. These are things that happened because hackers reflected on the open-source process with the conscious intention of improving it.

The "unified" part—the Open Source community has a sense of itself now that has become steadily more important relative to membership in what used to be more loosely connected subtribes—Perl programmers, Emacs fans, Linux users, BSD users, pro- or anti-GPL zealots, whatever. While those affiliations are still important and spawn way too many flamewars, the overarching "open-source" affiliation is now much more important for almost everyone in those subtribes.

There are two ways to look at that shift. One truth is that I engineered it as part of my sinister master plan for world domination, bwahahaha. There are specific things I did, like founding OSI and some choices I made in my early propaganda and people I chose and groomed for leadership positions, that were designed to unify the community and did, in fact, have that effect.

One insignia of the success of my meme-hacking (insert more demented mad-scientist-type cackling here) is the effect on tribes like the BSD people and the X developers, who weren't part of the Linux community to which I addressed my original propaganda. They have not only enlisted themselves into today's Open Source community, they've also reinterpreted their own pre-1997 histories into open-source language and categories!

The equal and opposite truth is that this was bound to happen sooner or later; you can only sleepwalk for so long before you stumble over something and wake up—and the open-source process is not just more important than any of the specific tools or languages or operating systems we apply with it, it's *obviously* more important. So it's not surprising that people's allegiances and self-identifications have shifted; if I hadn't social-engineered that, someone else would have. Later and more slowly, maybe, but I'm certain it would have happened without me eventually.

Significantly, the only subtribe that has even tried to maintain a strong identity in opposition to open source is the more extreme wing of the FSF crowd—and that is precisely because they believe ideology

“Two of my recent projects have been *gpsd* (a GPS monitoring daemon) and *Battle For Wesnoth* (a rather spiffy fantasy combat game).”

and goals are more important than working methods.

For me, the working method *is* the ideology. Our practice is more powerful than our preaching. What really persuades are deeds and results. Or, as I've sometimes put it: talk is cheap, shut up and show them the code.

GM: One remarkable change in the last decade has been the rise of Google, a company built almost entirely on top of open-source software, which employs some of the top hackers and promotes the training of a new hacker generation through its Summer of Code. What do you think are the good and bad aspects of Google as far as open source is concerned?

ESR: I have the same concerns everyone else does. On the one hand, Google is an important patron of the community and has been on the right side of a lot of battles. On the other hand, the concentration of power it's acquiring is enough to make anyone uneasy, and I'm disturbed by the way the founders' political slant has sometimes compromised the neutrality of the channel—the generally leftish tilt of Google News' source selection, for example, or the fact that it won't let you sell firearms through Google Ads. And yes, I'd be just as unhappy if it were conservative dogma apparently manifesting, or even my own off-the-spectrum radical libertarianism. For a company in Google's position, *any* political partisanship outside narrow issues connected to free speech is an abuse of trust.

I'm optimistic, though. Markets correct this sort of thing. One consequence of the Internet is that they're doing it faster all the time, especially for information goods. Microsoft will probably have been the last of the long-lived information monopolies; if Google is seen to emulate, say, the *New York Times'* combination of left-wing partisanship with unconvincing denials of same, I have no doubt that we'll see some countervailing equivalent of Fox News spun up in fairly short order.

GM: The rise of Google has added to Microsoft's difficulties, since it must now fight on two fronts....

ESR: Exactly...and an important subtlety here is that the desktop-Linux and Google-Web-services attacks are *different*. They're not just in defense on two business fronts, they have to beat two strongly divergent technical approaches with one platform. Tough lines.

GM: Microsoft's response to open source has been highly schizophrenic. How do you expect all this to pan out: what will Microsoft do, and what will it become?

ESR: If I had good answers to that, I'd go play the stock market and be rich. I can see the underlying trends like scaling laws clearly, but surface phenomena, like the rise and fall of individual corporations, have too much noise and time-jitter in them.

The only thing I'm sure of is that Microsoft's days of being able to

“This means there’s a huge Ubuntu-shaped power vacuum opening up in the desktop market.”

ship competitive software from closed source are numbered, let alone its days of maintaining monopoly lock-in. The Vista stall-out, and the scaling phenomena beneath it, guarantee that.

GM: What do you see as the big trends in open source at the moment?

ESR: My friend Rob Landley and I wrote a paper a year ago (“World Domination 201”, catb.org/~esr/writings/world-domination/world-domination-201.html), on how the transition to 64-bit hardware opened a critical window of opportunity for mass Linux adoption that is likely to close sometime in 2008. The two most interesting things to happen since are that 1) the hardware trend curves we looked at have been tracking our predictions like they were on rails, and 2) the two major opponents we were worried about have both been falling out.

Vista is a flop, and Apple is steaming away from the desktop market as fast as it can—it took “Computer” out of its corporate name a few months ago and now seems to want to be all about iPods and cell phones and media, oh my. Can’t blame Apple; profit margins and volumes are both higher in those markets.

This means there’s a huge Ubuntu-shaped power vacuum opening up in the desktop market. (A year ago, I thought it might be Linspire, but Linspire blew it big-time on several levels. One of those was soliciting strategic advice from me, swearing to follow it, and then doing the exact opposite.) If Mark Shuttleworth and his merry crew at Canonical don’t blow it, we’re going to win. And Ubuntu seems to me to be doing the right things, like hassle-free support for the evil proprietary multimedia codecs that nontechnical end users actually want.

Unfortunately, not blowing it requires not just doing the right things but doing them fast enough. Time is getting tight. We have at most another year before the market settles on the dominant desktop OS for the 64-bit era. That’s not a lot of release cycles even at open-source speed.

We’re so close. If we had even another nine months on the window, I don’t think I’d be worried. As it is, it’s going to be a damned near-run thing either way.

GM: As open-source commoditizes more and more of the software stack, do you think that free software will run out of big challenges and will eventually turn into a kind of mopping-up operation, filling in the holes?

ESR: No more so than software in general. One clue as to why is that “commoditizing” turns out to be an extremely misleading term for what’s actually going on. I’m writing a paper on that; it’s called “The Art of Commodity”. I think it may surprise people as much as my last one, “The Magic Cauldron”, did.

I’m not going to go into this in a lot of detail right now except to point you and your readers at the same source that started me thinking: Brent Williams’ “Open Source Business Models: A Wall Street Look at a Wild 2006 and the Prospects for Even

More Fun in 2007” (stephesblog.blogs.com/presentations/BrentWilliamsEclipseConV02.pdf).

This guy is a stock analyst who noticed something very interesting about the open-source software market, but lacked the tech background to understand the true significance of the anomaly he spotted.

GM: On a related note, do you think it is possible that the most exciting open-source projects will be those outside software—that is, the application of open-source ideas to content, business, science, politics and so forth? Do you see evidence of that happening already?

ESR: Well, there’s the open-access movement in scientific publishing. The principals in that one are very explicit about their debt to the open-source movement in software, which kind of closes a circle, because one of my key insights was that open-source development harnesses the effects of decentralized peer review the way scientists are supposed to do it.

Politics: I recently read that New Zealand is inviting the public to rewrite a fundamental part of its legal code on a wiki. Business on content: the company that owns *Dungeons & Dragons* released its stuff under an open-content license a few years back. (The funny part about that is it credited me with inspiring the move but probably had no idea that I was a *D&D* player from waaaaay back, like the 1974 first edition. I still have that original rule set in my basement.)

There are polyps of open-source thinking sprouting all over the place. But, most exciting? Not to me. I’m a programmer. Software is what I do, and the increasing pervasiveness of the Internet means that the effects of the *software* open-source movement will be felt everywhere.

There’s also the problem that, unfortunately, most of the people who work themselves into a lather about applying open-source principles elsewhere are...well, let’s be gentle and simply note that their idealism tends to greatly exceed their grasp of consequences.

But ask me again in another ten years; my answer might change.

GM: What do you think the open-source approach can offer to help solve the big challenges facing humanity and the planet today—things like climate change, sea acidification, water shortages, resource depletion and so on?

ESR: It’s obvious. Planned economies, rigid authority hierarchies, closed and secretive decision procedures—they don’t cope with situations of ambiguity, uncertainty and rapid changes in requirements well, if at all. They’re maladaptive because their decision processes get captured by what political economists call “agency problems”.

Every single one of the problems you listed has market-based, decentralist, open-system solutions that are superior to anything a top-down planner or bureaucrat would ever come up with. The danger isn’t that we’ll fail to respond, it’s that we’ll get locked in to “solutions” that do more harm than good simply because they fit somebody’s centralizing, authoritarian political agenda.

If you want a largest lesson from open source, here’s mine: trust decentralization over centralization, voluntarism over coercion, bottom-up over top-down, adaptation over planning, openness over secrecy, practice over ideology, and markets over politics. Freedom works. Now go *do it!* ■

Glyn Moody writes about open source at opendotdotdot.blogspot.com.

SXSW INTERACTIVE

MARCH 7-11, 2008 • AUSTIN TX

SXSW INTERACTIVE FESTIVAL: CONNECT, DISCOVER, INSPIRE

Attracting digital creatives and new media entrepreneurs, the 15th annual South by Southwest (SXSW) Interactive Festival gives you both practical how-to information as well as unparalleled career inspiration. Attend this legendary gathering of the tribes to renew your link to the cutting edge.

Opening Remarks by Henry Jenkins on Saturday, March 8

The Co-Director of the Comparative Media Studies Program at MIT, Jenkins has also authored numerous books including "Convergence Culture: Where Old and New Media Collide" and "Fans, Bloggers and Gamers: Exploring Participatory Culture."

Lea Alcantara (Lealea Design), **Natalie Zee Drieu** (Craft Magazine), **Tim Ferriss** (Author), **Jason Fried** (37signals), **Kelly Goto** (Goto Media), **Brewster Kahle** (Internet Archive), **Nathan Shedroff** (nathan.com), **Tatsuki Taomita** (Opera), and **Susan Wu** (Charles River Ventures).

REGISTER TO ATTEND SXSW 2008

Go to **sxsw.com** now to take advantage of early registration discounts and for up-to-date lists of panels, panelists, speakers and Web Awards finalists.

SOUTH BY SOUTHWEST INTERACTIVE FESTIVAL
March 7-11, 2008 | Austin, Texas | www.sxsw.com



GCC for Embedded Engineers

Read along to understand how GCC works, find out what all those other programs in the toolchain directory do, and learn some tips and tricks to become more comfortable with most indispensable tool in your project. **GENE SALLY**

GCC, the GNU Compiler Collection, is a tool used by nearly every embedded engineer, even those who don't target Linux. In release since 1987, supporting every processor known to man, GCC is a juggernaut of software engineering that, because of its ubiquity and ease of use, doesn't get the admiration it deserves.

When used in an embedded project, GCC capably does another trick, cross-compilation, without complaint. Simply invoke the compiler and the right things will happen. Under the covers, GCC is a very complex tool with lots of knobs to turn to fine-tune the compilation and linking process; this article looks at how to build a GCC cross-compiler, examines the process that GCC uses to compile a program and shares some productivity-boosting tips and tricks.

Building a Cross-Compiler

When starting an embedded project, the first tool needed is a cross-compiler, a compiler that generates code intended to work on a machine different from the one on which the code generation occurred. Sometimes, it's possible to obtain a prebuilt cross-compiler (from a commercial or noncommercial source), short-circuiting the need to build from source; however, some projects require that all tools must be re-creatable from source. No matter why GCC needs to be built, there are several different approaches to building a cross-compiler.

Quite possibly the easiest way is by using the crosstool Project, created by Dan Kegel and hosted at www.kegel.com/crosstool. Using this project involves downloading the source code and making one of the presupplied files feed the right parameters into the script that builds the compiler. The matrix of supported platforms and software versions can be found at www.kegel.com/crosstool/crosstool-0.43/buildlogs, and choosing something that's marked as working will yield a compiler in a few hours. crosstool will download the right software, even the patches, necessary to make the software work on the target platform. However, if the project requires support for an alternate C library, crosstool becomes more difficult to use.

Because many developers want to use uClibc, a smaller implementation of the C library, it's fortunate that this project has something similar to crosstool, called buildroot, located at buildroot.uclibc.net. As a bonus, buildroot, along with building a cross-compiler, also can be used to build a root filesystem for the board based on the related BusyBox Project. The user configures a buildroot run using a process similar to that of the kernel configuration to ready the build. This project doesn't have a chart of known working configurations like crosstool, so finding a working configuration can be difficult.

Finally, for the type of person who doesn't like the idea of wading

through somebody else's build scripts when things don't work, building a cross-compiler by hand isn't as daunting a process as one would expect. The following steps outline the process, where \$TARGET is the target processor and \$INSTALLAT is the directory where the compiler will reside after being built:

1. Download and build binutils:

```
$ tar xzf binutils-<version>.tar.gz
$ ./binutils-<version>/configure --target=$TARGET --prefix=$INSTALLAT
$ make ; make install
```

2. Copy the include and asm from the board's kernel to the installation directory:

```
$ mkdir $INSTALLAT/include
$ cp -rvL $KERNEL/include/linux $KERNEL/include/asm $INSTALLAT/include
```

3. Download and build bootstrap GCC. At this point, it's best to build the bootstrap GCC in its own directory and not the directory where it has been unpacked:

```
$ tar xzf gcc-<version>.tar.gz
$ mkdir ~/$TARGET-gcc ; cd ~/$TARGET-gcc
$ ../gcc-<version>/configure --target=$TARGET --prefix=$INSTALLAT
--with-headers=$INSTALLAT/include --enable-languages="c" -Dinhibit_libc
$ make all ; make install
```

4. Download and build glibc (or alternate libc) with the bootstrap compiler. Like GCC, the build of the library works best when you configure and make outside the source tree:

```
$ tar xzf glibc-<version> --target=$TARGET --prefix=$INSTALLAT
--enable-add-ons --disable-sanity-checks
$ CC=$INSTALLAT/bin/$TARGET-gcc make
$ CC=$INSTALLAT/bin/$TARGET-gcc make install
```

5. Build the final GCC. The bootstrap compiler was built to build the C library. Now, GCC can be built to use the cross-compiled C library when building its own programs:

```
$ cd ~/$TARGET-gcc
$ ../gcc-<version>/configure --target=$TARGET --prefix=$INSTALLAT
--with-headers=$INSTALLAT/include --enable-languages="c"
$ make all ; make install
```


At the end of this process, the newly built cross-compiler will be at \$INSTALLAT/bin. An oft-used strategy for those needing a specially configured GCC is to use crosstool or buildroot to download and patch the source files and then interrupt the process. At this point, the user applies additional patches and builds the components with the desired configuration settings.

Before leaving this section, there's a frequently asked question from embedded engineers targeting Pentium machines doing development on desktops that are essentially same. In this case, is a cross-compiler necessary? The answer is yes. Building a cross-compiler for this configuration insulates the build environment and library dependencies from the development machine that happened to be used to build the source code. Because desktop systems can change revisions several times a year, and not all team members may be using the same version, having a consistent environment for compiling the embedded project is essential to eliminate the possibility of build configuration-related defects.

The Toolchain: More Than Just a Compiler

The collection of programs necessary to compile and link an application is called the toolchain, and GCC, the compiler, is only one part. A complete toolchain consists of three separate parts: binutils, language-specific standard libraries and the compiler. Notably absent is the debugger, which is frequently supplied with the toolchain but is not a necessary component.

binutils

binutils (binary utilities), performs the grunt work of manipulating files in a way that's appropriate for the target machine. Key parts of the toolchain, such as the linker and assembler, reside in the binutils Project and aren't part of the GCC Project.

Hidden inside the binutils Project is another nifty bit of software, the BFD library, which technically is a separate project. The BFD, Binary Descriptor Library (the actual acronym unpacks to something too bawdy for this publication), provides an abstract, consistent interface to object files, such as handling details like address relocation, symbol translation and byte order. Because of the features supplied by BFD, most tools that need to read or manipulate binaries for target reside in the binutils Project to best take advantage of what BFD has to offer.

For the record, binutils contains the following programs:

- **addr2line**: given a binary with debugging information and an address, returns the line and file of that address.
- **ar**: a program for creating code archives that are a collection of object files.
- **c++filt**: demangles symbols. With classes and overloading, the linker can't depend on the underlying language to provide unique symbol names. c++filt will turn `_ZN5pointC1ERKS_` into something readable. A godsend when debugging.
- **gprof**: produces reports based on data collected when running code with profiling enabled.
- **nlmconv**: converts an object file into a Netware Loadable Module (NLM). If you've ever worked with NLMs, you probably did so with

your collar turned up and cringed when seeing ABEND on your terminal. It's noted here because nlmconv is rarely, if ever, distributed with a toolchain.

- **nm**: given an object file, lists symbols such as those in the public section.
- **objcopy**: translates a file from one format to another, used in the embedded file to generate S-Records from ELF binaries.
- **objdump/readelf**: reads and prints out information from a binary file. readelf performs the same function; however, it can work only with ELF-formatted files.
- **ranlib**: a complement to ar. Generates an index of the public symbols in an archive to speed link time. Users can get the same effect by using ar -s.
- **size**: prints out the size of various components of a binary file.
- **strings**: extracts the strings from a binary, performing correct target host byte order translation. It's frequently used as the slacker's way of seeing what libraries a binary links to, as ldd doesn't work for cross-compiled programs: `strings <binary> | grep lib`.
- **strip**: removes symbols or sections, typically debugging information, from files.

Table 1. Pros and Cons of Most Frequently Used C Libraries

Library	Pros	Cons
glibc	The canonical C library; contains the greatest amount of support for all C features; very portable; support for the widest number of architectures.	Size; configurability; can be hard to cross-build.
uClibc	Small (but not the smallest); very configurable; widely used; active development team and community.	Not well supported on all architectures; handles only UTF-8 multibyte characters.
DietLibC	Small, small, small; excellent support for ARM and MIPS.	Least functionality; no dynamic linking; documentation.
NewLib	Well supported by Red Hat; best support for math functions; great documentation.	Smallish community; not updated frequently.

Finally, for the type of person who doesn't like the idea of wading through somebody else's build scripts when things don't work, building a cross-compiler by hand isn't as daunting a process as one would expect.

C Library

The C language specification contains only 32 keywords, give or take a few, depending on the compiler's implementation of the language. Like C, most languages have the concept of a standard library supplying common operations, such as string manipulation, and an interface to the filesystem and memory. The majority of the programming that happens in C involves interacting with the C library. As a result, much of the code in the project isn't written by the engineers, but rather is supplied by the standard libraries. Picking a standard library that has been designed to be small can have a drastic impact on the final size of the project.

Most embedded engineers opt for using a C library other than the standard GNU C Library, otherwise known as glibc, to conserve resources. glibc was designed for portability and compatibility, and as such, it contains code for cases not encountered or that can be sacrificed on an embedded system. One example is the lack of binary compatibility between releases of the library. Although glibc rarely breaks an interface once published, embedded standard libraries do so without any qualms.

Table 1 outlines the most frequently used C libraries, with the pros and cons of each.

Preprocessor and Compiler

These components perform only a small slice of the work necessary to produce an executable. The preprocessor, for languages that support such a concept, runs before the compiler proper, performing text transformations before the compiler transforms the input into machine code for the target. During the compilation process, the compiler performs optimizations as specified by the user and produces a parse tree. The parse tree is translated into assembler code, and the assembler uses that input to make an object file. If the user wants to produce an executable binary, the object file is then passed to the linker to produce an executable.

How It All Fits Together

After looking at all the components in a toolchain, the following section steps through the process GCC takes when compiling C source files into a binary. The process starts by invoking GCC with the files to be compiled and a parameter specifying output to be stored to thebinary:

```
armv51-linux-gcc file1.c file2.c -o thebinary
```

GCC is actually a driver program that invokes the underlying

compiler and binutils to produce the final executable. By looking at the extension of the input file and using the rules built in to the compiler, GCC determines what programs to run in what order to build the output. To see what happens in order to compile the file, add the `-###` parameter:

```
armv51-linux-gcc -### file1.c file2.c -o thebinary
```

This produces virtual reams of output on the console. Much of the output has been clipped, saving untold virtual trees, to make it more readable for this example. The first information that appears describes the version of the compiler and how it was built—very important information when queried “was GCC built with thumb-interworking disabled?”

```
Target: armv51-linux
Configured with: <the contents of a autoconf command line>
Thread model: posix
gcc version 4.1.0 20060304 (TimeSys 4.1.0-3)
```

After outputting the state of the tool, the compilation process starts. Each source file is compiled with the `cc1` compiler, the “real” compiler for the target architecture. When GCC was compiled, it was configured to pass certain parameters to `cc1`:

```
"/opt/timesys/toolchains/armv51-linux/libexec/gcc/
armv51-linux/4.1.0/cc1.exe" "-quiet" "file1.c"
  "-quiet" "-dumpbase" "file1.c" "-mcpu=xscale"
  "-mfloat-abi=soft" "-auxbase" "file1" "-o"
  "/cygdrive/c/DOCUME~1/GENESA~1.TIM/LOCALS~1/Temp/ccC39DVR.s"
```

Now the assembler takes over and turns the file into object code:

```
"/opt/timesys/toolchains/armv51-linux/lib/gcc/
armv51-linux/4.1.0/../../../../armv51-linux/bin/as.exe"
  "-mcpu=xscale" "-mfloat-abi=soft" "-o"
  "/cygdrive/c/DOCUME~1/GENESA~1.TIM/LOCALS~1/Temp/ccm4aB3B.o"
  "/cygdrive/c/DOCUME~1/GENESA~1.TIM/LOCALS~1/Temp/ccC39DVR.s"
```

The same thing happens for the next file on the command line, `file2.c`. The command lines are the same as those for `file1.c`, but with different input and output filenames.

After compilation, `collect2` performs a linking step and looks for initialization functions (called constructor functions, but not in the object-oriented sense) called before the “main” section of the program. `collect2` gathers these functions together, creates a temporary source file, compiles it and links that to the rest of the program:

```
"/opt/timesys/toolchains/armv51-linux/libexec/gcc/
armv51-linux/4.1.0/collect2.exe" "--eh-frame-hdr"
  "-dynamic-linker" "/lib/ld-linux.so.2" "-X" "-m"
  "armelf_linux" "-p" "-o" "binary" "/opt/timesys/
toolchains/armv51-linux/lib/gcc/armv51-linux/
4.1.0/../../../../armv51-linux/lib/crt1.o"
  "/opt/timesys/toolchains/armv51-linux/lib/gcc/
armv51-linux/4.1.0/../../../../armv51-linux/lib/crti.o"
  "/opt/timesys/toolchains/armv51-linux/lib/gcc/
```

```

➔armv5l-linux/4.1.0/crtbegin.o"
➔"-L/opt/timesys/toolchains/armv5l-linux/lib/
➔gcc/armv5l-linux/4.1.0" "-L/opt/timesys/
➔toolchains/armv5l-linux/lib/gcc/armv5l-linux/
➔4.1.0/../../../../armv5l-linux/lib"
➔"/cygdrive/c/DOCUME~1/GENESA~1.TIM/LOCALS~1/
➔Temp/ccm4aB3B.o" "/cygdrive/c/DOCUME~1/
➔GENESA~1.TIM/LOCALS~1/Temp/cc60Td3s.o"
➔"-lgcc" "--as-needed" "-lgcc_s" "--no-as-needed"
➔"-lc" "-lgcc" "--as-needed" "-lgcc_s" "--no-as-needed"
➔"/opt/timesys/toolchains/armv5l-linux/lib/
➔gcc/armv5l-linux/4.1.0/crtend.o" "/opt/timesys/
➔toolchains/armv5l-linux/lib/gcc/armv5l-linux/
➔4.1.0/../../../../armv5l-linux/lib/crtn.o"

```

There are some other nifty things in here that warrant pointing out:

1. Here's the option that specifies the dynamic linker to invoke when running the program on the target platform:

```
"-dynamic-linker" "/lib/ld-linux.so.2"
```

On Linux platforms, dynamically linked programs actually load by running a dynamic loader, making themselves a parameter of the linker, which does the work of loading the libraries into memory and fixing up the references. If this program isn't in the same place on the target machine, the program will fail to run with an "unable to execute program" error message. A misplaced linker on the target ensnares every embedded developer at least once.

2. These files contain the code before the programmer's entry point (typically main, but you can change that too) and handle things like initialization of globals, opening the standard file handles, making that nice array of parameters and other housekeeping functions:

- crtbegin.o
- crt1.o
- crti.o

3. Likewise, these files contain the code after the last return, such as closing files and other housekeeping work. Like the prior items, these are cross-compiled during the GCC build:

- crtend.o
- crtn.o

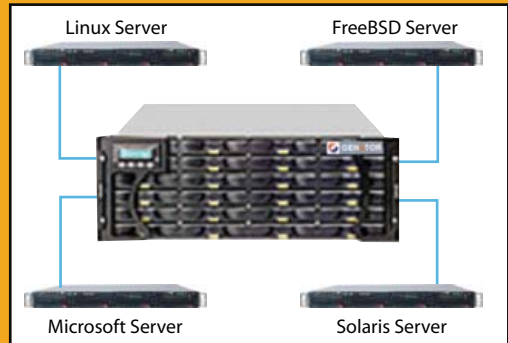
And, that's it! At the end of this process, the output is a program ready for execution on the target platform.

The spec File

Recall that GCC is a driver program that knows what program to invoke to build a certain output, which begs the question, "How does it know that?" This information that was built in to GCC when it was built is kept in the "specs". To see the specs, run GCC with the -dumpspecs parameters:



Linux - FreeBSD - x86 Solaris - MS etc.



GENSTOR STORAGE SOLUTIONS:

- Storage options - FC to SATA/SAS, FC to FC
- SAS to SAS/SATA, SCSI to SATA, SCSI to SCSI
- Exceptional Performance with Proven Reliability
- 24 TB in 4U with easy upgrade path
- Host Servers and Storage comes Pre-Configured with heterogeneous OS- Linux, * BSD, Solaris Microsoft etc.
- Fully redundant Storage solutions

Proven technology. Proven reliability

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

Quad Core Woodcrest



2 Nodes & Up to 16 Cores - in 1U

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

Features:

- 1U rack-optimized chassis (1.75in.)
- Up to 2 Quad Core Intel® Xeon® Woodcrest per Node with 1333 MHz system bus
- Up to 16 Woodcrest Cores Per 1U rackspace

Servers :: Storage :: Appliances

Genstor Systems, Inc.

780 Montague Express. #604
San Jose, CA 95131

www.genstor.com

Email: sales@genstor.com

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



```
armv5l-linux-gcc -dumpspecs
```

The console will fill with a few hundred lines of output. The spec file format evolved over years of development, and it's easier for the computer to read than for a person. Each line contains instructions for what parameters to use for a given tool. From the prior example, consider the command line for the assembler (with the path names removed for readability):

```
"<path>/as.exe" "-mcpu=xscale" "-mfloat-abi=soft"
  ➤ "-o" "<temppath>/ccm4aB3B.o" "<temppath>/ccC39DVR.s"
```

The compiler has the following in the specs for the assembler:

```
*asm:
%{mbig-endian:-EB} %{mlittle-endian:-EL} %{mcpu*:-mcpu=%*}
  ➤ %{march*:-march=%*} %{mapcs-*:-mapcs=%*}
  ➤ %{subtarget_asm_float_spec}
  ➤ %{mthumb-interwork:-mthumb-interwork}
  ➤ %{msoft-float:-mfloat-abi=soft}
  ➤ %{mhard-float:-mfloat-abi=hard} %{mfloat-abi=*}
  ➤ %{mfpu=*} %{subtarget_extra_asm_spec}
```

This line uses some familiar constructs explained below. Adequately discussing the minutiae of the spec file would require an article series in itself.

- `*asm`: this line tells GCC the following line will override the internal specification for the asm tool.
- `%{mbig-endian:-EB}`: the pattern `%{symbol:parameter}` means if a symbol was passed to GCC, replace it with parameter; otherwise, this expands to a null string. In our example, the parameter `-mfloat-abi=soft` was added this way.
- `%{subtarget_extra_asm_spec}`: evaluate the spec string `%(specname)`. This may result in an empty string, as it did in our case.

Most users don't need to modify the spec file for their compiler; however, frequently engineers who inherit a project need to have GCC recognize nonstandard extensions for files. For example, assembler source files may have the extension, `.arm`; in this case, GCC won't know what to execute, as it doesn't have a rule for that file extension. In this case, you can create a spec file containing the following:

```
.arm:
@asm
```

and use the `-specs=<file>` to pass that to GCC, so that it will know how to handle files with the `.arm` extension. The spec file on the command line will be added to the internal spec file after it has been processed.

Tips and Tricks of the Trade

The following tips and tricks should be, if they haven't already, stashed on the crib sheet of engineers who work with GCC.

Force GCC to use an alternate C library:

```
armv5l-linux-gcc -nostdlib -nostdinc -isystem
  ➤<path to header files> -L<path to c library>
  ➤-l <c library file>
```

This tells GCC to ignore everything it knows about where to find header files and libraries and instead uses what you tell it. Most alternate C libraries provide a script that performs this function; however, some projects can't use the wrapper scripts, and other times, when experimenting with several versions of a library, the flexibility and control of specifying this information directly is necessary.

Mixed assembler/source output:

```
armv5l-linux-gcc -g program.c -o binary-program
armv5l-linux-objdump -S binary-program
```

This is the best way to see exactly what GCC generated in relation to the input code. Doing the compilation with several different optimization settings shows what the compiler did for the given optimization. Because embedded development pushes the processor-support envelope, being able to see the generated assembler code can be instrumental in proving a defect in GCC's support for that processor. In addition, engineers can use this to validate that the proper instructions are generated when specifying processor-specific optimizations.

List predefined macros:

```
armv5l-linux-gcc -E -dM - < /dev/null
```

An invaluable tool for doing a port, this makes clear what GCC

Resources

uClibc, a replacement for the GNU C Library, optimized for size: www.uclibc.org.

dietlibc, another replacement for GNU C, the smallest of the group: www.fefe.de/dietlibc.

NewLib, a Red Hat-supported project for a minimal C library: sourceware.org/newlib.

GCC Internals—information about the guts and construction of GCC; it's very well written and a great guide for those curious about how GCC works: gcc.gnu.org/onlinedocs/gccint.

binutils—architecture-specific tools that smooth the way for development: www.gnu.org/software/binutils.

`info gcc`, from your command line, provides in-depth information about end-user-related aspects of GCC.

crosstool, a tool for building GCC cross-compilers, now the canonical way for doing so, is very easy to use: www.uclibc.org.

The Definitive Guide to GCC by Bill von Hagen—a great book covering all aspects of how to use GCC.

macros will be set automatically and the value. This will show not only the standard macros, but also all the ones set for the target architecture. Keeping this output and comparing it to a newer version of GCC can save hours of work when code fails to compile or run due to changes.

List dependencies:

```
armv5l-linux-gcc -M program.c
```

Formally, this command creates a separate make rule for each file on the command line showing all dependencies. The output is indispensable when trying to track down problems related to what header files a source file is using and tracking down problems related to forcing GCC to use an alternate C library. Deeply nested header files are both unavoidable and incredibly useful in any nontrivial C project and can consume hours when trying to debug. Using -MM instead of -M will show only nonsystem dependencies—useful noise reduction when the problem resides in the project files alone.

Show internal steps:

```
armv5l-linux-gcc -### program.c
```

This article already uses this command to make GCC show what

steps occur internally to build a program. When a program isn't compiling or linking properly, using -### is the fastest route to see what GCC is doing. Each command is on its own line and can be run individually, so:

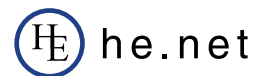
```
armv5l-linux-gcc -### program.c &> compile-commands
```

will produce a file compile—commands that the user can mark as executable and run a line at a time to pinpoint the exact cause of a problem.

Wrapping Up

GCC is a deceptively powerful, complex tool. The developers have created software that “does the right thing” with minimal information from the user. Because it works so well, users frequently forget to spend the time to learn about GCC’s capabilities. This article scratches the surface; the best advice is to read the documentation and invest a little time each day to learn how this tool always can do more than expected.■

Gene Sally has been working with all facets of embedded Linux for the last seven years and is cohost of LinuxLink Radio, the most popular embedded Linux podcast. Gene can be reached at gene.sally@timesys.com.



IP Transit Gigabit Ethernet

Run BGP+IPv6+IPv4

Colocation Full
Cabinet

Holds up to 42 1U
servers

\$400/month

\$5/Mbps

Full 100 Mbps
Port

Full Duplex

\$2,000/month

Order Today!

email sales@he.net or call 510.580.4190

he.net/ip_transit.html

GPG-Based Password Wallet

Keep your passwords safe in an encrypted file. CARL WELCH

Like many Internet addicts, I have way too many user name/password accounts to remember: accounts on social-networking sites, rarely used logins at work, on-line banking and so on. One solution to this problem is to use the same user name and password everywhere, but that's clearly not safe; if people get a hold of your account information in one place, they own all your other accounts too.

I wanted a relatively safe, flexible and easy way to store passwords and other useful confidential information. I also wanted it to be easily accessible, which meant that I'd like to get at it over a text-only SSH connection. And, I wanted it to be something that could move around from machine to machine without too much trouble.

A few months ago, I saw an article by Duane Odom on [linux.ocm](#) about a shell script that uses GPG to encrypt and decrypt a text file containing the user's list of passwords (or any kind of text). I liked this approach, as it met the following requirements:

1. It stores passwords in a well-encrypted text file (protected by a master password). The text file could contain anything and be formatted any way I want.
2. The entire interface is text (an ncurses password interface, followed by `less` or a text editor like `vim`), so you can access it over a nongraphical SSH session (see the [Accessing Your Password Wallet from the Computer at Your Friend's House](#) sidebar).

3. The script is built on standard utilities common to most Linux distributions (`gpg` and `dialog`).

Although I liked the way the original script worked, I wanted to add several features. So I made some alterations to the original, and the result is shown in Listing 1.

It's pretty easy to install; simply save the script somewhere in your `$PATH` and make it executable. Then, you just need to tell it where your encrypted password file should be. There are three ways to do this:

1. Set the `$WALLET_FILENAME` environment variable.
2. Set `$WALLET_FILENAME` in `~/.walletrc`.
3. Specify the filename with the `-c` command-line option.

The second method (which overrides the first method) is my preference—I have the following line in `~/.walletrc`:

```
WALLET_FILENAME=~/.docs/wallet.gpg
```

But, if I needed to use a different wallet file, I could override either of the first two methods with the command-line option by

Accessing Your Password Wallet from the Computer at Your Friend's House

One of things that has made wallet useful to me is the ability to reach it from anywhere. Here are a few hints for enabling SSH access to your broadband-connected Linux box at home.

Rather than try to memorize your computer's IP address (which may change unexpectedly anyway), you could sign up for a free dynamic DNS service like DynDNS. This lets you choose a memorable hostname for your computer, something like `carlslinuxbox.dyndns.org`. Some of these services want you to update your DNS record periodically (I think `dyndns.org` wants that to happen at least once a month). Rather than do that by

hand, you can run an auto-updater (like `inadyn`) in `cron`. Be careful when setting the update frequency—some dynamic DNS services suspend your account if you update too often (read the fine print).

If you're going to let the Internet talk to SSH on your Linux box, there are a few things you can do to make that a bit more secure. I recommend disabling the `PermitRootLogin` option in the `sshd_config` file. You also may want to run SSH on a nonstandard port using the `Port` option in `sshd_config`. If the script kiddies find SSH running on port 22, they'll throw a bunch of user names and passwords at it trying to

break in. Running SSH on a port other than 22 discourages this sort of thing to a large degree. And, make sure your firewall allows access to whatever port you use. Finally, if your computer is behind a consumer cable/DSL router, you'll have to configure the router to forward SSH traffic to your Linux box.

With those things done, next time you're at a friend's house, you could jump on a computer, download an SSH client (such as `putty`) and SSH to your Linux box (remembering to tell the SSH client your dynamic DNS hostname and the port number on which you're running SSH).

calling the script like this:

```
wallet -c ~/docs/other_wallet.gpg
```

wallet defaults to its read-only mode, in which it displays the decrypted version of your wallet file using less. But, if you include the `-e` command-line option (edit mode), the script decrypts your wallet file to

Listing 1. wallet Script

```
1 #!/bin/bash
2
3 . ~/bin/functions
4 is_installed gpg
5 is_installed dialog
6 is_installed mktemp
7 is_installed basename
8
9 if [ -f ~/.walletrc ]; then
10     . ~/.walletrc
11 fi
12
13 if [ -z $VISUAL ]; then
14     VISUAL=vi
15 fi
16
17 EDIT_PWFILE=0
18 while getopts 'ec:' OPTION
19 do
20     case $OPTION in
21         e) EDIT_PWFILE=1;;
22         c) WALLET_FILENAME="$OPTARG";;
23         ?) printf "usage: %s [ -e ] [ -c encrypted_file ]\n" \
24             $(basename $0) >&2
25             exit 1
26         ;;
27     esac
28 done
29 shift $((OPTIND - 1))
30
31 if [ -z "$WALLET_FILENAME" ]; then
32     echo "need the encrypted file specified by WALLET_FILENAME"
33     echo "(in ~/.walletrc or the envariable) or with the -c option"
34     exit 2
35 fi
36
37 if [ ! -f $WALLET_FILENAME ]; then
38     echo "$WALLET_FILENAME doesn't exist--attempting to create..."
39     echo "(you'll need to give gpg a master password)"
40     mkdir -p $(dirname $WALLET_FILENAME)
41     TMPFILE=$(mktemp /tmp/wallet.XXXXXX)
42     gpg -c -o $WALLET_FILENAME $TMPFILE
43     rm -f $TMPFILE
44     EDIT_PWFILE=1
45 fi
46
47 if [ $EDIT_PWFILE -eq 1 ]; then
48     is_installed $VISUAL
49 fi
50
51 # prompt the user for the password
52 PASSWORD=$( dialog --stdout --backtitle "Password Wallet" \
53     --title "Master Password" --clear --passwordbox \
54     "Please provide the master password." 8 40 )
55 if [ $? -ne 0 ]; then
56     echo "Failed to acquire master password"
57     exit 4
58 fi
59 if [ -z $PASSWORD ]; then
60     echo "Password is required"
61     exit 8
62 fi
63
64 # if we're not editing the file, just display it and quit
65 if [ $EDIT_PWFILE -eq 0 ]; then
66     echo $PASSWORD | gpg --decrypt --passphrase-fd 0 \
67         $WALLET_FILENAME | less
68     clear
69     exit 0
70 fi
71
72 # set up the directory in which the unencrypted wallet file
73 # will be edited
74 TMPDIR=$(mktemp -d /tmp/wallet.XXXXXX)
75 CLEARTEXT_WALLET_FILENAME=$TMPDIR/wallet
76
77 # try to ensure that cleartext wallet file is deleted,
78 # even after unexpected terminations
79 trap "{ rm -rf $TMPDIR; }" 0 1 2 5 15
80
81 # decrypt the password wallet--an error here probably means
82 # the user typed the wrong password to decrypt the wallet
83 echo $PASSWORD | gpg -o $CLEARTEXT_WALLET_FILENAME \
84     --passphrase-fd 0 \
85     $WALLET_FILENAME &> /dev/null
86 case $? in
87     0)
88         # decryption succeeded, so open the wallet in the editor
89         # and then re-encrypt it when the editor closes
90         mv $WALLET_FILENAME ${WALLET_FILENAME}.bak
91         $VISUAL $CLEARTEXT_WALLET_FILENAME 2> /dev/null
92         echo $PASSWORD | gpg -c -o $WALLET_FILENAME \
93             --passphrase-fd 0 \
94             $CLEARTEXT_WALLET_FILENAME &> /dev/null
95         if [ $? -eq 0 ]; then
96             clear
97         else
98             LAST_RESORT_FILENAME=$(mktemp ~/wallet.XXXXXX)
99             cp $CLEARTEXT_WALLET_FILENAME $LAST_RESORT_FILENAME
100            chmod 600 $LAST_RESORT_FILENAME
101            echo "gpg failed to encrypt your password wallet: I have"
102            echo "tried to put a CLEARTEXT copy of your wallet at"
103            echo $LAST_RESORT_FILENAME
104            exit 16
105        fi
106        exit 0;;
107     ?)
108        echo "error condition detected (invalid password?)"
109        exit 32;;
110     esac
```

a temporary location and opens it in a text editor (the script defaults to using vi, but you can set the \$VISUAL variable in the environment or in your ~/.walletrc file). When you close the editor, wallet encrypts the file and saves it to the original location.

The first time you run wallet, you won't have a wallet file, so wallet creates it for you and runs in edit mode.

How It Works

Let's dig in to the script to see how it works. The first thing it does is use the dot operator to source a file called functions, which appears as shown in Listing 2. Having wallet source an external file (with the dot operator) is essentially equivalent to inserting the contents of the sourced file (~/.bin/functions) at line 3 of wallet. Doing it this way allows other scripts to use the same code (a code library for shell scripts).

Listing 2. functions File

```
is_installed() {
    PROGRAM=$1

    PATHNAME=$( type $PROGRAM 2> /dev/null )
    if [ -z "$PATHNAME" ]; then
        echo "cannot locate $PROGRAM in path"
        exit 1
    fi
}
```

The functions file includes a function called is_installed, which uses the bash built-in type to see whether a program is installed. If is_installed doesn't find the program in your \$PATH, is_installed prints an error message and calls exit, which terminates wallet. So, if you run wallet and it quits with an error like "cannot locate dialog in path", you probably haven't installed the dialog package. Use your distribution's package management system (yum, apt-get, whatever) to install dialog and try again.

Input Validation

Lines 18 through 28 of the wallet script parse the command-line arguments using the getopts bash built-in. The while loop loops through the options specified by the string ec:. This means that wallet can accept the -e and -c options, and that the -c option requires an argument. As the while loop moves through the command-line arguments, the current option is assigned to the variable \$OPTION, and any argument to the current option is assigned to the variable \$OPTARG. Any unrecognized option results in an error message, and wallet exits. After the while loop completes, it's important to reset the \$OPTIND variable (this is necessary after any getopts call).

Running wallet the First Time

Lines 37 through 45 of the wallet script verify that the encrypted file exists, and create the file if it doesn't exist already. The -f test

checks to see whether \$WALLET_FILENAME exists as a normal file. If not, the test fails, and wallet assumes you are running wallet for the first time and that wallet needs to set up the working environment. wallet uses the command substitution syntax for creating the directory in which the encrypted file should exist (line 40):

```
mkdir -p $( dirname $WALLET_FILENAME )
```

The command inside the \$(...) runs first, and the result becomes the argument to mkdir. The dirname command returns the encrypted file's directory, and mkdir -p creates that directory (and any necessary parent directories).

Next, wallet needs to create the encrypted file (even though the unencrypted version will be empty). Line 41 uses mktemp to create an empty file in /tmp whose name ends in six randomly chosen characters. mktemp prints the name of the file it creates, so running this in a command substitution shell and assigning the result to \$TEMPFILE puts the name of the temporary file in \$TEMPFILE.

Now we see the first use of gpg. Line 42 uses gpg to encrypt the (empty) temporary file (\$TEMPFILE) via symmetric encryption (gpg's -c option) and to write the encrypted file to \$WALLET_FILENAME.

Listing 3. Password Generator Script

```
#!/bin/bash

. ~/.bin/functions
is_installed openssl

DIGEST="sha1"
RULER=0
DASH_N=""
while getopts 'mrn' OPTION
do
    case $OPTION in
        m) DIGEST="md5";;
        r) RULER=1;;
        n) DASH_N="-n";;
        ?) printf "usage: %s [ -m ] [ -r ]\n" $( basename $0 ) >&2
            exit 2
            ;;
    esac
done
shift $(( $OPTIND - 1 ))

if [ ! -z $DASH_N ]; then
    RULER=0
fi

DD=$( dd if=/dev/urandom bs=1k count=1 2> /dev/null \
| openssl dgst -$DIGEST )
echo $DASH_N $DD
if [ $RULER -eq 1 ]; then
    echo ' 5| 10| 15| 20| 25| 30| 35| 40|'
fi
```


wallet then deletes the temporary file. Because this is the first time wallet has run, it assumes that edit mode is appropriate and sets the \$EDIT_PWFILE flag.

Prompting the User for the Master Password

Line 52 uses the command substitution trick again, this time to prompt the user for the master password (used to encrypt the wallet file). The dialog man page describes the many ways that scripts using dialog can retrieve input from the user. This example uses dialog to create a simple password box. The `--stdout` option tells dialog to print the user's input (the master password) to standard output, so that it may be assigned to \$PASSWORD.

Line 55 inspects the bash variable \$?, which contains the exit code of the previous process (dialog, in this case). The convention is that an exit code of 0 indicates success (and wallet follows this convention in its own exit calls). If \$? differs from 0 on line 55, this indicates that dialog encountered an error, and wallet terminates with an error message.

Read-Only Mode

If \$EDIT_PWFILE is 0 (line 65), then wallet is running in read-only mode:

```
echo $PASSWORD | gpg --decrypt --passphrase-fd 0
↳ $WALLET_FILENAME | less
```

This tells gpg to decrypt \$WALLET_FILENAME and to read the password from standard input (fd 0). Piping \$PASSWORD into gpg enables gpg to decrypt the wallet file without interactively asking the user for the master password. The output (the decrypted wallet file) is printed to standard output, which is piped into less, allowing the user to page through the passwords, run searches and so on. When the user closes less, wallet clears the screen and exits.

The rest of the script assumes that \$EDIT_PWFILE is nonzero (that wallet is running in edit mode).

Edit Mode

In edit mode, wallet needs to decrypt the wallet file, open the decrypted file in a text editor, and then encrypt the edited file back to the original location. Line 74 uses mktemp to create a temporary directory, into which the wallet file will be decrypted. Line 75 sets \$CLEARTEXT_WALLET_FILENAME to be the name of a file inside the temporary directory.

Line 79 runs trap, a bash built-in. The first argument to trap is a command, and this is followed by a list of signals (for example, if someone runs `kill` on wallet). If wallet receives any of these signals after line 79, wallet will run the trapped command (deleting the decrypted wallet file) prior to exiting. This is an attempt to ensure that the decrypted file isn't left sitting around if wallet terminates unexpectedly.

Line 83 is like what we saw in read-only mode, with the addition of the `-o` option to gpg. This instructs gpg to write the decrypted file to \$CLEARTEXT_WALLET_FILENAME.

If gpg's exit code was 0, wallet renames the encrypted wallet file with a `.bak` extension (thus preserving a copy, in case something goes wrong) and opens the decrypted file in the text editor \$VISUAL. After the editor exits, wallet tells gpg to encrypt the

Password Generator

Listing 3 shows a short shell script that generates very random, impossible-to-remember passwords—perfect for storing in your wallet. `mkpass` dumps a kilobyte of random data into a digest algorithm to produce an ASCII password. By default, `mkpass` uses the SHA1 digest algorithm, but it can use MD5 if you supply `mkpass's -m` command-line option. And, if you give the `-r` option, `mkpass` prints a ruler below the password (useful if you need or want a password of a particular length).

If you're a vim user, try adding the following line to your `~/.vimrc` file:

```
map \mkpass i <CR><ESC>k$:r!~/bin/mkpass -n<CR>kJJ
```

Now when you're running vim (like when you're using wallet in edit mode), typing `\mkpass` in command mode will insert a password at the cursor location.

edited plain-text file at \$CLEARTEXT_WALLET_FILENAME and to write the encrypted wallet file back to \$WALLET_FILENAME. A nonzero exit status from this gpg call means that something went wrong in re-encrypting the wallet file, so wallet makes a copy of the plain-text file in your home directory and prints an error message.

Conclusion

wallet is a bash script for managing a password wallet. It's written to be usable over a text-only interface. Hopefully, this description of the code has helped you add an item or two to your bag of scripting tricks. ■

Carl Welch is a Web developer and Linux system administrator. He enjoys science fiction, is ambivalent to dentists and dislikes standard light switches. He maintains the lamest blog on planet Earth at mbrisby.blogspot.com.

Resources

"How to create a command-line password vault" by Duane Odom: www.linux.com/feature/114238

DynDNS Dynamic DNS: www.dyndns.org

inadyn Dynamic DNS Updater: inadyn.ina-tech.net

putty SSH Client: www.chiark.greenend.org.uk/~sgtatham/putty

Security in Qtopia Phones

Trolltech's Qtopia SXE takes a stab at making open-source phones more secure.

LORN POTTER

No one wants an insecure system, especially on a mobile or VoIP phone. Both users and operators want to feel confident that their phones won't be used secretly to send thousands of spam messages or viruses or to transfer huge files. Linux in the mobile space opens doors to everyone—including developers of malicious code. A locked-down and secure system does not necessarily mean the source code is closed.



Figure 1. Qtopia's Home Screen

The last thing people want on their phones is a malware application that secretly sends their details somewhere or launches a DOS attack using their costly GPRS account. Together with the Linux Intrusion Detection System (LIDS), Trolltech's Safe Execution Environment (SXE) delivers a safe environment in which to allow untrusted applications to be executed. Without SXE and LIDS, an unsuspecting user could install an unknown package. This could launch Qtopia's QCop, which handles Qtopia's interprocess communication (IPC) with LD_PRELOAD set to its own libc library. This means that its own code is loaded instead of the known libc in the system,

which could have disastrous results on the user's data, or worse, disrupt emergency communications.

Trolltech has an answer.

Trolltech recently announced the GPL version of Qtopia that includes SXE, as well as telephony libraries needed for GSM and VoIP-enabled phones. Trolltech has spent many person-hours developing SXE to help ensure that both operators and users are confident about installing native compiled applications. I say person-hours here because the lead developer for much of SXE's life was Sarah Smith—Trolltech's first female developer.

SXE is a little like a firewall. It prevents applications from accessing unauthorized services and hardware through domain controls. It goes beyond just plain sandboxing applications, which can blindly deny programs access to system resources. It applies policy rules for each program component and IPC. Qtopia applications send an IPC or QCop message when they want to open a window or send an SMS.

Upon installation, an application specifies what security domains are needed to function and is sandboxed by Qtopia. If the program is executed and then tries to access services beyond what it has been awarded, a security breach is logged, and the application is terminated and disabled. A dialog and SMS message are issued to the user regarding the breach. LIDS can complete the safe environment by controlling access to hardware, system-level services, programs and files.

The Qtopia Greenphone is an example of a working SXE and LIDS implementation, and this article discusses Qtopia version 4.3.0. The Qtopia open-source version, announced recently for the FIC's open-source Neo phone, also would benefit from SXE and a LIDS-enabled kernel.

Application Development

An SXE application starts with a developer creating a Qtopia application and packaging it in a Qpkg, which is based on the Itsy Package, but has a few security issues resolved. Namely, Qtopia does not allow an

Note:

Although sales of the Greenphone have been discontinued, support and development for it has not—neither has development of SXE for Qtopia. In this article, I use the Greenphone mainly as an example.

Trolltech is advising people interested in an open Qtopia phone to purchase an FIC Neo 1973, as it has ported Qtopia to that device and plans on supporting an SDK in the same light as the Greenphone SDK, scheduled to be released with the 4.3.1 release.

Advertiser Index

For advertising information, please contact our sales department at 1-713-344-1956 ext. 2 or ads@linuxjournal.com.
www.linuxjournal.com/advertising

Trolltech recently announced the GPL version of Qtopia that includes SXE, as well as telephony libraries needed for GSM and VoIP-enabled phones.

application install to run arbitrary scripts, but also, the package must specify which domains it needs access to in order to run. Qtopia then allows (or denies) that package only those domains that it requests.

An SXE domain is simply a name for an allowed access rights policy. Some of the default domains and their access rights in Qtopia are:

- docapi: user documents.
- pim: Personal Information Management data.
- window: graphic display.
- graphics: full-screen graphics display.
- net: access to WAP, GSM and GPRS.

Qtopia uses many more domains, and some of them, such as the base domain, should never be allowed access by any untrusted application. Operators or integrators can use the default Qtopia domains or create their own.

The third-party developer then specifies, in the package control file, in which domains the application needs to function. Much like a legacy ipkg, a qpk is simply a gzipped tarball of the file structure where the application lives, a desktop file much like those used in KDE and GNOME, plus a control file that specifies parameters, such as the application's name, maintainer, domain, file size and so forth.

The Greenphone SDK makes this easy with the use of the gph script, which can configure, compile, build the package and install or run it on the Greenphone. The developer needs to know only which domains the application is going to use. Starting a debug build of Qtopia in SXE_DISCOVERY_MODE, with SXE logging turned on, can help log these domain requests and subsequent denials. There is a significant performance decrease while running in discovery mode. It is only for the debugging process and not a final release. The developer then can add the appropriate domains to the application's .pro file.

Installation Time

After configuring the Package Manager to see the feed server, Qtopia reads a plain-text file named packages.list on the server. This file contains a list of all the packages available on the server, the domains the package is requesting, as well as the description, name, maintainer, size, license and md5sum of every package.

When users want to install a new package, they select it from a list. Users then are prompted with a dialog containing the specific domains that the application is asking to access (Figure 2). Users have a choice whether or not to install. The package then is downloaded to temporary storage, installed and sandboxed. By default, the untrusted packages live in /home/Packages, with the md5sum used as a directory name—for example,

Advertiser	Page #	Advertiser	Page #
ABERDEEN, LLC www.aberdeeeninc.com	C2	MICROWAY, INC. www.microway.com	C4, 59
2008 ANNUAL WEB SERVICE/SOA www.webservicesonwallstreet.com	35	MIKRO TIK www.routerboard.com	5
ASA COMPUTERS www.asacomputers.com	63, 85	O'REILLY MEDIA www.oreillynet.com	15
AVOCENT CORPORATION www.avocent.com/remotecontrol	1	POLYWELL COMPUTERS, INC. www.polywell.com	7
CARLNET www.carl.net	83	THE PORTLAND GROUP www.pggroup.com	11
CORAID, INC. www.coraid.com	29	RACKSPACE MANAGED HOSTING www.rackspace.com	C3
ETECH/GSP SHOW conferences.oreillynet.com/et2008	53	R CUBED TECHNOLOGIES www.rcubedtech.com	91
EMAC, INC. www.emacinc.com	32	R1SOFT, INC. www.r1soft.com	47
EMPERORLINUX www.emperorlinux.com	3	SCALE 07 www.socallinuxexpo.org	87
FAIRCOM www.faircom.com	23	SDG SYSTEMS www.sdgsystems.com	31
FLORIDA LINUX SHOW www.floridalinuxshow.com	33	SERVERS DIRECT www.serversdirect.com	9
GENSTOR SYSTEMS, INC. www.genstor.com	73	SILICON MECHANICS www.siliconmechanics.com	27, 45
HPC SYSTEMS www.hpcsystems.com	65	SXSW FESTIVALS AND CONFERENCES www.sxsw.com	69
HURRICANE ELECTRIC www.he.net	75	TECHNOLOGIC SYSTEMS www.embeddedx86.com	10
INFITECH www.infi-tech.com	38, 39	TRI-D SYSTEMS, INC. www.tri-dsystems.com	51
LOGIC SUPPLY, INC. www.logicsupply.com	6	WILEY TECHNOLOGY PUBLISHING www.wiley.com	19

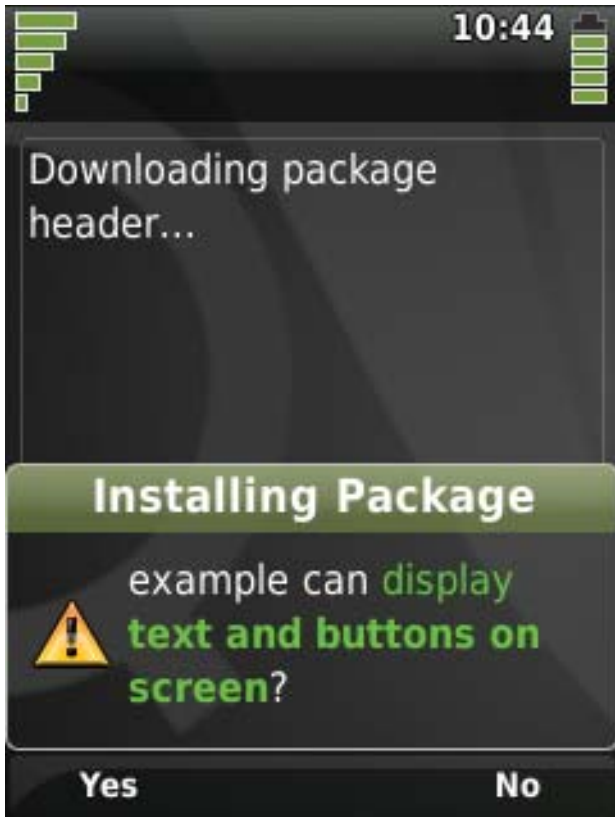


Figure 2. Package Installation

home/Packages/1e67fa93917fedb17f575fe0f2ee2cd8/bin/screenshot.

The Packages directory has a file structure, such as `bin/lib/pics/`, that symlinks to where the real binaries live. These symlinks use the md5sum in the name, such as `1e67fa93917fedb17f575fe0f2ee2cd8_screenshot` → `../1e67fa93917fedb17f575fe0f2ee2cd8/bin/screenshot`.

This file path is known to Qtopia, so it can find your shiny new application, and then adds it to the main applications list. This information now lives in the Qtopia content database. In previous versions of Qtopia, all this data simply lived in the filesystem, and Qtopia scanned to find the applications. The Package Manager then runs the `sxe_sandbox` script to create the LIDS rules for this application.

Runtime

Users start an untrusted application by clicking on its icon from the main menu. In Qtopia versions previous to 4.3.0, the untrusted and installed applications were accessible from the Installed Packages appli-

Namely, Qtopia does not allow an application install to run arbitrary scripts, but also, the package must specify which domains it needs access to in order to run.

cation. To make sure an application tries to access only the domains it was granted, Qtopia monitors service access requests with SXEMonitor. If the application tries to access something it did not initially request, such as the `phonemmm` domain, a breach is registered (Figure 3). The application is terminated, and Qtopia alerts the user with a dialog. It also, however, sends the user an SMS message directly to the Messages inbox. If this application continues to create breaches, Qtopia disables the program completely.

LIDS plays an integral part in all this. SXE works together with LIDS policies to make sure files that should not be accessible are not accessed. You must have LIDS enabled in the kernel to take advantage of SXE. The Mandatory Access Control (MAC) system in LIDS controls lower-level filesystem access. Without it, Qtopia can deny applications access to Qtopia services and tasks in the domain policies, but there would be nothing stopping an application from changing those access rights to something more advantageous for a malicious application.

A number of script templates ship with Qtopia, which live in `etc/sxe_qtopia`, that help with the creation of LIDS rules during both the root filesystem creation and package installation. The LIDS-enabled Greenphone writes these policy rules during the first boot after a flash of a system update. An operator can, of course, do this to the filesystem before deployment.

When integrators create a new application or service, they need to add them to Qtopia's `etc/sxe.profiles` file. This file contains a list of domains and the services and QCop messages associated with

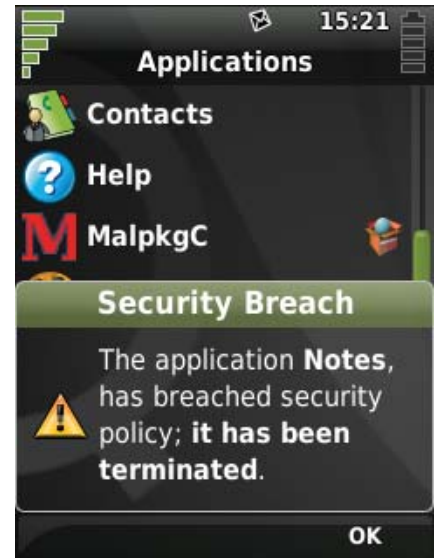


Figure 3. Security Breach Alert



Figure 4. Installed Package in Main Menu

them. It is processed by Qtopia at install time to create SXE policies. Integrators also might need to add it to the Package Manager's source code, so it can display the domain's verbose characteristics to the user. This helps users make at least a knowledgeable choice as to whether to install the package.

Qtopia.net has two feeds set up with simulated malware packages to test, for both the 4.3.0 Greenphone (qtopia.net/packages/feed/4.3/greenphone) and its SDK (qtopia.net/packages/feed/4.3/sdk). There, you can get the latest Greenphone SDK to try out yourself (Figure 6).

To enable a LIDS kernel, download the LIDS patches from the LIDS Web site, build the patched kernel, build the LIDS filesystem and configure the policy scripts. Qtopia comes with scripts to help define LIDS policies based on domains. For example, the script

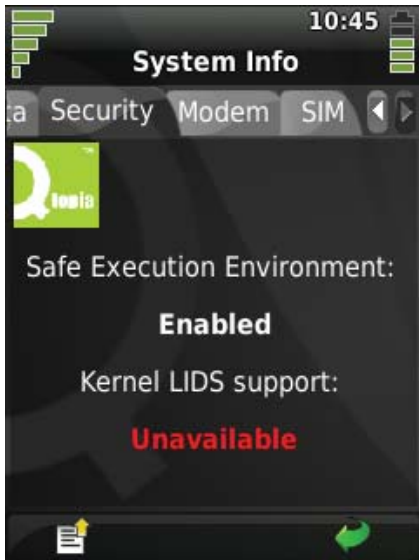


Figure 5. Security Info Showing SXE and LIDS Status



Figure 6. List of Fake Malware Packages on Qtopia.net Feed

SXE works together with LIDS policies to make sure files that should not be accessible are not accessed.

`etc/sxe_domains/sxe_qtopia_bluetooth` creates a LIDS rule like this:

```
lidsconf -A POSTBOOT -s ${BIN} -o LIDS_SOCKET_CREATE -j ENABLE
```

for applications that are granted access to the Bluetooth domain.

I won't go into the gory details of creating LIDS policies, but more information can be found at the LIDS Web site (lids.org) and in the Trolltech documentation (doc.trolltech.com/qtopia4.3/sxe.html).

SXE and LIDS can make your Linux phone more enjoyable and worry-free by giving you the confidence that untrusted applications you download will do only the things they are allowed to do. ■

Lorn Potter works for Trolltech as a Software Engineer in the Systems Group, MES. He is an American who lives in sunny Brisbane, Australia, with his Australian wife, two-year-old son and newborn daughter. He was previously the Qtopia Community Manager and has been developing open-source software for seven years. He also has worked as a musician, sound engineer and snow ski-bum in Colorado, Alaska and Upper Michigan.

DEDICATED SERVERS
Total Linux Support

Trustix suSE

carinet

STARTING AT 1GB DDR400 RAM — 160GB SATA2 HDD
 INTEL BOARDS & CPUS
60 \$ 100MBPS DEDICATED CISCO PORT
 1300GB THROUGHPUT INCLUDED

CARI.NET/LJ
 888.221.5902

Separate the Static from the Dynamic with Tomcat and Apache

Hosting servlets via Apache, mod_jk, Tomcat, mod_ssl and a few rewrite rules.

ALAN BERG

Hosting multiple Java Web-enabled applications with Apache/SSL in combination with Tomcat is potentially highly detailed. Separating the dynamic from the static content requires URL rewriting and aliases. This article discusses one viable configuration.

I describe the basics of how to host multiple Java Web applications using a pure Apache project approach. In other words, I explain how to apply Apache, mod_ssl, some rewrite rules and the Tomcat Servlet container to gain control of a consistent and viable production environment. In real life, I am a more-than-a-little-busy developer, and one of my more-recent tasks was to define and implement a structure to host a complex database-intensive Web-enabled searchable publication mechanism through the life cycle. I condense the experience gained and explain the most relevant details here.

The basics of placing an Apache Web server in front of multiple Tomcat servers is explained in an article by Daniel McCarthy on the *Linux Journal* Web site (see Resources). I take this article somewhat further by adding the ability to provide secure communication via SSL and show how to optimize performance by separating dynamic content, such as JSP pages, from static content, such as HTML and images. Further security issues also are nodded at briefly.

Preparations

The following preparations are for those who want to generate a working instance of the infrastructure mentioned. This infrastructure involves a locally configured Apache server running with two Tomcat instances, all being referenced from a Web browser via different loopback (127.0.0.x) addresses. This article is still worth reading without following through with the recipe.

I assume that the following have been installed: Apache 1.3x Web server, mod_ssl, mod_jk and two instances of a Tomcat 5.5.x server, one running the ajp1.3 connector on the standard port of 8009 and the shutdown port of 8005, and the other on port 8019 and 8015. I have chosen a plain-old stable and reliable Apache 1.3.x server over an Apache 2.x version on the principle that you shouldn't fix what isn't broken. At the Institutes for which I have been responsible, during the past few years they have run Apache 1.3.x without issue, the system administrators have built up their knowledge, and the systems are maintained and patched to the highest levels and snugly sit in the maturity section of the Web server's life cycle. The same applies for the choice of mod_jk over mod_jk2. In fact, mod_jk2 development has been discontinued due to the complexity of configuration.

If you have a Debian-based Linux distribution, to install the Apache server without compiling, try the following:

```
sudo apt-get install apache
sudo apt-get install libapache-mod-jk
sudo apt-get install libapache-mod-ssl
```

You should now have a running Apache instance with the configuration files sitting under /etc/apache.

For the Tomcat servers, you have two choices. The first is to use one instance of the binary and then two instances of the configuration, and then run a startup script that applies the unique instance of the binary with different configurations. The second choice is to use two copies of the Tomcat server and modify the server.xml file. The advantage of the first approach is the avoidance of replication of executable code. However, this is nearly always a false economy. The second approach has advantages for complex environments where you want to host different versions of Tomcat servers. The second approach is more relevant for Application Service Providers that have multiple customers. A division exists between code that is written for Java 1.5 that runs natively in Tomcat 5.5 (without installing the 1.4 compatibility package) and Java 1.4 that runs in Tomcat 5. Furthermore, the Servlet implementation is more up to date the newer the Tomcat version. Due to the current velocity of change, software that is hosted for more than a year can be considered legacy, so there always will be a demand for the use of older but still reliable servers.

Next, we want to test only on the loopback addresses with no packets reaching the network. This can be achieved by modifying the /etc/hosts file to something similar to:

```
127.0.0.10 bronze_a
127.0.0.11 silver
127.0.0.12 gold
```

Therefore, every time you type `https://bronze_a`, no DNS lookups are necessary. The packets from the browser never will reach the Internet and will stay local to 127.0.0.10.

In the main Apache configuration file, `httpd.conf`, you will find an include line that tells Apache to look under the `conf.d` directory for further configuration. For example:

```
Include /etc/apache/conf.d
```

Every time a package is installed that requires configuration changes for Apache, you will find an extra configuration file within the `conf.d` directory. In fact, if you want (for a nice aside), try to install

Drupal and read the Drupal.conf file that is dumped.

I want to keep our work separate from the rest of the world's. No doubt, we will generate mistakes during playtime. Add a second line to include a directory for our virtual hosting files:

```
Include /etc/apache/vhosts
```

Then, make the directories /etc/apache/ssl and /etc/apache/vhosts. Later, we will place our certificates and server keys in the SSL directory, one set per virtual host.

Next, check the httpd.conf file to see whether the SSL engine is turned on. I want to turn the engine off until enabled per virtual host. So, the line `SSLEngine On` should change to `SSLEngine Off`.

Now we have an Apache 1.3.x server that is ready for action.

If you have not set up your Tomcat servers yet, you need to modify the following lines under the `tomcat_root/conf/server.xml` file for the second instance. Change the port numbers to 8015 for the shutdown command and port 8019 for the AJP/1.3 connector:

```
<Server port="8005" shutdown="SHUTDOWN">
<Connector port="8080"
<Connector port="8009"
    enableLookups="false" protocol="AJP/1.3" />
```

For the sake of security, change the shutdown attribute from the value SHUTDOWN to some randomly long string. Otherwise, perhaps on the worst day under a badly defended system, a cracker can Telnet in and type SHUTDOWN, and then your server is down. Also, I would comment out the 8080 connector. There is no need to expose Tomcat directly to the Internet.

Only one task is left—to create two Web applications. Under the `webapps` directory of the first Tomcat instance, create a `bronze_a` directory, and then under that directory, create a `WEB-INF` directory. Place the following `web.xml` file in `WEB-INF`:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<web-app xmlns="http://java.sun.com/xml/ns/j2ee"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://java.sun.com/xml/ns/j2ee
    http://java.sun.com/xml/ns/j2ee/web-app_2_4.xsd"
    version="2.4">

    <display-name>BRONZE_A</display-name>
    <description>
        BRONZE_A Dynamic
    </description>
    <welcome-file-list>
        <welcome-file>
            index.jsp
        </welcome-file>
    </welcome-file-list>
</web-app>
```

Notice the mention of `web-app_2_4.xsd`. This `web.xml` file will not work under Tomcat 5, which uses the 2.3 standard. Under the `webapps/bronze_a` directory, place the following `index.jsp` file. This is our poor yet relevant example of dynamic content:

ASA COMPUTERS

The Expert on Customized Servers!

www.asacomputers.com
1-800-REAL-PCS

Hardware Systems for the open source community - Since 1989.

(Linux, FreeBSD, NetBSD, OpenBSD, Solaris, MS etc.)

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

Your Custom Appliance Solution!!

"Let us know your Needs..."



"We will build you a Solution..."

AMD Opteron(TM) Value Server Starts at \$847

- 1U 14" Deep 260W.
- AMD Opteron 140 CPU.
- 512MB PC 3200 DDR ECC Unbuffered.
- Support upto 8GB DDR RAM.
- 40GB SATA Hard Disk.
- 2x 10/100 Mbps Lan.



Quad AMD Opteron(TM) Server Starts at \$2,812

- 1U AMD Opteron Model 840.
- 2GB Memory. Max 128 GB
- Supports upto 64GB FBDIMM.
- 80 GB SATA II Hotswap Hard Drive.
- 2x Integrated Dual 10/1000 LAN.



Dual AMD Opteron(TM) Storage Starts at \$4,120

- 5U Dual AMD Opteron Model 246.
- iSCSI or NAS Software Options.
- Support upto 18TB of Storage.
- Fail Hard Drive LED Indicator.



Dual AMD Opteron(TM) Storage Starts at \$8,445

- 8U AMD Opteron Model 246.
- 4TB of Storage (36TB Max).
- 1GB RAM
- 2 x 10/100/1000 Gigabit LAN.
- NAS or iSCSI Software Options.



Why Do Business With ASA?

"We Provide Approved EVAL Server..."

Since 1989, ASA has served customers like Cisco, Juniper, Caltech, Fermilab and most Universities. We provide a total custom solution with OS of your choice.

Excellent pre and post-sales support.

"Reliable hardware at the most competitive prices".

Please call or contact us for your next hardware purchase.



2354 Calle Del Mundo, Santa Clara, CA - 95054

www.asacomputers.com

Email : sales@asacomputers.com

Tel: 1-800-REAL-PCS, Fax: 408-654-2910.



```
<%String mess="Hello World from Bronze_a"; %>
<%=mess%> <br><%=request.getRequestURI()%>
```

Follow the same procedure for the second instance, but replace the string `bronze_a` with `silver` under the `webapps/silver` directory of the second Tomcat instance.

Working Together

Making the Apache and Tomcat servers talk with each other is surprisingly straightforward. If this doesn't already exist somewhere within the `httpd.conf` file, add the following lines to the end of the file:

```
JkWorkersFile /usr/local/apache/conf/workers.properties
JkLogFile /usr/local/apache/logs/mod_jk.log
JkLogLevel error
```

The exact location of the `worker.properties` file is left to your discretion. The `JkLogFile` and `JkLogLevel` values are not necessary, as we will override them within the virtual host files. However, for peace of mind, I like to place default values in case of misconfiguration later. The worker property defines how the connections behave. The first line defines the list of workers—in this case, `bronze` and `silver`. The next lines are for the details of configuration for each worker set. `bronze` attaches itself to port 8008 and `silver` to port 8019, with both sets talking the `AJ1.3` protocol. These two worker sets are mentioned later in the virtual host files:

```
worker.list=bronze,silver

worker.bronze.port=8009
worker.bronze.host=localhost
worker.bronze.type=ajp13

worker.silver.port=8019
worker.silver.host=localhost
worker.silver.type=ajp13
```

Virtual Hosting

Virtual hosting is the hosting of multiple servers on one machine by listening for either incoming hostnames or IP addresses. Using multiple virtual hosts with SSL works only for IP-based virtual hosting. Let me explain by example. First, say I want to view a normal transaction between a Web browser and a server. To achieve this, I use the rather excellent Apache SOAP tool `TcpTunnelGui`. To do this, first download the current archive from the Apache SOAP Web site (see Resources). On expanding it, you will see a directory called `lib`. Perform the following actions, and if all goes well, you will have Java installed locally and have brought up the GUI:

```
cd lib
java -cp ./soap.jar org.apache.soap.util.net.TcpTunnelGui
➔9001 localhost 80
```

The GUI displays the text from any TCP connection going through port 9001 and redirects the input back to localhost 80. Feel free to change localhost to point to your own test Web server. In your browser, type `http://localhost:9001`. Expect to see the following

type of transaction:

```
Accept: */*
Referer: http://localhost:9001
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible;
➔MSIE 6.0; Windows NT 5.1; CHWIE_NL70;
➔SV1; .NET CLR 1.0.3705; .NET CLR 1.1.4322)
Host: localhost:9001
Connection: Keep-Alive
```

As you can see, the browser sends information about itself and also the host and referrer variable. Apache uses the host variable in name-based virtual hosting to work out which configuration file to apply. By typing `https://localhost:9001`, you will get a garbled response similar to:

```
?L^A^C?+^A
```

The Host variable is not available until the SSL encryption is complete. Therefore, having a different SSL certificate per virtual host requires that the SSL process occurs before configuration. Yes, it's the proverbial chicken-and-egg problem. Why do we need multiple SSL certificates and, thus, IP-based configuration in the first place? The answer has to do with the `cn` attribute in the SSL certificate. For a certificate to be accepted as a valid server certificate by browsers, the `cn` attribute has to be defined with the value of the hostname of the target server. So, for the IP address 127.0.0.10, we need a certificate with `cn=bronze_a`, and for the IP address 127.0.0.11, a certificate with `cn=silver`.

To generate self-signed certificates with your own local CA, you need to have `OpenSSL` installed:

```
sudo apt-cache search openssl
sudo apt-get install openssl
```

Use these three commands to generate a self-signed certificate:

```
openssl req -new -out silver.csr
openssl rsa -in privkey.pem -out silver.key
openssl x509 -in silver.csr -out silver.cert
➔-req -signkey silver.key -days 365
```

The first command generates the certificate request. Remember, the `cn` attribute must be the same value as the hostname contained within your virtual host—for example, `silver` or `bronze_a`. The other attributes can be of any text value you consider reasonable.

The second command moves the password from the newly generated server's private key to `silver.key`, removing the password protection. This is needed; otherwise, every time you restart Apache, you will be asked to type in the password at the command line. The final line generates a relevant certificate based on the certificate request. Place both the cert and key files in the `/etc/apache/ssl` directory. Perform the same action for `bronze_a`. Remember to defend the `ssl` directory with the least permissions possible.

To activate both port 80 and 443 for `bronze`, add the following

virtual host under vhosts:

```
Listen 127.0.0.10:443
<VirtualHost 127.0.0.10:443>

ServerName bronze_a
Alias /static/ /var/www/customers/bronze_a/content/
RedirectMatch ^/$ https://bronze_a/bronze_a/

SSLEngine On
SSLCertificateFile ssl/bronze_a.cert
SSLCertificateKeyFile ssl/bronze_a.key

JkMount /bronze_a/* bronze
JkMount /bronze_a bronze
JkLogFile /usr/local/apache/logs/bronze_a_mod_jk.log
JkLogLevel info
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "

RewriteEngine on
RewriteRule ^/(.*)$ https://%{SERVER_NAME}/$1 [R,L]
RewriteRule ^/(.*)$ http://%{SERVER_NAME}/$1 [R,L]
</VirtualHost>
```

```
Listen 127.0.0.10:80
<VirtualHost 127.0.0.10:80>

ServerName bronze_a
Alias /static/ /var/www/customers/bronze_a/content/
RedirectMatch ^/$ https://bronze_a/bronze_a/

JkMount /bronze_a/* bronze
JkMount /bronze_a bronze
JkLogFile /usr/local/apache/logs/80_bronze_a_mod_jk.log
JkLogLevel info
JkLogStampFormat "[%a %b %d %H:%M:%S %Y] "

RewriteEngine on
RewriteRule ^/(.*)$ https://%{SERVER_NAME}/$1 [R,L]
RewriteRule ^/(.*)$ http://%{SERVER_NAME}/$1 [R,L]

</VirtualHost>
```

For silver, create a similar virtual host but with the 127.0.0.11 IP address, and replace the string bronze_a with the string silver. Double-check that your SSL certificate and private key are pointed to correctly.

It's back! Bigger and Better!

February 8-10th, 2008
Westin LAX, Los Angeles, California

Commercial and non-profit exhibitors!

Over 60 speakers, talking about the latest developments in Open Source!

Expanded specialty sessions on Friday:

WIOS - Women In Open Source

DOHCS - Demonstrating Open Source Health Care Solutions

OSSIE - Open Source Software in Education

The Sixth Annual
Southern California Linux Expo

SCALE 6x



Your community-based Open Source Software show in the Southwest!

<http://www.socallinuxexpo.org>

Use Promo code LJAD for a 30% discount on admission to SCALE

Turn on SSL with these three little commands:

```
SSLEngine On
SSLCertificateFile ssl/bronze_a.cert
SSLCertificateKeyFile ssl/bronze_a.key
```

The rewrite rules are culled from the mod_ssl FAQ. What is happening is that you have control over the relative URLs, so you can switch between the SSL and non-SSL port easily. When you use /url:NOSSL as a URL, the URL is rewritten to HTTP instead of HTTPS, and the same is true for HTTP to HTTPS using /url:SSL.

Enable the mounting of the Tomcat server by the workers with the following:

```
JkMount /bronze_a/* bronze
JkMount /bronze_a bronze
```

It is good practice to separate log files used potentially for debugging—for example:

```
JkLogFile /usr/local/apache/logs/80_bronze_a_mod_jk.log
```

Living with Static and Dynamic Content

Apache is better than Tomcat for delivering static content, security and URL reshaping. Therefore, it is of global benefit to separate the static and dynamic content of your site and allow Apache to deal with the static content via the filesystem and the dynamic content via mod_jk. One instance of the URL remapping is the top-level redirect. We had mounted mod_jk at /bronze_a. If a user had typed `http://bronze_a/`, he or she would either have found an empty page or seen a pretty file listing. You can resolve this issue by placing an `index.html` page at the top-level location or by redirecting down. The redirection is achieved via:

```
RedirectMatch ^/$† https://bronze_a/bronze_a/
```

To make sure the right page is picked up by the uri `/bronze_a/`, the following lines exist in the `web.xml` file:

```
<welcome-file-list>
  <welcome-file>
    index.jsp
  </welcome-file>
</welcome-file-list>
```

A simple method to link to the static content is to use an alias within the virtual host. For example, `https://bronze_a/static/`:

```
Alias /static/ /var/www/customers/bronze_a/content/
```

Developing Java Web applications tends to be a team sport. Static content, such as images (at least in my environment), tend to change more than the application itself. Therefore, you may consider doing the obvious and setting an FTP root above the static content, but not above the more sensitive dynamic content. Then, you can force the Web application to go through a full series of tests before placing any new version in production. In

fact, you may even consider a hybrid solution. Developers like to work through CVS. By placing both static and dynamic content within a war file, you keep all your code and content together and have a synchronized deployment via the re-installation of the war file. This simplifies deployment, and system administrators have to perform the same repetitive task only when new property files or content is approved. Next, you would need to add some `AliasMatch` rules to treat certain URLs as file locations, dishing the files up directly rather than through mod_jk, thus avoiding potential performance hits. For example:

```
AliasMatch /web/customers/(.*)/javascript/(.*)
  /usr/local/tomcat6/webapps/$1/javascript/$2
AliasMatch /web/customers/(.*)/images/(.*)
  /usr/local/tomcat6/webapps/$1/images/$2
AliasMatch /web/customers/(.*)/css/(.*)
  /usr/local/tomcat6/webapps/$1/css/$2
```

This would map files in the CSS, JavaScript or image directories in the Web application as static content. For example, `https://xxxxx/web/customers/little.com/javascript/editor.js` translates to `/usr/local/tomcat6/webapps/little.com/javascript/editor.js`.

Conclusion

There are many ways to kill a cat, which, for cats, is most unfortunate. This article has shown one approach to hosting Web applications. I do not pretend that this is the only approach; it's simply one that has worked for me. At great speed I have mentioned mod_ssl, mod_jk and one approach to separating static and dynamic content. I hope this article has given you enough information to have a go at testing your hosting concepts yourself. With some basic configuration, it is relatively straightforward to control your SSL-enabled virtual hosts. ■

Alan Berg, Bsc, MSc, PGCE, has been a lead developer at the Central Computer Services at the University of Amsterdam for the last seven years. In his spare time, he writes computer articles. He has a degree, two Masters' degrees and a teaching qualification. In previous incarnations, he was a technical writer, an Internet/Linux course writer and a science teacher. He likes to get his hands dirty with the building and gluing of systems. He remains agile by playing computer games with his kids who (sadly) consistently beat him. You can contact him at reply.to.berg@chello.nl.

Resources

Apache SOAP: ws.apache.org/soap

"Connecting Apache's Web Server to Multiple Instances of Tomcat" by Daniel McCarthy: www.linuxjournal.com/article/8561

mod_ssl: www.modssl.org

OpenSSL: www.openssl.org

Tomcat Home Page: tomcat.apache.org

Virtual Hosting: httpd.apache.org/docs/1.3/vhosts

LINUX JOURNAL



1994-2006 ARCHIVE

ISSUES 1-152 of *Linux Journal*

www.LinuxJournal.com/ArchiveCD

The 1994–2006 Archive CD,
back issues, and more!

Creating VPNs with IPsec and SSL/TLS

How to create IPsec and SSL/TLS tunnels in Linux. RAMI ROSEN

VPN (Virtual Private Network) is a technology that provides secure communication through an insecure and untrusted network (like the Internet). Usually, it achieves this by authentication, encryption, compression and tunneling. Tunneling is a technique that encapsulates the packet header and data of one protocol inside the payload field of another protocol. This way, an encapsulated packet can traverse through networks it otherwise would not be capable of traversing.

Currently, the two most common techniques for creating VPNs are IPsec and SSL/TLS. In this article, I describe the features and characteristics of these two techniques and present two short examples of how to create IPsec and SSL/TLS tunnels in Linux and verify that the tunnels started correctly. I also provide a short comparison of these two techniques.

IPsec and Openswan

IPsec (IP security) provides encryption, authentication and compression at the network level. IPsec is actually a suite of protocols, developed by the IETF (Internet Engineering Task Force), which have existed for a long time. The first IPsec protocols were defined in 1995 (RFCs 1825–1829). Later, in 1998, these RFCs were deprecated by RFCs 2401–2412. IPsec implementation in the 2.6 Linux kernel was written by Dave Miller and Alexey Kuznetsov. It handles both IPv4 and IPv6. IPsec operates at layer 3, the network layer, in the OSI seven-layer networking model. IPsec is mandatory in IPv6 and optional in IPv4. To implement IPsec, two new protocols were added: Authentication Header (AH) and Encapsulating Security Payload (ESP). Handshaking

and exchanging session keys are done with the Internet Key Exchange (IKE) protocol.

The AH protocol (RFC 2404) has protocol number 51, and it authenticates both the header and payload. The AH protocol does not use encryption, so it is almost never used.

ESP has protocol number 50. It enables us to add a security policy to the packet and encrypt it, though encryption is not mandatory. Encryption is done by the kernel, using the kernel CryptoAPI. When two machines are connected using the ESP protocol, a unique number identifies this connection; this number is called SPI (Security Parameter Index). Each packet that flows between these machines has a Sequence Number (SN), starting with 0. This SN is increased by one for each sent packet. Each packet also has a checksum, which is called the ICV (integrity check value) of the packet. This checksum is calculated using a secret key, which is known only to these two machines.

IPsec has two modes: transport mode and tunnel mode. When creating a VPN, we use tunnel mode. This means each IP packet is fully encapsulated in a newly created IPsec packet. The payload of this newly created IPsec packet is the original IP packet.

Figure 2 shows that a new IP header was added at the right, as a result of working with a tunnel, and that an ESP header also was added.

There is a problem when the endpoints (which are sometimes called peers) of the tunnel are behind a NAT (Network Address Translation) device. Using NAT is a method of connecting multiple machines that have an “internal address”, which are not accessible directly to the outside world. These machines access the outside world

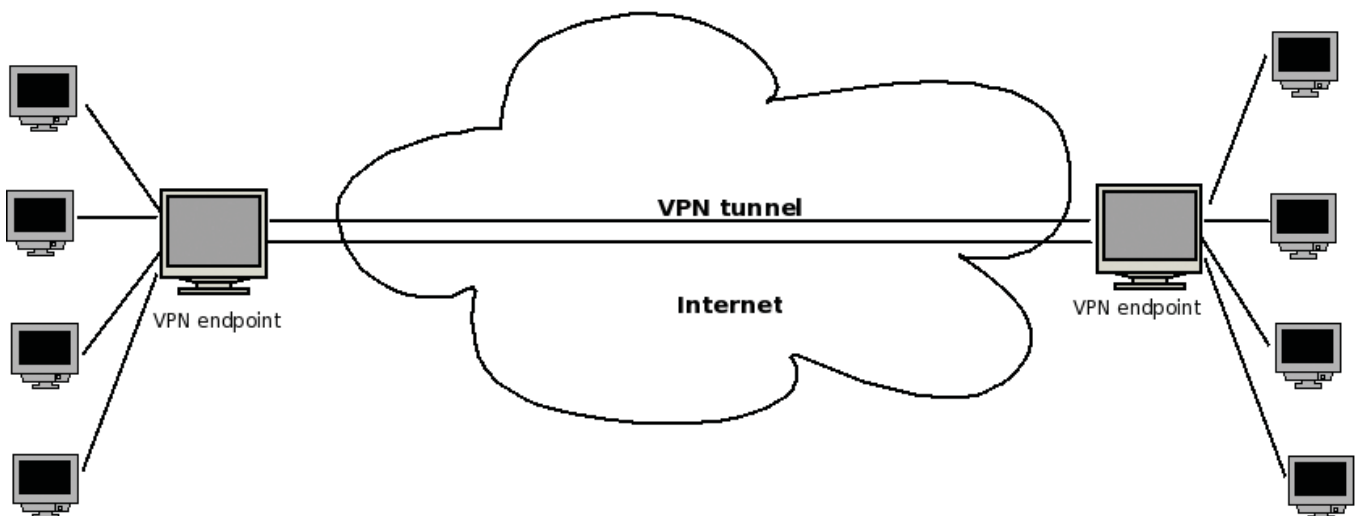


Figure 1. A Basic VPN Tunnel

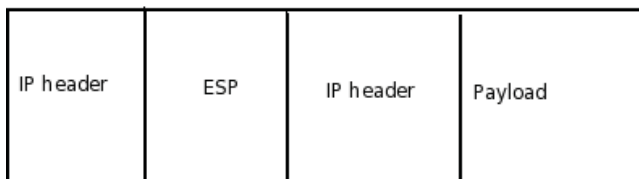


Figure 2. An IPsec Tunnel ESP Packet

the contents of the IP packet. As a result, this packet will be rejected by the peer because the signature is wrong. Thus, the IETF issued some RFCs that try to find a solution for this problem. This solution commonly is known as NAT-T or NAT Traversal. NAT-T works by encapsulating IPsec packets in UDP packets, so that these packets will be able to pass through NAT routers without being dropped. RFC 3948, UDP Encapsulation of IPsec ESP Packets, deals with NAT-T (see Resources).

Openswan is an open-source project that provides an implementation of user tools for Linux IPsec. You can create a VPN using Openswan tools (shown in the short example below). The Openswan Project was started in 2003 by former FreeS/WAN developers. FreeS/WAN is the predecessor of Openswan. S/WAN stands for Secure Wide Area Network, which is actually a trademark of RSA. Openswan runs on many different platforms, including x86, x86_64, ia64, MIPS and ARM. It supports kernels 2.0, 2.2, 2.4 and 2.6.

Two IPsec kernel stacks are currently available: KLIPS and NETKEY. The Linux kernel NETKEY code is a rewrite from scratch of the KAME IPsec code. The KAME Project was a group effort of six companies in Japan to provide a free IPv6 and IPsec (for both IPv4 and IPv6) protocol stack implementation for variants of the BSD UNIX computer operating system.

KLIPS is not a part of the Linux kernel. When using KLIPS, you must apply a patch to the kernel to support NAT-T. When using NETKEY, NAT-T support is already inside the kernel, and there is no need to patch the kernel.

When you apply firewall (iptables) rules, KLIPS is the easier case, because with KLIPS, you can identify IPsec traffic, as this traffic goes through ipsecX interfaces. You apply iptables rules to these interfaces in the same way you apply rules to other network interfaces (such as eth0).

When using NETKEY, applying firewall (iptables) rules is much more complex, as the traffic does not flow through ipsecX interfaces; one solution can be marking the packets in the Linux kernel with iptables (with a setmark iptables rule). This mark is a member of the kernel socket buffer structure (struct sk_buff, from the Linux kernel networking code); decryption of the packet does not modify that mark.

Openswan supports Opportunistic Encryption (OE), which enables the creation of IPsec-based VPNs by advertising and fetching public keys from a DNS server.

OpenVPN

OpenVPN is an open-source project founded by James Yonan. It provides a VPN solution based on SSL/TLS. Transport Layer Security (TLS) and its predecessor, Secure Sockets Layer (SSL), are cryptographic protocols that provide secure communications data transfer on the Internet. SSL has been in existence since the early '90s.

The OpenVPN networking model is based on TUN/TAP virtual devices; TUN/TAP is part of the Linux kernel. The first TUN driver in Linux was developed by Maxim Krasnyansky.

OpenVPN installation and configuration is simpler in comparison with IPsec. OpenVPN supports RSA authentication, Diffie-Hellman key agreement, HMAC-SHA1 integrity checks and more. When running in server mode, it supports multiple clients (up to 128) to connect to a VPN server over the same port. You can set up your own Certificate Authority (CA) and generate certificates and keys for an OpenVPN server and multiple clients.

OpenVPN operates in user-space mode; this makes it easy to port OpenVPN to other operating systems.

through a machine that does have an Internet address; the NAT is performed on this machine—usually a gateway.

When the endpoints of the tunnel are behind a NAT, the NAT modifies

Linux Laptops

Starting at \$799



Linux Desktops

Starting at \$375



Linux Servers

Starting at \$899



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



309.34.CUBED
shopcubed.com

NAT-T works by encapsulating IPsec packets in UDP packets, so that these packets will be able to pass through NAT routers without being dropped.

Example: Setting Up a VPN Tunnel with IPsec and Openswan

First, download and install the ipsec-tools package and the Openswan package (most distros have these packages).

The VPN tunnel has two participants on its ends, called left and right, and which participant is considered left or right is arbitrary. You have to configure various parameters for these two ends in `/etc/ipsec.conf` (see `man 5 ipsec.conf`). The `/etc/ipsec.conf` file is divided into sections. The `conn` section contains a connection specification, defining a network connection to be made using IPsec.

An example of a `conn` section in `/etc/ipsec.conf`, which defines a tunnel between two nodes on the same LAN, with the left one as 192.168.0.89 and the right one as 192.168.0.92, is as follows:

```
...
conn linux-to-linux
#
```

```
# Simply use raw RSA keys
# After starting openswan, run:
# ipsec showhostkey --left (or --right)
# and fill in the connection similarly
# to the example below.
left=192.168.0.89
leftrsasigkey=0sAQPP...
# The remote user.
#
right=192.168.0.92
rightrsasigkey=0sAQON...
type=tunnel
auto=start
```

...

You can generate the `leftrsasigkey` and `rightrsasigkey` on both participants by running:

```
ipsec rsasigkey --verbose 2048 > rsa.key
```

Then, copy and paste the contents of `rsa.key` into `/etc/ipsec.secrets`.

In some cases, IPsec clients are roaming clients (with a random IP address). This happens typically when the client is a laptop used from remote locations (such clients are called Roadwarriors). In this case, use the following in `ipsec.conf`:

```
right=%any
```

TECH TIP

Removing Duplicate Lines in Unsorted Text without Losing Input Order

Here's the problem. We need to remove duplicate lines from unsorted text from within the shell. This normally would be the job of `sort -u` or `sort | uniq`, except that in either case, we lose the input order. For example, if this is the input file:

```
$ cat >/tmp/numbers <<EOF
one
two
three
one
three
four
EOF
```

Running it through `sort -u` would get this:

```
$ sort -u /tmp/numbers
four
one
three
two
```

The solution:

```
$ n1 /tmp/numbers | sort -k2 -u | sort -n | cut -f2-
one
two
three
four
```

For platforms where the `n1` command is not available, `awk` could be used to simulate this behavior:

```
$ awk -v 'OFS=\t' '{print NR, $0}' /tmp/numbers | sort -k2 -u
1| one
2| two
3| three
4| four
```

What it does is pretty simple—it adds a record number field, sorts the input ignoring that field, restores the original order using the record numbers and then strips that field out.

—VENKY TV

instead of:

```
right=ipAddress
```

The %any keyword is used to specify an unknown IP address.

The type parameter of the connection in this example is tunnel (which is the default). Other types can be transport, signifying host-to-host transport mode; passthrough, signifying that no IPsec processing should be done at all; drop, signifying that packets should be discarded; and reject, signifying that packets should be discarded and a diagnostic ICMP should be returned.

The auto parameter of the connection tells which operation should be done automatically at IPsec startup. For example, auto=start tells it to load and initiate the connection; whereas auto=ignore (which is the default) signifies no automatic startup operation. Other values for the auto parameter can be add, manual or route.

After configuring /etc/ipsec.conf, start the service with:

```
service ipsec start
```

You can perform a series of checks to get info about IPsec on your machine by typing ipsec verify. And, output of ipsec verify might look like this:

```
Checking your system to see if IPsec has installed and started correctly:
Version check and ipsec on-path [OK]
Linux Openswan U2.4.7/K2.6.21-rc7 (netkey)
Checking for IPsec support in kernel [OK]
NETKEY detected, testing for disabled ICMP send_redirects [OK]
NETKEY detected, testing for disabled ICMP accept_redirects [OK]
Checking for RSA private key (/etc/ipsec.d/hostkey.secrets) [OK]
Checking that pluto is running [OK]
Checking for 'ip' command [OK]
Checking for 'iptables' command [OK]
Opportunistic Encryption Support [DISABLED]
```

You can get information about the tunnel you created by running:

```
ipsec auto --status
```

You also can view various low-level IPsec messages in the kernel syslog.

You can test and verify that the packets flowing between the two participants are indeed esp frames by opening an FTP connection (for example), between the two participants and running:

```
tcpdump -f esp
tcpdump: verbose output suppressed, use -v or -vv for full protocol decode
listening on eth0, link-type EN10MB (Ethernet), capture size 96 bytes
```

You should see something like this:

```
IP 192.168.0.92 > 192.168.0.89: ESP(spi=0xd514eed9,seq=0x7)
IP 192.168.0.89 > 192.168.0.92: ESP(spi=0x3a1563b9,seq=0x6)
IP 192.168.0.89 > 192.168.0.92: ESP(spi=0x3a1563b9,seq=0x7)
IP 192.168.0.92 > 192.168.0.89: ESP(spi=0xd514eed9,seq=0x8)
```

When you apply firewall (iptables) rules, KLIPS is the easier case, because with KLIPS, you can identify IPsec traffic, as this traffic goes through ipsecX interfaces.

Note that the spi (Security Parameter Index) header is the same for all packets; this is an identifier of the connection.

If you need to support NAT traversal, add nat_traversal=yes in ipsec.conf; nat_traversal=no is the default.

The Linux IPsec stack can work with pluto from Openswan, racoon from the KAME Project (which is included in ipsec-tools) or isakmpd from OpenBSD.

Example: Setting Up a VPN Tunnel with OpenVPN

First, download and install the OpenVPN package (most distros have this package).

Then, create a shared key by doing the following:

```
openvpn --genkey --secret static.key
```

You can create this key on the server side or the client side, but you should copy this key to the other side in a secured channel (like SSH, for example). This key is exchanged between client and server when the tunnel is created.

This type of shared key is the simplest key; you also can use CA-based keys. The CA can be on a different machine from the OpenVPN server. The OpenVPN HOWTO provides more details on this (see Resources).

Then, create a server configuration file named server.conf:

```
dev tun
ifconfig 10.0.0.1 10.0.0.2
secret static.key
comp-lzo
```

On the client side, create the following configuration file named client.conf:

```
remote serverIpAddressOrHostName
dev tun
ifconfig 10.0.0.2 10.0.0.1
secret static.key
comp-lzo
```

Note that the order of IP addresses has changed in the client.conf configuration file.

The comp-lzo directive enables compression on the VPN link.

You can set the mtu of the tunnel by adding the tun-mtu directive. When using Ethernet bridging, you should use dev tap instead of dev tun.

The default port for the tunnel is UDP port 1194 (you can verify

this by typing `netstat -nl | grep 1194` after starting the tunnel).

Before you start the VPN, make sure that the TUN interface (or TAP interface, in case you use Ethernet bridging) is not firewalled.

Start the vpn on the server by running `openvpn server.conf` and running `openvpn client.conf` on the client.

You will get an output like this on the client:

```
OpenVPN 2.1_rc2 x86_64-redhat-linux-gnu [SSL] [LZO] [EPOLL]
built on
Mar  3 2007
IMPORTANT: OpenVPN's default port number is now 1194, based on an
official
port number assignment by IANA. OpenVPN 2.0-beta16 and earlier
used 5000
as
the default port.
LZO compression initialized
TUN/TAP device tun0 opened
/sbin/ip link set dev tun0 up mtu 1500
/sbin/ip addr add dev tun0 local 10.0.0.2 peer 10.0.0.1
UDPv4 link local (bound): [undef]:1194
UDPv4 link remote: 192.168.0.89:1194
Peer Connection Initiated with 192.168.0.89:1194
Initialization Sequence Completed
```

You can verify that the tunnel is up by pinging the server from the client (ping 10.0.0.1 from the client).

The TUN interface emulates a PPP (Point-to-Point) network device and the TAP emulates an Ethernet device. A user-space program can open a TUN device and can read or write to it. You can apply iptables rules to a TUN/TAP virtual device in the same way you would do it to an Ethernet device (such as eth0).

IPsec and OpenVPN—a Short Comparison

IPsec is considered the standard for VPN; many vendors (including Cisco, Nortel, Check Point and many more) manufacture devices with built-in IPsec functionalities, which enable them to connect to other IPsec clients.

However, we should be a bit cautious here: different manufacturers may implement IPsec in a noncompatible manner on their devices, which can pose a problem.

OpenVPN is not supported currently by most vendors.

IPsec is much more complex than OpenVPN and involves kernel code; this makes porting IPsec to other operating systems a much heavier task. It is much easier to port OpenVPN to other operating systems than IPsec, because OpenVPN runs entirely in user space and is not involved with kernel code.

Both IPsec and OpenVPN use HMAC (Hash Message Authentication Code) to authenticate packets.

OpenVPN is based on using the OpenSSL library; it can run over UDP (which is the default and preferred protocol) or TCP. As opposed to IPsec, which runs in kernel, it runs in user space, so it is heavier than IPsec in terms of performance.

Configuring and applying firewall (iptables) rules in OpenVPN

TECH TIP

Cool SSH Automation Trick

Do you need to give passwordless SSH access to users but also need to restrict what can be run? Here I give interactive and non-interactive examples.

First, you need to generate authentication keys (no passphrase) using `ssh-keygen`. In this tip, I generated `rsa` keys. The keys, by default, are saved as `~/.ssh/id_rsa` and `~/.ssh/id_rsa.pub`.

You need to append the public key as follows:

```
suman@shri:~/.ssh$ cat id_rsa.pub >> authorized_keys
```

Prepend the command (that you want to be executed automatically) to the signature. For example, if `uptime` is the command to be executed automatically whenever an SSH request comes from the user for which you generated the key, the key entry in `authorized_keys` file should look like this:

```
command="/usr/bin/uptime" ssh-rsa <LONG ENCRYPTED STRING>
```

Now, you can use the private key to execute the `uptime` command automatically. The private key in this example, by default, was saved as `~/.ssh/id_rsa`. I copied it to the remote host and saved it as `uptime.key`. Make sure this file has 600 permissions. Then, you

can do the following (from the remote machine):

```
suman@strangeloop:~ % ssh -T -i uptime.key suman@shri
15:11:46 up 4 days, 3 min,  3 users,  load average: 0.00, 0.00, 0.00
```

This technique also can be used for interactive programs. Below is a simple interactive shell script:

```
#!/bin/sh
echo -n "Hi! Enter your fav distro: "
read DISTRO
echo "Your fav distro is $DISTRO"
```

Here I have created another set of keys as I did above. I saved the private key as `distro.key`. Prepend `command="<full_path_to_script>"` to the public key in the `authorized_keys` file, and you will be able to do:

```
suman@strangeloop:~ % ssh -T -i distro.key suman@shri
Hi! Enter your fav distro: Debian
Your fav distro is Debian
```

—SUMAN CHAKRAVARTULA

is usually easier than configuring such rules with Openswan in an IPsec-based tunnel.

Acknowledgement

Thanks to Mr Ken Bantoft for his comments.■

Rami Rosen is a computer science graduate of Technion, the Israel Institute of Technology, located in Haifa. He works as a Linux and Open Solaris kernel programmer for a networking startup, and he can be reached at ramirose@gmail.com. In his spare time, he likes running, solving cryptic puzzles and helping everyone he knows move to this wonderful operating system, Linux.

Resources

OpenVPN: openvpn.net

OpenVPN 2.0 HOWTO: openvpn.net/howto.html

RFC 3948, UDP Encapsulation of IPsec ESP Packets:
tools.ietf.org/html/rfc3948

Openswan: www.openswan.org

The KAME Project: www.kame.net

TECH TIP

Tweaking Forced Hard Disk Checks

There is nothing worse than when you boot up your Linux machine to show Windows users a neat feature, such as your slick 3-D Compiz desktop, and while booting, you get the message that one of your partitions needs to be force-checked because it hasn't been checked in 23 boots. So while e2fsck trudges through that giant partition scanning for errors, the people you were attempting to convert are rolling their eyes and trying to contain their laughter about your so-called superior OS.

You can tweak the conditions under which the filesystem check will run at boot time using the command `tune2fs`. Using a `-c` option allows you to tweak the number of boots that will trigger a forced check, and using `-i` allows you to change the time interval between checks in the format: `[days]weeks[months]`.

If you want to initiate disk checks manually only (a risky idea if you forget—proceed at your own risk!), use this variant of the command:

```
tune2fs -c 0 -i 0
```

—MAX LUEBBE

Do you take

"the computer doesn't do that"

as a personal challenge?

So do we.

LINUX
JOURNAL™

Since 1994: The Original Monthly Magazine of the Linux Community

Subscribe today at www.linuxjournal.com



Why to Build on FOSS in the First Place

Linux, free software and open source may be “generic”. But, that’s why you need it. DOC SEARLS

The GNU Project has been around since 1983, Linux since 1991, *Linux Journal* since 1994 and the Open Source Initiative since 1998. That means some of us been explaining this stuff for going on a quarter century—or more, in some cases.

Yet, we’re not being clear. What’s obvious to us is still not apparent to others, even after years of explaining what it’s all about. Hardly a week goes by when I don’t find myself explaining, for example:

- Free software and open source are not just ways to cheap out.
- Asking “How do you make money with it?” is the wrong question.
- Much more money is made by using Linux than by selling it.

It helps to use the Net as an example, because it’s more than something you’ve gotta have; it’s foundational.

But Linux, free software and open source aren’t there yet. They’ve “won” in many cases, but their advantages are plain only to technologists, and far from all of those. Those of us who understand it are still being opaque to high-level decision-makers.

Last October the on-line magazine *Baseline* ran a story titled “CIOs Told to Make Conspicuous Contribution to Revenue”. It was a report from the Gartner Symposium/ITxpo in Orlando. The goal, they were told, was to “try to define IT as an integral part of the business”, and “aligned with initiatives that really make a difference for the business”. Said Mark P. McDonald, group Vice President for Gartner’s executive programs, “In 2008, your goal should be to stamp out generic IT.”

The problem word, of course, is generic. Linux is generic. Free software and open-source building materials are all generic. They aren’t no-names, but they are, by intention, commodities. Yet common wisdom says that if you want something to be known, to be

unique, to be valuable, it can’t be generic.

Linux and the whole FOSS portfolio, now numbering several hundred thousand code bases, are both generic *and* valuable. They just don’t have a value that’s made to sell. As Eric S. Raymond pointed out long ago, open-source code has enormous “use value”. Thanks to that use value, Google can exist. Amazon can exist. The Net itself can exist.

What’s missing is the connection between pure use value and all kinds of sale value. That’s what we’ve been calling the because effect. You make money because of free and open code, not just with it.

I suggest the relationship here is between foundations and the structures that rest on them. You can talk about architecture and design all day, but none of it will be worth anything if it doesn’t sit on a strong foundation. This fact does not diminish the importance of foundations. Quite the opposite. Foundations are, in nearly all cases, 100% useful and 0% flashy. Their job is not to augment the building, but rather to augment the geology below it.

Now, let’s go back to *Baseline*’s coverage of the Gartner event where CIOs were being warned about “generic IT”. A companion piece, titled “The 10 Most Important Technology Areas for 2008, Per Gartner”, outlines a pile of nongeneric things IT can do to win the hearts and minds on companies’ top floors and corner offices. Among them are “social software”, “Web platform and Web-oriented architecture”, “metadata management” and “green IT”. All of those are bound to deploy easier, and work better, if they’re built on free and open foundations.

Back in the November 1999 issue of *Linux Journal*, I wrote a column titled “Hacking an Industry”. Here’s an excerpt, from a section where I was talking about e-commerce:

Now think about the infrastructure involved here....“Huge” doesn’t cover it. What’s it going to take to build out the infrastructure behind all that? We

know it’ll take two things for sure: Linux and Apache—two well-proven kinds of building material. Of course, Windows 2000 will be involved too. There are just too many people already constructing this new skyline with Microsoft tools and building materials. The difference is that the builders themselves help improve Linux, Apache and other open-source products. They can’t do the same for Microsoft.

One developer put it to me this way: “When I’m building a skyscraper, I want to know there’s rebar in the concrete. With Linux, I know. With Microsoft, I don’t. In fact, NT’s memory leaks prove to me there isn’t rebar in there. Since I have to work with NT for political reasons, I just cope with it. But I know if we could see the source, we could probably fix the problem pretty fast.”

More than eight years later, IT’s foundations include a lot more than Linux and Apache. But they still don’t show off. They just support everything. Yet this can’t be obvious if people still want to burden foundations with making money or “contributing to revenue”.

So the real challenge here is understanding infrastructure—of knowing, clearly, what’s foundational and what isn’t. Even with the Net, which has the clearest pure use value on Earth, most of us still don’t grok that it’s a pure utility like water, roads and waste treatment. Yes, it has costs, but its use value verges on the absolute: you have to have it.

So that’s the point. Running your IT on FOSS is as necessary as erecting your buildings on a solid foundation. You can find that out now or later, when you have to build your IT architecture all over again. ■

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



Russ Barnard, President, FlapDaddy Productions

“Fanatical Support™ saved me from my own mistake.”

“Not long ago, I reformatted one of our servers. Not until I was driving home did I learn that I brought our entire site down in the process. I called my guy at Rackspace and he said, ‘We’re already on it.’ By the time I pulled in the driveway, my site was back up. Now that’s Fanatical Support.”

Keeping little mistakes from causing big problems is one definition of Fanatical Support. What will yours be?

Watch Russ’s story at www.rackspace.com/fanatical
1-888-571-8976

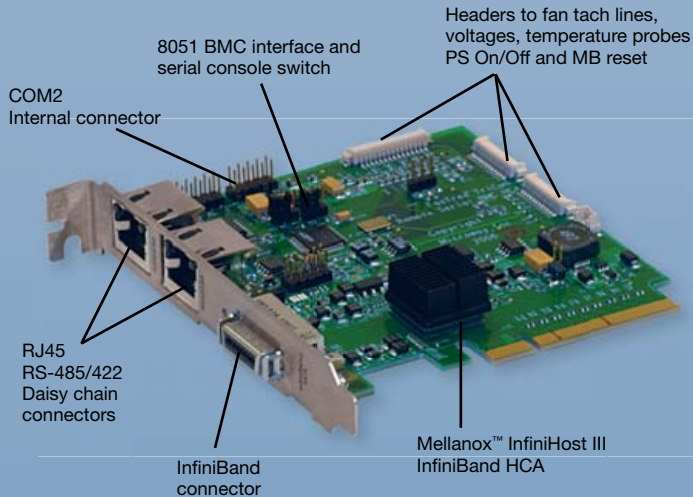


Affordable InfiniBand Solutions

4 Great Reasons to Call Microway NOW!

1 **TriCom™**

- DDR/SDR InfiniBand HCA
- "Switchless" serial console
- NodeWatch web enabled remote monitor and control



2 **FasTree™**

- DDR InfiniBand switches
- Low latency, modular design
- 24, 36 and 48 port building blocks



3 **InfiniScope™**

- Monitors ports on HCA's and switches
- Provides real time BW diagnostics
- Finds switch and cable faults
- Lane 15 interface
- Logs all IB errors



4 **ServaStor™**

- Extensible IB based storage building blocks
- Redundant and scalable
- Parallel file systems
- Open source software
- On-line capacity expansion
- RAID 0,1,1E, 3, 5, 6, 10, 50



Upgrade your current cluster, or let us design your next one using Microway InfiniBand Solutions.

To speak to an HPC expert
call **508 746-7341** and ask
for technical sales or email
sales@microway.com
www.microway.com

 **Microway**
Technology you can count onsm