Cucumber | Fldigi | Vimperator | Firewall Builder | knockd | Xen

LINUX JOURNAL™

Since 1994: The Original Magazine of the Linux Community

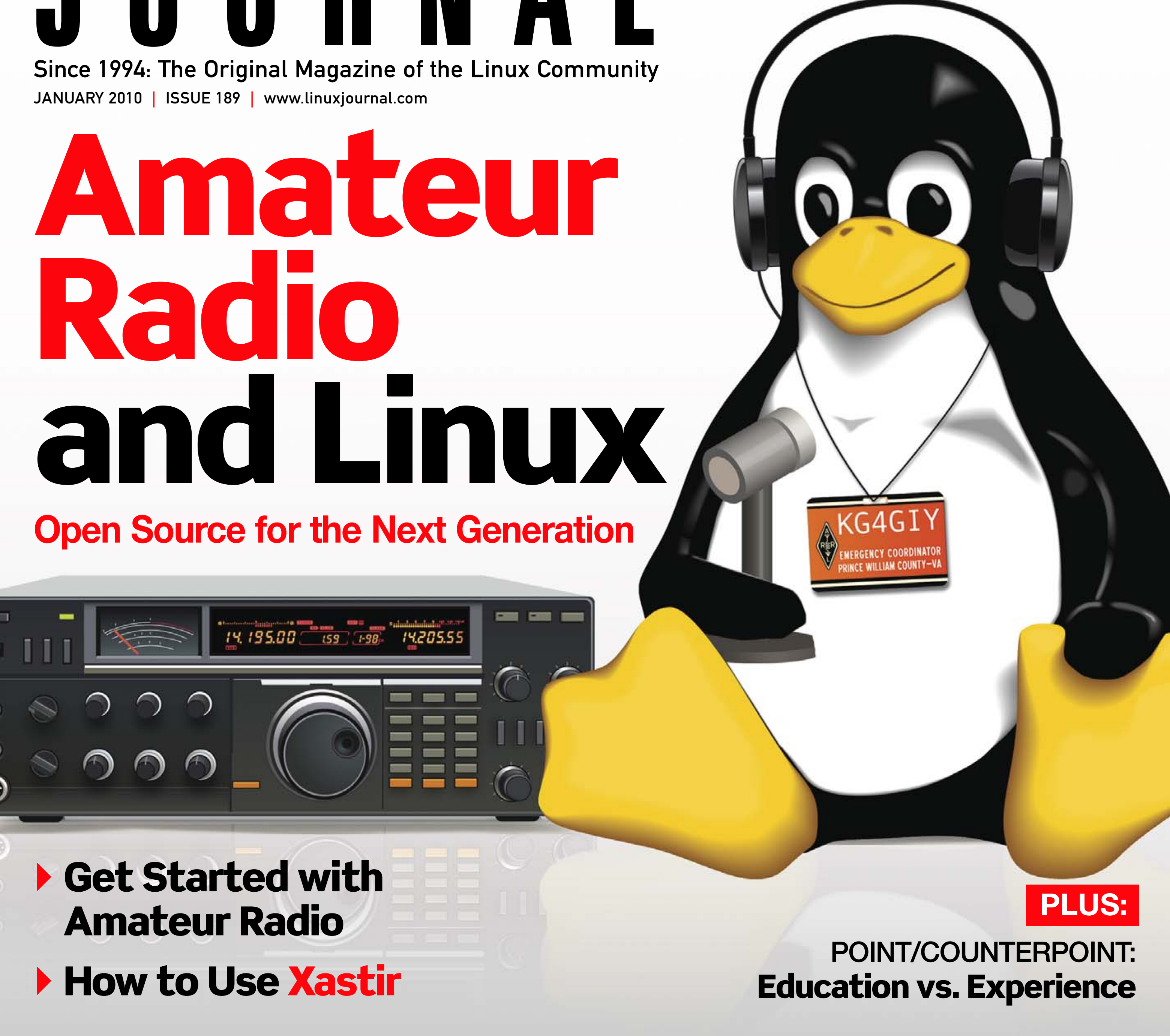
JANUARY 2010 | ISSUE 189 | www.linuxjournal.com

Linux Security Challenges for 2010

Make Your Own Software Virtual Appliance

Amateur Radio and Linux

Open Source for the Next Generation



- Get Started with Amateur Radio
- How to Use Xastir

PLUS:

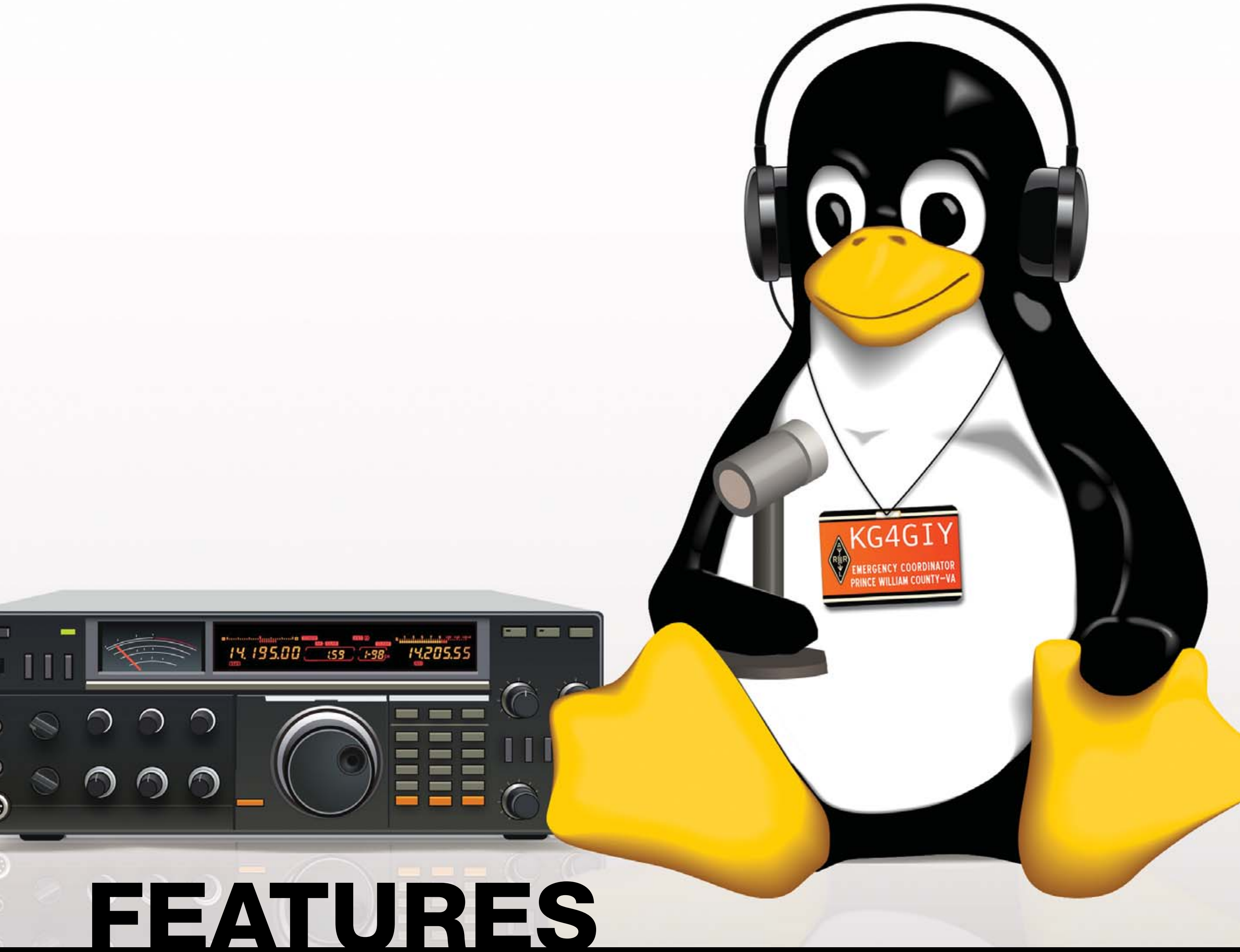
POINT/COUNTERPOINT: Education vs. Experience

Port-Knocking Security with knockd

New Features in Firewall Builder 3.0

$5.99US $5.99CAN

CONTENTS JANUARY 2010 Issue 189



FEATURES

AMATEUR RADIO

MPLS for the masses

$39 95

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

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

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

Features:

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

Benefits:

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

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

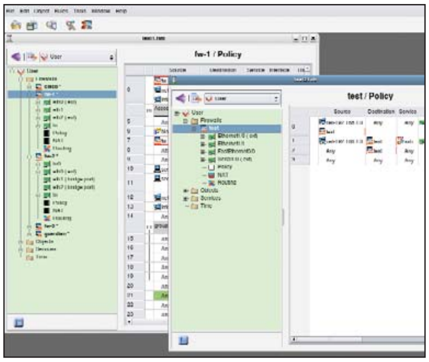
MikroTik routerboard

www.mikrotik.com/mpls

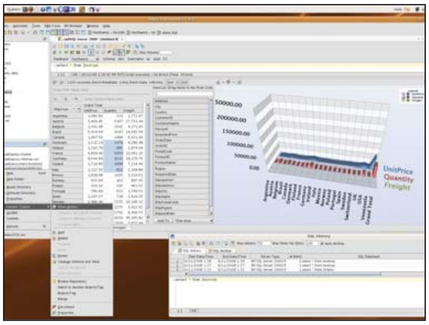
CONTENTS JANUARY 2010 Issue 189

COLUMNS

INDEPTH



IN EVERY ISSUE



Next Month

DESKTOP

It may be getting cold up north, but the Linux desktop just keeps on sizzling. It may not have caught fire yet, but it keeps on getting better.

Next month, read Bruce Byfield's fourth semi-bi-annual (that is, when it seems like a good time) comparison of OpenOffice.org and MS Office. The results may surprise you!

As most new KDE distros move to make KDE4 the standard rather than an option, the interest in KDE4 development is bound to accelerate. Get ahead of the game and find out how to write Plasmoids for KDE4.

If you're still having trouble getting rid of that other OS, find out how to make it a bit more palatable by running your KDE4 applications on Windows, and before long, you may be running the KDE4 desktop on Windows. Now that's cooking with gas!

USPS *LINUX JOURNAL* (ISSN 1075-3583) (USPS 12854) 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 $29.50/year in the United States, $39.50 in Canada and Mexico, $69.50 elsewhere. POSTMASTER: Please send address changes to *Linux Journal*, PO Box 16476, North Hollywood, CA 91615. Subscriptions start with the next issue. Canada Post: Publications Mail Agreement #41549519. Canada Returns to be sent to Bleuchip International, P.O. Box 25542, London, ON N6C 6B2

ABERDEEN
SERVERS AND STORAGE

WHAT'S THE DEAL WITH THESE GUYS?

Sometimes you have to ask, "What are they thinking?"

Companies need to increase ROI without being taken to the cleaners by manufacturers selling servers featuring entry-level benefits with enterprise-level pricing.

Aberdeen gets it. Businesses are in desperate need of Network Attached Storage servers that simply deliver the best bang for the buck.

Look at these features and benefits:

	Dell PowerVault	HP StorageWorks	Aberdeen AberNAS
Hot-Swap Disk Drives	✓	✓	✓
Hardware RAID	✓	✓	✓
Dual Port Gigabit Ethernet	✓	✓	✓
Built-in Replication	✓	✓	✓
Microsoft® WSS 2008 Models	✓	✓	✓
iSCSI Target	✗	✓	✓
Linux Storage System Models	✗	✓	✓
System Recovery Disk	✗	✓	✓
DAS Storage Expansion	✗	✓	✓
VMware® Ready Certified	✗	✗	✓
Independent OS Drive	✗	✗	✓
Out of Band RAID Management	✗	✗	✓
Available w/ 2TB Drives	✗	✗	✓
Warranty	3 Years	3 Years	5 Years

Who gives you the best bang for the buck?

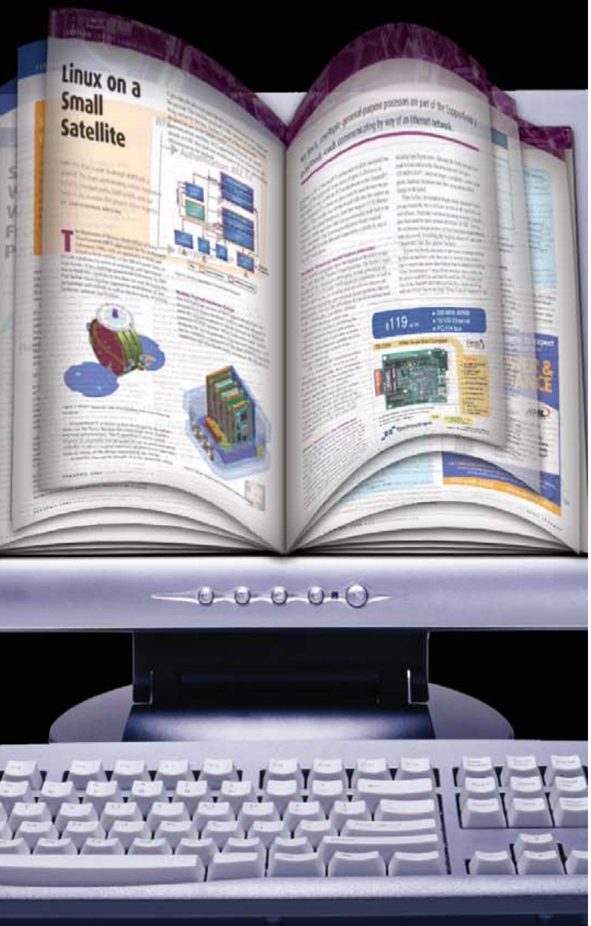
	Dell PowerVault NX300	HP StorageWorks X1400	Aberdeen AberNAS 163
Intel® Xeon® Processor	E5504 2GHz	E5504 2GHz	E5504 2GHz
Memory	3GB	2GB	3GB
Drive Interface	SATA	SATA	SATA
Installed Capacity	2TB	2TB	2TB
Rails	Included	Included	Included
Windows Storage Server 2008	$3,419	$4,635	$2,995
Linux Storage System	Not Available	Not Available	$2,995

intel® Xeon® inside™

Powerful. Intelligent.

Prices for the above specific configurations obtained from the respective websites on Oct. 12, 2009. Intel, Intel Logo, Intel Inside, Intel Inside Logo, Pentium, Xeon, and Xeon Inside are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries. VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. For terms and conditions, please see www.aberdeeninc.com/abpoly/abterms.htm. lj032

888-297-7409
www.aberdeeninc.com/lj032



LINUX JOURNAL

Executive Editor	Jill Franklin jill@linuxjournal.com
Senior Editor	Doc Searls doc@linuxjournal.com
Guest Editor	David A. Lane, KG4GIY kg4giy@arrl.net
Associate Editor	Shawn Powers shawn@linuxjournal.com
Associate Editor	Mitch Frazier mitch@linuxjournal.com
Art Director	Garrick Antikajian garrick@linuxjournal.com
Products Editor	James Gray newproducts@linuxjournal.com
Editor Emeritus	Don Marti dmarti@linuxjournal.com
Technical Editor	Michael Baxter mab@cruzio.com
Senior Columnist	Reuven Lerner reuven@lerner.co.il
Security Editor	Mick Bauer mick@visi.com
Hack Editor	Kyle Rankin lj@greenfly.net
Virtual Editor	Bill Childers bill.childers@linuxjournal.com

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

Proofreader	Geri Gale

Publisher	Carlie Fairchild publisher@linuxjournal.com
General Manager	Rebecca Cassity rebecca@linuxjournal.com
Sales Manager	Joseph Krack joseph@linuxjournal.com
Associate Publisher	Mark Irgang mark@linuxjournal.com
Webmistress	Katherine Druckman webmistress@linuxjournal.com
Accountant	Candy Beauchamp acct@linuxjournal.com

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

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

Subscriptions
E-MAIL: subs@linuxjournal.com
URL: www.linuxjournal.com/subscribe
PHONE: +1 818-487-2089
FAX: +1 818-487-4550
TOLL-FREE: 1-888-66-LINUX
MAIL: PO Box 16476, North Hollywood, CA 91615-9911 USA
Please allow 4–6 weeks for processing address changes and orders
PRINTED IN USA

LINUX is a registered trademark of Linus Torvalds.

PRINTED WITH SOY INK®

SCALE 8x

The Eighth Annual

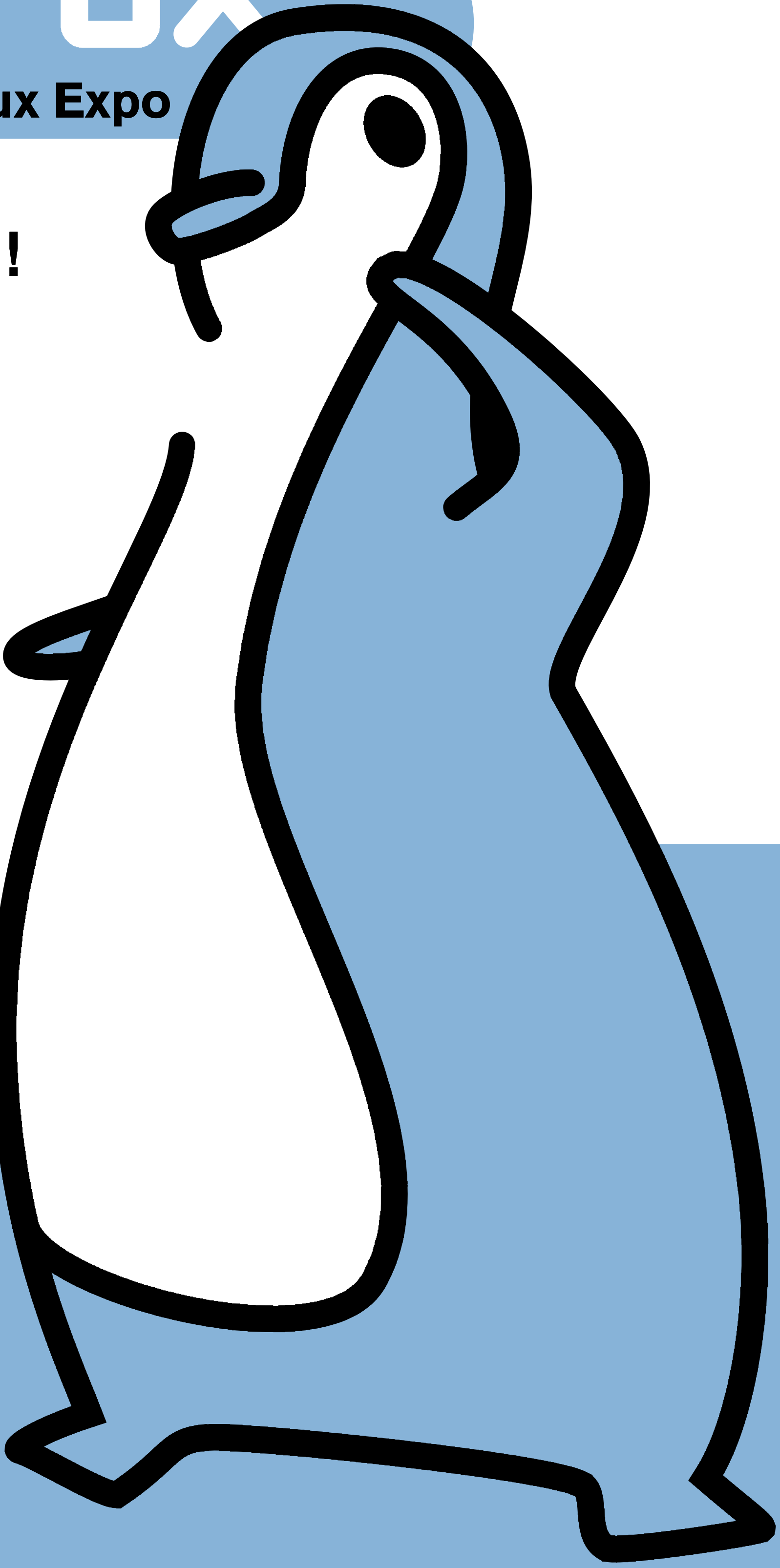
Southern California Linux Expo

Mark your calendars!

The 8th Annual
Southern California
Linux Expo
is coming!

Five session tracks!
More speakers!
Same great location!

February 19-22, 2010
Westin LAX
Los Angeles, California



http://www.socallinuxexpo.org for more info

Use Promo code LJAD for a 30% discount on admission to SCALE

DAVID A. LANE
GUEST EDITOR

When All Else Fails—Amateur Radio, the Original Open-Source Project

"When all else fails"—in 2003, the Amateur Radio Relay League used this as the motto for Field Day, the annual demonstration of its capabilities to the public. It rapidly became the touch phrase for the Amateur Radio Emergency Service—probably the most public of all aspects of Amateur Radio and the operators that are usually first to respond in an emergency. To me, it also is the quintessential definition of open source. When something is missing in a commercial system, we look to open source for the solution. A better operating system? A better management platform? How about a better ERP system? Open source has become the answer. But, what about a better communications system? When we think *open source*, our minds naturally turn to software, usually *NIX-based, running on Linux. What most people tend not to think about is the open-source nature of Amateur Radio. While operators most often are seen working in emergency situations, many of the modern conveniences we have today—cell phones, satellites, wireless devices—were developed and tested by radio amateurs.

What's Amateur Radio? Have you heard of ham radio? It's the same thing. Although some revel in the term, get a group of operators together, and you will find more stories for the origins of the term *ham radio* than operators. But, because the Federal Communications Commission calls us Amateurs Radio operators, we should call ourselves that too. Of course, this is not amateur in the rookie sense of the term, but in the non-pecuniary sense. Many consider us communication professionals.

Amateur Radio has been around almost since Marconi invented the thing—hmm, was Marconi a ham? Feel free to discuss that among yourselves. Early on, Marconi's invention was put to the test, and ever since, when disaster strikes, Amateur Radio operators have been some of the first to respond, providing communications support when there is little or no communications infrastructure. Before we can talk about it, however, we need some basic language, and like most technologies, Amateur Radio has a language all its own. This month, Dan Smith gets us started by explaining some of the shorthand we use.

Suppose I want to send data from one point to another without a wire between them? Shawn just passed me an 802.11a/b/g/n device, and we are off to the races, right? But, what if there is no network to plug it in to? Gary Robinson shows us how to pass data without an 802.11a/b/g/n device using your PC, Fldigi and an Amateur Radio transceiver. This is one of the ways that FEMA and other aid agencies send supply lists, like how many cell towers need to be brought into an area during the early stages of an emergency when the only ones passing messages are the Amateurs.

Got a GPS? How about 30,000 runners, more than half of whom will not make it to the end of the marathon, whether it is the Seattle, Boston, New York or Marine Corps Marathons? Using Xastir, an open-source version of APRS, a GPS and a rig, you can tell where the straggler bus is to pick up the runners who do not finish. You will find Amateur Radio operators at each of these marathons providing communications support. Curt Mills, Steve Stroh and Laura Mills take us through setting up and configuring Xastir so you can use it at your next event.

This month, however, we are not all about Amateur Radio. Dave Taylor tells us how to automate Twitter feed responses with a command-line bot (which begs the question, how many of Dave's responses are *not* automated?). Over at the forge, Reuven puts down his mallet and introduces us to Cucumber, an integration testing framework for Ruby, complete with vegetable jokes.

It looks like Kyle Rankin needs to buy a vowel as Dr hjkl dives into the Vimperator. (Sounds like what I did with my Christmas turkey, actually.) But seriously, I think Kyle has it in for the humble mouse as he shows you how to browse in Firefox using your keyboard.

We also have a security roundup, starting with Mick Bauer's discussion of the potential threats for the new year. Dirk Elmendorf tackles security from the amateur position (think Amateur Radio, not rookie). And, just to make sure you have all the security tools we can give you, Vadim Kurland describes what is new in Firewall Builder 3.0. Finally, Federico Kereki helps us implement some port-knocking security with knockd, and Doc Searls has the last word, opining on the control of one's personal data.

2010 is going to be another big year for virtualization, and Matthew Hoskins dives into virtual appliances with Xen on Linux. I can think of several projects for this technology already!

Whew, it's a jam-packed issue! Whether or not you are an Amateur, there is something in here for everyone. If you are interested in becoming an Amateur Radio operator, here in the US, visit **www.hello-radio.org**; in Canada, **www.rac.ca**; and in the UK, **www.rsgb.org**. For other countries, check with your local communications ministry or department, because *When all else fails....*■

David A. Lane, KG4GIY, has been licensed as an Amateur Radio operator since 2000 and has been working with Linux since 1995. During the day, he is an Infrastructure Architect. During an emergency, he is an Emergency Coordinator for Prince William County ARES. And on weekends, he makes pasta!

USENIX Upcoming Conferences

First Workshop on Sustainable Information Technology (SustainIT '10)

Co-located with FAST '10

FEBRUARY 22, 2010, SAN JOSE, CA, USA
http://www.usenix.org/sustainit10

2nd USENIX Workshop on the Theory and Practice of Provenance (TaPP '10)

Co-located with FAST '10

FEBRUARY 22, 2010, SAN JOSE, CA, USA
http://www.usenix.org/tapp10

8th USENIX Conference on File and Storage Technologies (FAST '10)

Sponsored by USENIX in cooperation with ACM SIGOPS

FEBRUARY 23–26, 2010, SAN JOSE, CA, USA
http://www.usenix.org/fast10

3rd USENIX Workshop on Large-Scale Exploits and Emergent Threats (LEET '10)

Co-located with NSDI '10

APRIL 27, 2010, SAN JOSE, CA, USA
http://www.usenix.org/leet10
Submissions due: February 25, 2010

2010 Internet Network Management Workshop/ Workshop on Research on Enterprise Networking (INM/WREN '10)

Co-located with NSDI '10

APRIL 27, 2010, SAN JOSE, CA, USA
http://www.usenix.org/inmwren10
Paper registration due: February 5, 2010

9th International Workshop on Peer-to-Peer Systems (IPTPS '10)

Co-located with NSDI '10

APRIL 27, 2010, SAN JOSE, CA, USA
http://www.usenix.org/iptps10

7th USENIX Symposium on Networked Systems Design and Implementation (NSDI '10)

Sponsored by USENIX in cooperation with ACM SIGCOMM and ACM SIGOPS

APRIL 28–30, 2010, SAN JOSE, CA, USA
http://www.usenix.org/nsdi10

2nd USENIX Workshop on Hot Topics in Parallelism (HotPar '10)

Sponsored by USENIX in cooperation with ACM SIGMETRICS, ACM SIGSOFT, ACM SIGOPS, and ACM SIGARCH

JUNE 14–15, 2010, BERKELEY, CA, USA
http://www.usenix.org/hotpar10
Submissions due: January 24, 2010

8th Annual International Conference on Mobile Systems, Applications and Services (MobiSys 2010)

Jointly sponsored by ACM SIGMOBILE and the USENIX Association

JUNE 14–18, 2010, SAN FRANCISCO, CA, USA
http://www.sigmobile.org/mobisys/2010/

USENIX Technical Conferences Week

2010 USENIX Annual Technical Conference (USENIX ATC '10)

JUNE 23–25, 2010, BOSTON, MA, USA
http://www.usenix.org/atc10
Submissions due: January 11, 2010

USENIX Conference on Web Application Development (WebApps '10)

JUNE 23–25, 2010, BOSTON, MA, USA
http://www.usenix.org/webapps10
Paper titles and abstracts due: January 4, 2010

3rd Workshop on Online Social Networks (WOSN 2010)

JUNE 22, 2010, BOSTON, MA, USA
http://www.usenix.org/wosn10
Paper submissions due: February 18, 2010

2nd USENIX Workshop on Hot Topics in Cloud Computing (HotCloud '10)

JUNE 2010, BOSTON, MA, USA

19th USENIX Security Symposium (USENIX Security '10)

AUGUST 11–13, 2010, WASHINGTON, DC, USA
http://www.usenix.org/sec10
Submissions due: February 5, 2010

USENIX: The Advanced Computing Systems Association

TECHNICAL SESSIONS AND TRAINING PROGRAM INFORMATION AND HOW TO REGISTER ARE AVAILABLE ONLINE AND FROM THE USENIX OFFICE:
http://www.usenix.org/events | Email: conference@usenix.org | Tel: +1.510.528.8649 | Fax: +1.510.548.5738

letters

RSpec | Virtualization | iSCSI SAN | AMQP | IPv4 Anycast | HIPL
LINUX JOURNAL
Virtualization Shootout
Solve Mobility Problems with HIPL
How Big Is Your Infrastructure?
Improve Availability with Anycast
Use AMQP for Enterprise Messaging
Storage Speed and Redundancy with iSCSI SAN
PLUS:
POINT/COUNTERPOINT: Ext3 vs. XFS

KVM and VNC

In Bill Childers' "Virtualization Shootout: VMware Server vs. VirtualBox vs. KVM" in the November 2009 issue, there's a mistake: KVM does support headless VNC. That's the way I use it in a Debian server to have a Win Server VM for hosting an antivirus server for Win PCs, not available for GNU/Linux.

--
Dani Gutiérrez

Bill Childers replies: *I indeed made a mistake when doing the research for this article. (Don't let Kyle Rankin know, I'll never hear the end of it!) Turns out, you can get a VNC-enabled console simply by adding a* `-vnc :1` *switch when starting your KVM virtual machine. Sadly, when doing research for the article, my Google-fu failed me—I kept getting instructions for the "KVM" type of keyboard/video/mouse switches. I'm glad it supports headless mode, and I'm sorry I missed that in my research. However, this does highlight the issue that KVM's usability isn't where it should be to compete with VirtualBox and VMware on that playing field. This is a moving target though, and I'm confident the developers will continue to improve KVM.*

Ruby Articles

I have been a dedicated *Linux Journal* reader for many years and generally like the mixture of different topics. However, in recent months, I've grown more and more annoyed with the series of Ruby articles. Reuven M. Lerner is doing a good job. It is not about the quality of his work; it is about choosing the right topics for the magazine. Ruby is not by any means the most popular language (**www.langpop.com**) in either the open-source or commercial space. I think it is time to yield magazine space to other programming languages.

--
MK

You are absolutely correct in regard to Ruby not being the most popular language in use. I think it might be the current state of the language's maturity that is sparking so much interest. I'm not a programmer at all (well, the occasional bash script, but that doesn't count), so I asked a similar question a while back. Reuven's column is still very highly rated by our readership. Although Ruby may not be the most popular language, it does appear to garner a lot of interest. I can only assume that, in time, it will pass, and the next fancy language will take its place. In the meantime, thank you for the letter. I'm sure Reuven will read it and keep an ear out for what is most sought after.—Ed.

VirtualBox

I recently installed Sun's VirtualBox after reading the "Virtualization Shootout" article in the November 2009 issue. It's a great tool that allows me to keep Windows set up for those occasions when I can't figure out how to do something in Linux. As a relative newbie who works in an all-Microsoft environment, I wasn't getting the exposure to Linux I wanted, because I had to dual-boot between the two OSes. Virtualization gives me the best of both worlds. So, now I boot only to the Ubuntu Linux host and open my virtual Windows XP guest system when needed.

I love your magazine and look forward to every issue. I read all the columns, most of the reviews and at least scan the Indepth articles. Every month, I find new information and ideas I use to grow in my understanding of Linux. Next year, I plan to install Linux from scratch for the knowledge it will bring. Thanks again for the all information and expertise you pack into every issue.

--
Paul West

Zarafa

I know you get lots of e-mail messages saying "why don't you write about ...", but I will add this for the record: take a look at Zarafa—it's a fully native Linux-compatible GPLv3 groupware alternative for Exchange *with* Outlook support at no extra cost!

I have spent approximately four years researching GPL groupware solutions and have tried several (Scalix, eGroupWare, Zimbra, Citadel and Open-Xchange), but none of them come close to matching the features and support of this app. It does calendaring, task lists, contacts, e-mail, BlackBerry and mobile phone sync. It's easy to deploy, and it sits on top of your own server's MTA.

I don't work for the company, nor do I sell any of its stuff, but it is a very cool application! Check it out, and maybe you could do a review for your readers (**www.zarafa.com**).

--
Trent Murray

Darn it Trent, now you've given me another toy to play with! Thanks for the tip; I'll be sure to check it out. I too have been looking for a full-featured groupware option.—Ed.

Reader Ramblings

I am a bit of a newbie to Linux, although some of my computing days go back to systems using 12AT7 (tubes!) as memory elements. I am also a radio amateur, as you probably can guess from my e-mail address. I have had a few pleasant surprises since switching from the offerings coming out of Washington state. I recently connected an ICOM 7200 to an Ubuntu box and found the USB drivers were already present. Nice! I installed Fldigi, changed the CI-V address on the 7200 to 6E, set Fldigi for Hamlib mode and told it I was using an ICOM 756ProIII (there is no definition yet for the 7200), and all things worked. Try that with the competition! I could add a few other nice things, but this letter would be a bit long.

Incidentally, I got my first copy of *Linux Journal* on a draw from the Manitoba UNIX User Group, of which I am a member.

I became rather disgusted with you know who a few years ago, after finding that Encarta 2001 would not run under XP. It seems XP does not maintain backward compatibility with earlier releases. I get the impression that a certain company has the idea of releasing a new OS version every few years, with no capability of running on anything less than the latest hardware and then discontinue the support for the older OS, and everybody has to upgrade. $$$$! I realize that the days of the 4.77MHz XT are long gone, but a system using a 1.5GHz Intel P4M isn't too shabby, and a version from a well-known manufacturer apparently will have problems with the new "7" when it comes out.

One thing I would like to see as a beginner is a "newbies column", where those of us who are green at Linux can learn the fundamentals of the system. *Ubuntu Kung-Fu* by Keir Thomas was another prize and a big help to me.

--
Gerald K. Sherman

I'm a fairly geeky guy, but I must admit you out-geeked me with Fldigi'd Hamlibs and such! But as luck would have it, this issue's Features are all related to Amateur Radio—hope you enjoy the articles. We try to keep the content in the magazine diverse enough to reach everyone, but I'll admit some of the articles are even over my head. You should check out our Web site, which has many more articles and some helpful videos often aimed at new users. Either way, we'll continue to try reaching all user levels in the print magazine. If Linux Journal starts to no longer be useful and fun, be sure to call us out!—Ed.

awk Tip

This is in reference to Dave Taylor's column in the April 2009 issue on word and letter counts. The filter used for counting occurrences of patterns on stdin is:

```
sort  | uniq -c
```

I appreciate the beauty of this filter owing to its simplicity. In practice though, I find this becoming slower and slower as the data size increases and the cost of sorting increases. I use the following code, which does the job extremely fast:

```
awk ' { CNT[$0]++ }
END { for (i in CNT) print i, CNT[i] }
'
```

One also can derive probability of occurrence easily as:

```
END { for (i in CNT) print i, CNT[i]/length(CNT) }
```

I appreciate that the focus of this column is mainly on the shell, although I felt that a couple lines of awk is not bad for the above problem, as it avoids sort, which can be costly at times. (I'm not sure whether the use of associative arrays in bash would be equally compact.)

--
Mayuresh Warunjikar

Dave Taylor replies: *Good use of awk, Mayuresh. My goal is always to show what's quick and fast—not always what's most efficient or highest performance. Sometimes that means it doesn't scale well, but frankly, my view of shell script programming is that most scripts break down once you really push hard on them anyway, at which point it's smart to rewrite it into a faster, more powerful app.*

msfetch

Recently, I created a tool to help sysadmins like myself—ones that love Linux!—to be able to download their Microsoft Exchange mail using HTTP Outlook Web Access. It downloads e-mail, formats e-mail properly to match .mbox file format and appends additional headers. Then, you can view your mail through the console. Maybe it will interest some of your readers. It is absolutely free, GPLv3, and it works very well. See **www.isrcomputing.com/msfetch**. Love your journal, by the way.

--
Paul Grinberg

Neat Paul! We'll have to check it out. Thanks for the link.—Ed.

Closed Captioning

Regarding the letter asking for transcripts of the video Tech Tips on LinuxJournal.com in the November 2009 issue: I agree that they would be pretty much useless; however, CC (closed captioning) would be *very useful*! Those us us who would like to watch a very relevant and work-related video in our office without speakers and Linux users who are hearing impaired really would appreciate captions. Please consider adding captions to your otherwise awesome videos.

--
Jon Harrell

You're not the first (or second, or third, or twelfth) person to ask for something like this. I will keep trying to figure out a way to make closed captioning, or something similar, a feasible addition to the Tech Tips. Perhaps we should open it up to the readership to submit CC files to go along with the OGG videos. In fact, that's a really neat idea. I'm glad we had this talk!—Ed.

clac

In “Exploring Lat/Lon with Shell Scripts” [November 2009], Dave Taylor writes, “To accomplish any sophisticated mathematics in a Linux shell, we’re pretty much stuck with bc.” No! Shed your stone tools! Throw off your bc and dc chains! There is a better way. The clac utility is available at **clac.sourceforge.net**, and it has the following features:

- Handles trig, complex math, log and so on.
- Supports infix expressions (no Reverse Polish jokes, please).
- Uses Python underneath (Python syntax welcome, but not required).
- Single source file—take it where you need it.
- User-extensible via Python: import/write the modules you want.
- Script-friendly: expression(s) via arguments or stdin, evaluation printed to stdout.

clac takes expressions like:

```
"sin(pi/4)"
"round( degrees(phase( e** ( 2j ) ) ) )"
"sum([ x**2 for x in ( $CSV ) ] )"
```

and does what you would expect a CLI tool to do: prints out the answer in a fraction of a second.

--
Mark Allen Borgerding

Dave Taylor replies: *Cool, thanks for bringing that to our attention!*

PHOTO OF THE MONTH

Have a photo you’d like to share with *LJ* readers? Send your submission to publisher@linuxjournal.com. If we run yours in the magazine, we’ll send you a free T-shirt.

Here’s a photo of my five-year-old daughter who found your issue on Linux security especially interesting—submitted by Mattias Villani.

LINUX JOURNAL

At Your Service

MAGAZINE

PRINT SUBSCRIPTIONS: Renewing your subscription, changing your address, paying your invoice, viewing your account details or other subscription inquiries can instantly be done on-line, **www.linuxjournal.com/subs**. Alternatively, within the U.S. and Canada, you may call us toll-free 1-888-66-LINUX (54689), or internationally +1-818-487-2089. E-mail us at subs@linuxjournal.com or reach us via postal mail, Linux Journal, PO Box 16476, North Hollywood, CA 91615-9911 USA. Please remember to include your complete name and address when contacting us.

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

LETTERS TO THE EDITOR: We welcome your letters and encourage you to submit them at **www.linuxjournal.com/contact** or mail them to Linux Journal, 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**.

Polywell Linux Solutions

More Choices, Excellent Service, Great Prices!

Serving the Industry for More Than 20 Years

Quiet Storage NAS/SAN/iSCSI

8TB $1,999

12TB $2,599

30TB $6,599

- Dual Gigabit LAN
- RAID-5, 6, 0, 1, 10
- Hot Swap, Hot Spare
- Linux, Windows, Mac
- E-mail Notification
- Tower or Rackmount

The Best Terminal PC

LD-001

Silent Eco Green PC

NVIDIA® GeForce Graphics
Energy efficient, Quiet and Low
Voltage Platform. starts at $199

Mini Fanless PC

ITX-10A Mini-PC $199

with Solid State Drive, DC Power

ITX-20A with SlimDVD

ITX with VESA / Wallmount

ITX-30A with PCI Riser

ITX-1000C with 4LAN
and WiFi Option

Nvidia® ION™ PC
ITX-40A with Blu-Ray or DVD

5U-48Bay 96TB Storage Server

5048A

4U-24Bay 48TB Storage

Hardware RAID-6, NAS/iSCSI/SAN Storage
Mix SAS and SATA, 4 x GigaLAN or 10Gbit LAN

4024SS

Mini-1U Internet Server $499

Intel Dual-Core or Quad-Core Processor,
Dual GigaLAN, 4GB to 8GB RAM,
2 x 500GB RAID HD

1002S-1U945GCL2

We Custom Design Chassis and
Motherboard for OEM Customers

Polywell OEM Services, Your Virtual Manufacturer
Prototype Development with Linux/FreeBSD Support
Small Scale to Mass Production Manufacturing
Fulfillment, Shipping and RMA Repairs

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

888.765.9686
linuxsales@polywell.com
www.polywell.com/us/Lx

POLY
POLYWELL

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

NVIDIA, ION, GeForce and combinations thereof are trademarks of NVIDIA Corporation. Other names are for informational purposes only and may be trademarks of their respective owners.

diff -u

WHAT'S NEW IN KERNEL DEVELOPMENT

The **Linux scheduler** is pretty controversial, in large part because contributors can test their code and find it to be faster than someone else's code, and that other person can run tests showing the exact opposite! It all depends on what you test for. Some tests examine only how many processes can be created in the shortest amount of time. Other tests try to mimic interactive behavior and examine how smooth that behavior is. When you add to this the fact that the scheduler needs to work as well as possible for the full range of computer hardware that might be running Linux, the problem becomes far more complex. It's no small wonder that disagreements become heated, and some contributors choose to go off and code on their own and abandon the quest to elevate their scheduler to the status of Official Linux Scheduler.

In that spirit, **Con Kolivas** has been quietly writing **BFS** (Brain F*** Scheduler), incorporating everything he's learned about schedulers over the years. But, his work recently got the attention of **Ingo Molnar**, maintainer of the current in-kernel scheduler, and the ensuing discussion highlighted some of the issues surrounding scheduler coding. For example, one of Con's main goals with his scheduler was to make the user interface as smooth as possible for desktop systems—that is, systems with eight CPUs or thereabouts. He wanted to play games, listen to music and do other highly interactive things, without having a jerky display. His technique was (among other things) to abandon support for systems that had thousands of CPUs onboard.

Clearly, Linux has to support both small and large systems, so Con's scheduler wouldn't work as the primary in-kernel process scheduler. And, because **Linus Torvalds** has said he doesn't want to include multiple schedulers in the kernel, it's unlikely that Con's code ever could be included in the official tree. So he's been coding it up on his own, working with a small group of interested users who don't mind patching their kernels to get the smoothest possible performance.

It seems like a very cool project. Maybe at some point, his code will be offered as an option to desktop users. But for the moment, the kernel has to stay with the scheduler that supports the most systems and that attempts to have the broadest possible range of decent performance.

The **VMware** folks are discontinuing **VMI** support. Virtualization features in hardware are making it pointless to try to do the same thing in software. Hardware virtualization probably will be everywhere soon, they say, so there's no reason for VMI to emulate multiple CPUs in software anymore. **Alok Kataria** asked on VMware's behalf about the best way to deprecate and remove the VMI code from the kernel. As it turned out, the best way is going to be to keep supporting it for another year or so, until it truly has no more users. In the meantime, VMI is being added to various lists of features that are soon to be removed, and code is being added to alert users that they should brace for the time when VMI no longer will be available. Kudos to VMware for working with the kernel folks to find the best way to do this. It's nice to see industry paying attention to open-source methods.

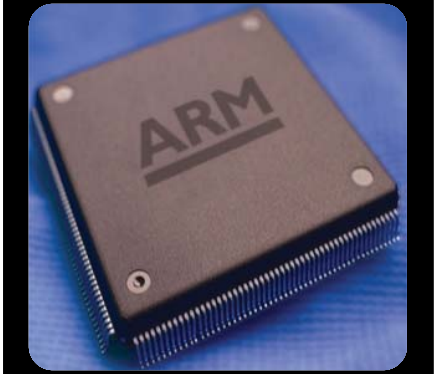
CPUidle is a new project by **Arun R. Bharadwaj** that attempts to save power on PowerPCs by recognizing when CPUs are idle enough to put into low power modes. It does this by analyzing the current system usage and essentially making educated guesses about what's going to happen in the near future. In addition to being a cool and useful project, we can look forward to many future flame wars surrounding exactly how to determine when to put a CPU into a low power mode, as hardware designs and software usage patterns become more and more complex.

Jakub Narebski took a survey of **Git** users and made the results available at **tinyurl.com/GitSurvey2009Analyze**. Thousands of people participated, virtually all of whom are happy with it. It's interesting to see that 22% of respondents use Git under **Windows**. If you haven't tried Git yet, or if it's been a while, you really should check it out. It takes a problem that stumped many people for years and makes it look easy. The flame wars, the bitter competition between projects and the venomous **BitKeeper** disputes have all melted away. Git is a tool of peace.

—ZACK BROWN

Ultraportables That Cost an ARM, But No Leg

Whether you love Netbooks or hate Netbooks, there's no doubt that Intel currently owns the market when it comes to providing the hardware. Unfortunately, Intel's attempts to make CPUs that consume less power also have resulted in Netbook computers that are painfully slow. And, don't get me started on the GMA500 video chipset. (Actually, feel free to read my rant on the *Linux Journal* Web site regarding the GMA500 chipset at **www.linuxjournal.com/content/how-kick-your-friends-face-gma500**.)

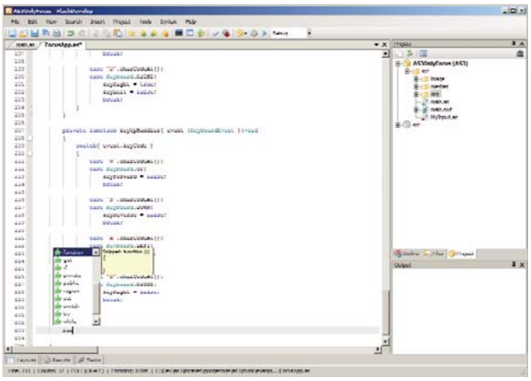


Thankfully for users, while Intel works to tweak its Atom line of processors, other companies are working to enter the Netbook market too. Unfortunately for Intel, one particular company has more experience with low-power, high-performance computing than Intel could ever fathom—ARM. Ironically, while Intel keeps pushing its clock speeds down to save a bit of power, ARM has been ramping its speeds up. Intel's latest Atom processor, at the time of this writing, has a whittled-down speed of 1.3GHz; the new ARM-based CPUs are around 2GHz.

Certainly, Intel has reason to fret as the embedded market, which is dominated by ARM, begins to scale up to the Netbook market. The real fear, however, should be that once ARM invades the Netbook and notebook markets, there is no reason for it to stop. Imagine racks of servers with ARM-based CPUs operating at Intel-class speeds while sipping electricity and staying cool enough to run without heatsinks. Oh yeah, guess who else might be a little afraid? That's right. Windows doesn't run on the ARM architecture, but Linux sure does.

—SHAWN POWERS

NON-LINUX FOSS



FlashDevelop Code Snippets (from flashdevelop.org)

FlashDevelop is a free and open-source code editor designed for developing Flash applications. It supports code snippets, code completion, code navigation, code generation, bookmarks, task handling and syntax highlighting for ActionScript 2 and 3, MXML and haXe. It contains an integrated Project Manager and project templates to get your projects off the ground quickly.

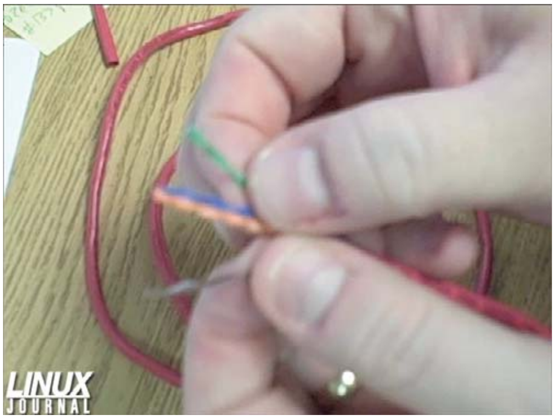
FlashDevelop integrates with the Flash CS3, Flex SDK, Mtasc and haXe compilers. It can execute the compiler and parse error/warning output for quick access to the offending code. FlashDevelop has a plugin architecture for extending it and integrating it with other tools as well.

The editing power behind FlashDevelop is provided by the open-source Scintilla editor library. The Scintilla library was first released in 1999 and provides the editing capabilities in a number of open-source projects, such as Notepad++, Notepad2 and Open Komodo.

FlashDevelop is a .NET 2.0 application, and it runs on Windows XP and newer versions of Windows. Currently, the only way to run it on Linux or OS X is via virtualization software—which is to say, it currently will not run on Mono. FlashDevelop is released under the MIT License. The current version is 3.0.4.

—MITCH FRAZIER

On-line Videos: Now Out of the Box!



At the *Linux Journal* Web site, we've been providing tech tips for quite some time. Although there seems to be no end to the tips we come up with (thanks in part to user contributions), lately we've decided to step it up a notch. We certainly will continue to provide command-line tips, GUI shortcuts and unconventional Linux tips, but we're also adding some practical real-world tips as well. Whether you need to brush up on cable-crimping skills or figure out how to track down a rogue DHCP server, you'll want to catch our new tech tips.

As always, you can find our videos on our main Web site: **www.linuxjournal.com**. See you there!

—SHAWN POWERS

LJ Index January 2010

1. Year that the :) emoticon was invented (September 19): **1982**

2. Year the Internet was first proposed (May 22): **1973**

3. Twitter year-over-year percent growth rate (February 2008 to February 2009): **1,382**

4. Twitter monthly visitors in millions: **7.0**

5. Facebook monthly visitors in millions: **65.7**

6. MySpace monthly visitors in millions: **54.1**

7. Club Penguin (Disney not Linux) monthly visitors in millions: **6.1**

8. Percent of Twitter users that fizzle out during the first month: **60**

9. Percent of Twitter users that fizzled out before Oprah: **70**

10. Percent of Facebook and MySpace users that fizzle out: **30**

11. Percent of US farms that had Internet access in 1999: **20**

12. Percent of US farms that had Internet access in 2007: **57**

13. Percent of US farms that had Internet access in 2009: **59**

14. Percent of US farms that use satellite for Internet access: **13**

15. Percent of US farms that use cable for Internet access: **13**

16. Percent of crop farms that use computers in their business: **45**

17. Percent of livestock farms that use computers in their business: **37**

18. Number of *LJ* issues since the first National Debt figure appeared (September 2008): **17**

19. Change in the debt since then (trillions of dollars): **$2.54**

20. Average debt change per issue (billions of dollars): **$149.7**

Sources: *1, 2: www.distant.ca/UselessFacts / 3–10: Nielsen Online / 11–17: NASS (National Agriculture Statistics Service) / 18–20: Math*

Finding xargs

In the November 2009 issue, I looked at some tools used by computational scientists when they are doing pre- and post-processing on their data. This month, I examine ways to automate this work even further. Let's start with the assumption that we have a number of steps that need to be done and that they need to be done on a large number of data files.

The first step is to find out which files need to be processed. The most obvious way to do that is simply to create the list of files manually. Doing it this way is the most time-intensive, and as with all manual operations, it can lead to mistakes due to human error. This step lends itself to automating, so it should be automated. The most popular utility for creating lists of files is the find command, which finds files based on a number of different filters.

The most common filter is the filename. You can use find to locate where a certain class of files lives on your drive. Let's say you have a group of data files that need to be processed, all ending in .dat. The find command to search for them would be the following:

```
find . -name "*.dat" -print
```

The first option tells find where to start looking for files. In this case, you are telling find to start looking in the current directory. You also could give it any directory path, and it would start its search at that location. The second option, `-name`, tells find what to filter on. In this case, you are telling find to filter based on the filename. This filter is actually a filename pattern, `"*.dat"`. This pattern needs to be quoted so the shell doesn't try to expand the * character. The last option, `-print`, tells find to print the search results to standard output. You can redirect this to a file or pipe it to another command. By default, find separates the filenames with new-line characters. A problem can occur if you have odd characters in your filenames, like blanks or new lines, and you feed this as input to another program. To help mitigate that issue, use the `-print0` option instead of `-print`. This tells find to separate the results with a null character rather than a new-line character.

There are many other filter options. The first one to be aware of is `-iname`. This is like `-name`, except that it is case-insensitive. There also is a group of filter options around time: `-amin n` and `-atime n` filter based on access time. `-amin n` looks for files that were accessed n minutes ago. `-amin +n` looks for files that were accessed more than n minutes ago, and `-amin -n` looks for files that were accessed less than n minutes ago. The `-atime n` option looks for files that were accessed n*24 hours ago. Again, +n and -n behave the same way. The same tests can be made using `-cmin n` and `-ctime n` to test the file's created time, and `-mmin n` and `-mtime n` test the modified time of a file. The last filter option to use in this instance is `-newer filename`. This option checks to see whether the files being searched for are newer than some reference file named `filename`. This might be useful if you want to generate a list of files that have been created since your last processing run.

Now that you have your list of files, what do you do with it? This is where you can use another command called xargs. xargs, in its simplest configuration, takes lines of input and runs some command for each line. This simplest use looks like this:

```
xargs -I {} run_command {}
```

Here, xargs reads lines of input from standard input. The `-I {}` option tells xargs to replace any subsequent appearances of `{}` with the line of input that was just read in. Using `{}` isn't mandatory. You can use any characters to act as placeholders. In this example, xargs executes `run_command {}` for each line of input, substituting the input line into the place held by `{}`. If you dump your list of files into a text file, you can tell xargs to use it rather than standard input by using the `-a filename` option. Also, xargs will break input on blanks, not just new-line characters. This means if your filenames have blanks in them, xargs will not behave correctly. In this case, you can use the `-0`, which tells xargs to use the null character as the delimiter between chunks of input. This matches with the `-print0` option to find from above. If you aren't sure whether the given command should be run on every file listed, you can use the `-p` option. This tells xargs to prompt you for confirmation before each command is run. The last option to xargs that might be useful is `-P n`. This option tells xargs to run n number of commands in parallel, with the default value for n being 1. So, if you have a lightly loaded dual-core machine, you might want to try using `-P 2` to see if the processing happens a bit faster. Or on a quad-core machine, you might want to try `-P 4`.

Putting this all together, let's look at a hypothetical job you may have to do. Let's say that you have a series of data files, ending in .dat, scattered about your home directory. You ran a processing job last week and named the file containing the list of filenames to be processed list1.txt. Now, you want to process all the data files that have been created since then. To do so, you could run the following commands:

```
find ~ -name "*.dat" -newer list1.txt -print >list2.txt
xargs -a list2.txt -I {} run_processing {}
```

This way, you would have a list of the files that had just been processed stored in the file list2.txt, for future reference. If some of the filenames contain blanks or other white-space characters, you would probably run the following instead:

```
find ~ -name "*.dat" -newer list1.txt -print0 >list2.txt
xargs -a list2.txt -I {} -0 run_processing {}
```

Hopefully, this helps you automate some of the routine work that needs to be done, so you can get to more interesting results. Go forth and automate.

—JOEY BERNARD

DRM, DMCA and OST (Other Scary Things)

It's obvious to most Linux users that Digital Rights Management (DRM) is a really bad idea. It's not just because DRM-encoded media usually won't play with our operating system, but rather because we understand the value of openness. The really sad part is that DRM, at least on some level, is attempting to do a good thing.

Take Wil Wheaton for example. You may know Wil from television or movie acting, but he's also a successful writer. Along with writing books, he often performs them and sells the audiobook version of his work. Because Wil is against DRM, his books are available as unrestricted MP3 files. That's great for those of us who like to listen to such things on Linux or a non-iPod MP3 player. Unfortunately, some users confuse DRM-free with copyright-free. When otherwise decent individuals don't think through the ramifications of redistributing someone's copyrighted work, you end up with situations like this: **tinyurl.com/stealfromwil**.

If you put yourself in Mr Wheaton's shoes for a second, you can see how tempting it is for authors—and, more important, publishing companies—to DRM-encode their work. Theoretically, if a piece of media is "protected" with DRM, only those people who purchase the rights to enjoy it can enjoy it. Unfortunately, as everyone but the big companies that insist on using it know, all it manages to do is punish the honest people. People who have tried to play Amazon Video on Demand videos on their Linux desktops or listen to Audible audiobooks on their no-name MP3 players know exactly what I mean.

The truth of the matter is, if people are dishonest enough to use copyrighted materials they haven't paid for, DRM does little more than give them an "excuse" for their pirating ways. Right now, users are given the choice of paying money for limited, restricted access to media, or to download illegally fully functional, cross-platform, unrestricted media for free from torrent sites. I have two messages for two different groups:

1. Media publishing companies: make it easy for users to buy access to your media, and make that media flexible, archive-able and affordable. Yes, people will pirate your stuff—just like they do now. The difference will be that your honest clients won't hate you, and you'll actually gain some clients because you will be offering what people really want.

2. Frustrated users: look, I'm with you. DRM frustrates me too. Although I'm not expecting to convert those among us who choose to pirate media, I would hope that we'd all support those companies (and individuals, in Wil Wheaton's case) that "get it". The only way we'll be part of the solution when it comes to eliminating DRM is actually to buy non-DRM stuff. At the very least, every time you pirate something because you can't buy it legitimately, e-mail the companies and let them know. If they see lost sales, perhaps they will rethink their approach to digital media.

I could go on and on about the insanity of Blu-ray DRM and the like, but I don't have the energy. Plus, I want to go watch the hi-def movie I just ripped on my Linux laptop. I'll have to remember to e-mail the movie producer about my morally justifiable, but legally questionable ways....

—SHAWN POWERS

More Ham Radio Options at LinuxJournal.com

Now that you are fully immersed in all things ham-related, join us on-line for more ham radio discussions at **www.linuxjournal.com/forums/hot-topics/ham-radio**. Correspond with other Linux enthusiasts who share an interest in Amateur Radio and discuss related open-source projects. Drop us a note in the forum, and let us know what projects you are working on, more about your radio interests and any advice you may have for other operators.

Further resources also are available at **www.linuxjournal.com/ham**. Whether you are a newcomer or an experienced ham, we hope you'll find LinuxJournal.com to be a valuable resource as you delve further into your radio operations.

—KATHERINE DRUCKMAN

They Said It

The Itanium approach...was supposed to be so terrific—until it turned out that the wished-for compilers were basically impossible to write.
—Donald Knuth

This continues to be one of the great fiascos of the last 50 years.
—John C. Dvorak, in "How the Itanium Killed the Computer Industry" (2009)

We decided to innovate our way through this downturn, so that we would be further ahead of our competitors when things turn up.
—Steve Jobs, as the 2001 recession was fading

IBM is the company that is notable for going the other direction. IBM's footprint is more narrow today than it was when I started. I am not sure that has been to the long-term benefit of their shareholders.
—Steve Ballmer (during the last ten years, IBM's share price has increased by 30%, and Microsoft's has decreased by 30%)

We're in it to win it....IBM, we're looking forward to competing with you in the hardware business.
—Larry Ellison (in a response to IBM's spreading of FUD over Sun's future after the Oracle buyout)

The longer this takes, the more money Sun is going to lose.
—Larry Ellison (Sun's revenues have fallen dramatically since the buyout was announced. Sun is losing about $100 million a month as European regulators delay approving the buyout.)

We act as though comfort and luxury were the chief requirements of life, when all that we need to make us happy is something to be enthusiastic about.
—Einstein

While we haven't won yet, Red Hat will continue fighting for the good of technology and for the good of innovation.
—Jim Whitehurst, President and CEO, Red Hat (referring to the Bilski case)

Cucumber

Cucumber's plain-text integration testing is as useful as it is innovative.

REUVEN M. LERNER

People used to say that open-source technologies were excellent at mimicking and copying proprietary projects, but that they had few original ideas of their own. This was never completely true, but I believe that today it is demonstrably false, and that it becomes increasingly false over time. In many respects, the bleeding edge of innovation in the computer industry is taking place within the Open Source community. If you're looking for the future of operating systems, network protocols, Web development or programming techniques, the odds are quite good that you'll find it by looking at open source.

Ruby on Rails, the well-known Web development framework, is such a technology; nearly every other language now has its equivalent of Rails. Rails certainly has been influential on other frameworks in many ways, from its use of MVC to its pervasive "convention over configuration" stance. But, Rails also is leading the way through its use and encouragement of automated testing. Testing has existed for some time in the computer world, but the Ruby community in general and the Rails community in particular are almost fanatical in their desires to test programs extensively, at multiple levels. Such testing, as many others have written over the years, gives you the confidence you need when changing, improving and just refactoring existing code.

For the past few months, I have been looking at a number of testing technologies that came from the Ruby and/or Rails worlds, which I believe are likely to have an impact on other, slower-moving languages and environments. The one thing all these tests have in common is that they are of interest primarily to the programmer. That is, the programmers working on a project all might agree on the need for testing and even on a framework for doing it. The nontechnical managers of a project, although they might benefit from such testing, don't participate in this process very closely, because they typically cannot do so. After all, even Shoulda's syntax, for all its simplicity over standard test/unit constructs, is still in code and, thus, is hard for nonprogrammers to read and understand.

Enter Cucumber, the oddly named but extremely powerful integration testing framework begun by Aslak Hellesoy. Cucumber is part of the BDD (behavior-driven design) school of thought, which argues that development should begin with a specification, and then the code will be written to match that specification.

Typically, in frameworks such as RSpec and Shoulda, the specification is written in code. Cucumber takes a different approach, making it possible to write specifications in English, or other natural languages, and for the computer to take care of translating those specs into an executable format. In this way, the specs still are executable, but they also are readable by nonprogrammers, making it easier to discuss at meetings or in documents. I haven't experienced this firsthand, but it should be possible for nonprogrammers to read the Cucumber stories and begin to execute them.

Cucumber can be used in a variety of testing contexts, but it is most commonly used by Rails programmers for integration tests, and many of the people using it are raving about the effect it has on their development process. This month, I introduce Cucumber, which I believe has the potential to change the ways integrated tests are written and executed dramatically, and which is another clever invention by a member of the Open Source community. I'll walk through the creation of some simple Cucumber tests, and I'll point out where it requires just a bit more ripening. (Vegetable jokes are a staple of the Cucumber community, for better or worse. So, if you plan to use Cucumber, it'll be useful if you find such humor a-peel-ing.)

Installing and Using Cucumber

Cucumber has undergone a number of rapid transformations in the past year alone, thanks in no small part to a growing number of contributors, as well as a great deal of excitement and exposure within the Ruby community. As a result, it is sometimes hard to keep up with version numbers and documentation.

Fortunately, the installation process for Cucumber remains fairly straightforward; it comes packaged as a Ruby gem, which means that you can install it with:

```
gem install cucumber
```

At the time of this writing, Cucumber is at version 0.4. Moreover, while Cucumber (and many other Ruby gems) have been hosted by GitHub (a well-known commercial repository for the Git version-control system), it recently was announced that GitHub no longer will support the creation of

Ruby gems. So, you might need to look around for Cucumber's official repository when you read this.

Once you have installed Cucumber, you need to add its functionality to your Rails application. You can do this with the following:

```
script/generate cucumber
```

This puts the appropriate Rake tasks into place (in lib/tasks/cucumber.rake), adds the initial default step definitions (that is, low-level test implementations) and the overall system support that is necessary for Cucumber to function. All of the files associated with Cucumber are put inside the features directory, which can be somewhat confusing to the uninitiated.

Once these new files are in place, you can run Cucumber as follows:

```
rake cucumber
```

Cucumber will run through any file with a .feature suffix in the features directory. If you just installed Cucumber, no such files will exist yet, and you will see output like this:

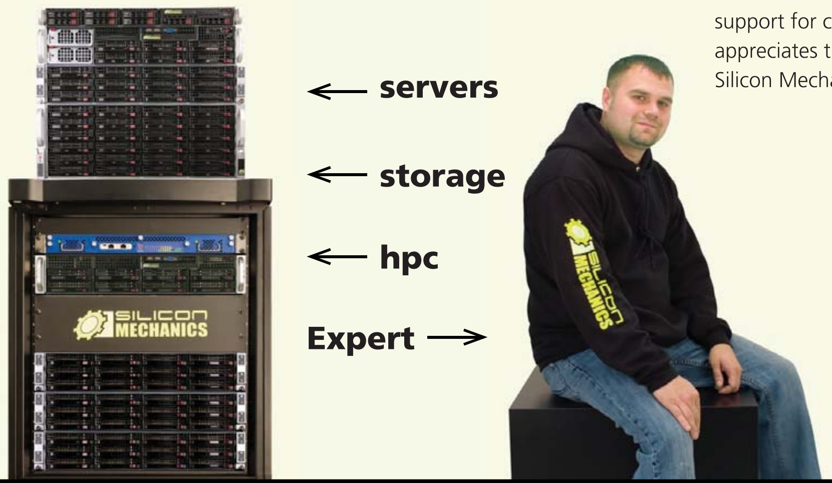
```
0 scenarios
0 steps
0m0.000s
Loaded suite /usr/bin/rake
Started

Finished in 0.000232 seconds.

0 tests, 0 assertions, 0 failures, 0 errors
```

This is similar to the output you would get from running `rake test` without any tests installed. So let's get started and write something.

Cucumber uses a different vocabulary for creating tests and specifications than you might be used to. Each Cucumber file describes a single "feature" of the application and has a .feature suffix. A feature typically will be a small slice of the application—anything from authentication, to sending or receiving messages, to producing a report. The feature traditionally is described with a line like the following:



```
Feature: Home page
```

As you can see, this file begins with the word "Feature:", and then contains a description. This description, like many others in Cucumber, appears in the output later on, as a helpful description.

Following the Feature declaration, you describe the feature, typically in the form of a "story", as used in many agile teams and in the BDD world in general. (If you are new to stories, I suggest reading Dan North's blog post on the subject; see Resources.) Here is a typical story:

```
As a user,
I want to be able to log in
So that I can use the system
```

The feature is then tested in a number of ways, each of which is known as a scenario. The idea is that each scenario describes a situation (Given) in which the user performs some actions (When), and then sees some results (Then). The scenario should be as specific as possible, testing a particular path through the application's interface.

Cucumber provides a wide variety of Webrat step definitions, such that you can tell Cucumber to go to a page, to fill in a form or to use selection lists, check boxes and radio buttons.

Note that the scenario is not meant to test one particular controller, model, library or other element of the code. Rather, the scenario should represent a particular action from the user's perspective, which might cover one controller and one model, or a dozen of each. It's normal and reasonable to have a number of scenarios for each feature. It also is reasonable to assume that the number of scenarios will grow over time, as you (and your users) stress the application in new and different ways, and you uncover bugs that need to be covered by new scenarios. A scenario consists of one or more steps, which are translated into working Ruby code. Here is a sample scenario:

```
Scenario: Get to the login screen

    Given a user named "Reuven" "Lerner" with an e-mail
    address "reuven@lerner.co.il"
    When I go to the home page
    Then I should see "Web site"
    Then I should see "Login"
```

I put this scenario (with its story and one feature) into features/login.feature, and then ran `rank cucumber`. Cucumber responded by going through the file, executing the scenario I had defined. Well, it tried to execute the scenario; here is what I actually saw on the screen:

```
Feature: Home page

As a user,
I want to be able to log in
So that I can use the system

  Scenario: Get to the login screen
  # features/login.feature:7
    Given a user named "Reuven" "Lerner" with an e-mail
     ➥address "reuven@lerner.co.il" # features/login.feature:9
      Undefined step: "a user named "Reuven"" (Cucumber::Undefined)
      features/login.feature:9:in `Given a user named "Reuven" "Lerner"
      with an e-mail address "reuven@lerner.co.il"'
    When I go to the home page
    # features/step_definitions/webrat_steps.rb:15
    Then I should see "Web site"
    # features/step_definitions/webrat_steps.rb:123
   Then I should see "Login"
   # features/step_definitions/webrat_steps.rb:123

1 scenario (1 undefined)
4 steps (3 skipped, 1 undefined)
0m0.012s
You can implement step definitions for undefined steps
 with these snippets:

Given /^a user named "([^\"]*)" "([^\"]*)" with an
 ➥e-mail address "([^\"]*)"$/ do
|arg1, arg2, arg3|
  pending
end

rake aborted!
Command failed with status (1):
 [/System/Library/Frameworks/Ruby.framework/...]
```

In other words, Cucumber looked for a definition that would handle my step "Given a user", but did not find one. It stopped interpreting my scenario and threw an error. Cucumber then went further, reminding me that I needed to define this step and giving me an outline for it.

A step definition, as you can see from Cucumber's suggestion, is a regular expression attached to a Ruby block. The regular expression is matched against the Given (or When or Then) statement, with one item matched using parentheses (the standard way of matching in a regular expression), which is then handed to the block as an argument.

Now, let's take the simple step definition and stick it into features/step_definitions/authentication.rb. When rerunning `rake cucumber`, Cucumber no longer can complain that this step definition is not defined. Rather, it signals that because the step definition is pending, it cannot continue with the rest of the scenario. Let's define this Given step:

```
Given /^a user named "([^\"]*)" "([^\"]*)" with an
 ➥e-mail address "([^\"]*)"$/ do
|first_name, last_name, email|
    @person = Person.create(:first_name => first_name,
                            :last_name => last_name,
                            :password => 'password',
                            :email_address => email)
end
```

You might have noticed that this step definition changed from the original, expecting two quoted words rather than one, with the block taking two parameters rather than one. Let's change the scenario definition so that it contains the step:

```
Given a user named "Reuven" "Lerner" with an e-mail
 address "reuven@lerner.co.il"
```

Running this in Cucumber gives the following:

```
Scenario: Users who go to the home page are asked to log in
# features/login.feature:7

Given a user named "Reuven" "Lerner" with an e-mail
 ➥address "reuven@lerner.co.il"
 # features/step_definitions/authentication.rb:1
  When I go to the home page
  # features/step_definitions/webrat_steps.rb:15
  Then I should see "Web site"
  # features/step_definitions/webrat_steps.rb:123
  And I should see "You must first log in"
  # features/step_definitions/webrat_steps.rb:123

1 scenario (1 passed)
4 steps (4 passed)
0m0.473s
Loaded suite /usr/bin/rake
Started

Finished in 0.000167 seconds.

0 tests, 0 assertions, 0 failures, 0 errors
```

If you are wondering who defined the three final steps, look no further than the right-hand side of the output: Webrat, a browser simulator written in Ruby, understands a large number of browser-style step definitions, including "I go to" and "I should see", allowing you to test for the presence or absence of text in each situation. Cucumber provides a wide variety of Webrat step definitions, such that you can tell Cucumber to go to a page, to fill in a form or to use selection lists, check boxes and radio buttons.

This is basically what it means to work with Cucumber. You create a feature in a .feature file and write one or more scenarios in that .feature file, the lines of which are matched by regular expressions defined in the step_definitions directory. The fact that the .feature file is written in English, from the perspective of the user, means you can show it to nontechnical managers or clients. They even can help write scenarios, and if the scenarios aren't written perfectly for the purposes of Cucumber, they can understand that you are trying to test the application from a variety of perspectives.

It feels a bit strange (at least, it did in my experience) to write scenarios in Cucumber, because you're basically writing code, but in full English sentences. It also took me some time to internalize the fact that each English sentence is similar to a subroutine call, invoking a particular piece of code in the step_definitions directory. Over time, you presumably will create a large library of such step definitions, which you then mix and match within your Cucumber scenarios to test your system.

Here is a second scenario I wrote, in order to test logging in:

```
Scenario: Users can log in by entering their
 name and e-mail address
Given a user named "Reuven" "Lerner" with an e-mail
 ➥address "reuven@lerner.co.il"
When I go to the home page
 And I fill in "reuven@lerner.co.il" for "email_address"
 And I fill in "password" for "password"
 And I press "submit"
Then I should see "Welcome back to the site, Reuven!"
```

Once my two scenarios pass, I commit them to version control and keep them in my application, in the features directory.

If my new scenario doesn't pass, I go through the same iterative process as before—either writing step definitions or fixing bugs in the code to make sure the steps pass. But, Cucumber is a bit slow to execute, so it can be a pain to run through all the features and all the scenarios. So, you can run Cucumber manually, rather than via Rake:

```
cucumber features/login.feature
```

You even can indicate that you want to run only the feature starting on line 13 of the file in question:

```
cucumber features/login.feature:13
```

This can be a real time-saver when you have a lot of scenarios in a single file and you are trying to debug only one of them.

More about Cucumber

Cucumber is a well-thought-out system, with a large number of features and abilities that correspond closely with Web developers' needs. First, the step definitions in Cucumber can use either

Resources

For an excellent introduction to the "story" approach to BDD, including features and scenarios, see this blog posting by Dan North, one of the leading lights in the BDD community: **dannorth.net/whats-in-a-story**.

The home page for Cucumber is **cukes.info**. That page contains documentation, screencasts and pointers to other resources to get you started testing with Cucumber.

One particularly nice presentation about Cucumber is at: **www.slideshare.net/linoj/cucumber-how-i-slice-it-presentation-924254**.

The home page for RSpec is **rspec.info**, and it contains installation and configuration documentation, as well as pointers to other documents.

The Pragmatic Programmers recently released a book called *The RSpec Book*, written by RSpec maintainer David Chelimsky and many others actively involved in the RSpec community. If you are interested in using Cucumber (or RSpec), this book is an excellent starting point.

The home page for Shoulda is **thoughtbot.com/projects/shoulda**. The documentation there is a good starting point, but you probably will need to play with it a bit in order to get the hang of things.

RSpec (the default) or Shoulda, two BDD testing frameworks that have become quite popular in the Rails community.

As I mentioned previously, you can use Cucumber to test models and controllers, and not just for integration testing. My personal preference has been to use Cucumber in this way, however, because it provides a nice, user-side perspective on things and lets you test the site as a user would.

Well, it lets you test the site as a user would, but with one large caveat: Webrat is a great tool, but it doesn't support JavaScript. This means if your site has a great deal of AJAX and JavaScript, you will not be able to test it via Webrat. There are ways to get around this problem, both using Webrat (for example, by including textual links in your application) and by using external testing systems, such as Selenium or Celerity/Culerity. But I have yet to find a system that is easy to integrate, reliable and runs on servers as well as on my desktop.

The fact that Cucumber is slow is a bit of a drawback; it can take quite a while to run through all of the scenarios on a large application. One solution is to use Cucumber's tag feature, which allows you to give one or more tags to a scenario. Then you can run all the scenarios with that tag, across all the features.

If you want to avoid hitting the database each time you create or update (or find) an object, you can integrate a factory (for example, Factory Girl) with Cucumber. This can speed things up, as well as give you a great deal of flexibility in creating scenarios and testing your application.

Conclusion

Cucumber is an innovative approach to testing that has really grown on me and demonstrates the power of English-language, story-based testing to a degree I have yet to see elsewhere. If you are developing in Ruby, I strongly suggest you take a look at integrating Cucumber into your own work.■

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.

DAVE TAYLOR

Listening to Your Twitter Stream

Answer simple Twitter queries automatically.

Last month wrapped up with a problem so complex we had to delve into a different programming language to create a solution to the mathematics of calculating the distance between two lat/lon points on the globe. My head's still spinning. I long ago graduated computer science, so what the heck?

This month, I thought we should move back to something a bit more fun and perhaps a bit less complicated (well, maybe not, we'll see) and return to Twitter.

What I've been thinking about is how helpful it would be to have a bot that listened to my Twitter stream and answered simple queries directly without human intervention. Stores could have a bot respond to queries like "hours?" and "address?", and students could have their schedule preprogrammed, and the bot could answer queries like "class?" by indicating what class students were in at that moment.

In fact, there's a local startup here in Boulder, Colorado, that is moving down this path called Local Bunny (**localbunny.com**), but it's doing a real, fully thought-out solution. By comparison, I'm going to show you a bubblegum and bailing wire approach!

Listening to Your Twitter Stream

To track a Twitter stream from an individual, it's quite easy: a call to the right URL with curl does the trick:

```
curl http://twitter.com/status/user_timeline/davetaylor.xml
```

That'll give you my last dozen tweets or so, along with a lot of additional information, all in XML format.

What we want, however, are mentions of an account or pattern, which require you to supply login credentials. This call is a bit more complicated, but you still can accomplish it with curl:

```
curl -u "davetaylor:$pw" http://www.twitter.com/statuses/mentions.xml
```

Here, I've set pw to my account password (you don't really want to know my password, do you?). The output, however, is something else. For an individual tweet, there are 42 lines of information that come back (for a 140-character tweet).

It's too much to show you here, but try the command yourself and be astonished at the output.

To trim it down, let's use grep with a regular expression to extract the Twitter ID of the person who sent the Tweet that mentions @DaveTaylor, and the tweet itself:

```
<text>@DaveTaylor  Have them send the money in gold bullion.</text>

  <screen_name>LenBailey</screen_name>

<text>@DaveTaylor Escrow.com</text>

  <screen_name>Ed</screen_name>
```

You can see here that the first tweet is from @LenBailey, and the second from @Ed.

Turning this into coherent output is a tiny bit tricky, because we really want to merge line pairs into a single line that denotes message and ID. That's a job for awk:

```
awk '{if (NR % 2 == 1) { printf ("%s",$0) } else { print $0 }}'
```

Now, if we feed the curl output to this, we'll see:

```
<text>@DaveTaylor  Have them send the money in gold bullion.</text>
<screen_name>LenBailey</screen_name>

<text>@DaveTaylor Escrow.com</text>  <screen_name>Ed</screen_name>
```

Next step: let's get rid of the XML artifacts and reformat it to be a bit easier to parse. We also can axe @DaveTaylor, because we know it's to this account already (in the actual code, it's one invocation, but here it's easier to show it in two lines for legibility):

```
sed 's/@DaveTaylor //;s/<text>//;s/<\/text>//' |
sed 's/   <screen_name>/ == /;s/<\/screen_name>//'
```

```
www.xetrade.com ?  == kiasuchick
 Have them send the money in gold bullion.  == LenBailey
Escrow.com == Ed
```

That's more like it!

Parsing Twitter Messages

Let's start by doing something simple. If you "@" my Twitter account with the command date, it'll detect it, actually run the date command, and send out the results on my behalf.

To do this, we'll want to split the data stream into "tweet" and "tweeter", but we can do this in a tricky way by tweaking the earlier awk string to create name=value pairs:

```
awk '{if (NR % 2 == 1) { printf ("msg=\"%s\"; ",$0) }
 ➥else { print "id="$0 }}'
```

The result:

```
msg="escrow"; id=Stepan
msg="www.xetrade.com ?"; id=kiasuchick
msg=" Have them send the money in gold bullion.  "; id=LenBailey
msg="Escrow.com"; id=Ed
```

Nice. Now we can use the underutilized eval command in the growing script to set the variables msg and id to the two, and then check msg for known values. Now, if you're sharp, you'll realize tweets that include double quotes are a bit of a problem, but fortunately, the Twitter API is smart too. All single quotes pass through as is, but double quotes are rewritten as the HTML entity &quot;.

Let's pause for a second so I can show you what I've built so far:

```
$curl -u "davetaylor:$pw" $inurl | \
  grep -E '(<screen_name>|<text>)' | \
  sed 's/@DaveTaylor //;s/  <text>//;s/<\/text>//' | \
  sed 's/    <screen_name>//;s/<\/screen_name>//' | \
  awk '{if (NR % 2 == 1) { printf ("msg=\"%s\"; ",$0) }
   ➥else { print "id="$0 }}' >
$temp
```

That grabs the 20 most-recent tweets for the specified user and converts them into `msg="message"` and `id=userid` for each one. Fed to eval in a loop, we now have a very easy way to parse things:

```
while read buffer
do
  eval $buffer
  echo Twitter user @$id sent message $msg
done < $temp
```

Let's wrap up the column here for now, but next month, we'll take the next step and actually parse the Twitter "@" messages being sent to me, trying to find those that match the predefined queries we've set, act upon them and respond.

This is going to be a pretty cool project when we're done!■

Dave Taylor has been involved with UNIX since he first logged in to the on-line network in 1980. That means that, yes, he's coming up to the 30-year mark now. You can find him just about everywhere on-line, but start here: www.DaveTaylorOnline.com. In addition to all his other projects, Dave is a film critic for a number of local publications. You can read his reviews at www.DaveOnFilm.com.

Linux Security Challenges 2010

MICK BAUER

Security challenges and worries for 2010: we live in interesting times indeed!

In August 2005, I wrote a Paranoid Penguin column titled "The Future of Linux Security", in which I described what I thought were the biggest challenges of Linux security in 2005 and the most promising new technologies for addressing them.

In that 2005 column, I suggested that virtualization might become a more important tool for isolating vulnerable applications and solutions than Mandatory Access Controls (MACs), such as SELinux and AppArmor. I also predicted that anomaly detection would become much more important than signature-matching, as the underlying engine behind most antivirus (AV) and intrusion detection/prevention systems (IDS/IPS).

So far, neither of those predictions has come to pass. We're still stuck with predominately signature-based AV and IDS/IPS technologies that are largely incapable of detecting "zero-day" malware that's too new for anyone to have yet created a corresponding signature or against polymorphic malware that alters itself from generation to generation.

Virtualization overwhelmingly has been driven by hardware resource management and other operational and economic concerns rather than security. In fact, virtualization, as most commonly deployed nowadays, is arguably a bigger *source* of security issues than it is a security tool (for example, for isolating vulnerable applications or services from other parts of a given system).

Am I embarrassed about those predictions not panning out? Not as much as I am disappointed. I still believe that AV and IDS/IPS *need* to evolve past signature-matching, and I still think virtualization has the potential to be a bigger part of security solutions than it is of security problems.

This month, more than five years since my last such overview, I'm devoting a column to my thoughts on what constitute some of the biggest Linux and Internet security challenges for 2010 and to my ideas on how we might address those challenges. This is by no means a comprehensive survey (time and space didn't permit me even to touch on mobile computing or embedded Linux, for example), but I think you'll agree that the issues I do cover represent some of the most far-reaching security challenges that affect not only the Linux community in particular, but also the Internet community at large.

Assets and Attackers

Before I zero in on specific technical areas, a quick word about the things we're defending and the people who are attacking them is in order, because those items have changed significantly since I started writing Paranoid Penguin. In the old days, we were concerned primarily with preserving network and system integrity against intruders whom we assumed were most likely to be bored suburban teenagers or industrial spies.

Governments, of course, worried about other types of spies, but I'm talking about civilian and corporate space (and generalizing heavily at that). The point being, the classic attack scenario involved people trying to remote-root compromise some Internet-facing system so they could deface your Web site, steal proprietary information or use that system as a platform for launching attacks on other systems, possibly including systems "deeper inside" your internal corporate network.

We still worry about that scenario, of course. But over the past decade, there has been an explosion in identity theft across a wide spectrum: simple e-mail-address harvesting for the purpose of spamming; stealing, trafficking in or illegally generating credit-card numbers for making fraudulent purchases; full-blown assumption of other people's names, social-security numbers (or other non-US identifiers), bank account numbers and so forth, for the purpose of fraudulently opening new credit accounts; laundering money gained in other criminal activity, and so on.

Sometimes identity theft is achieved through the old-school, console-intensive attacks of yore, against databases housing dense concentrations of such data. Much more commonly nowadays, it involves sophisticated malware that either infiltrates a given bank or merchant and works its way to its databases or harvests data at the *client* level, possibly even by capturing individual user's keystrokes.

Because spam, fraud and identity theft in general are so lucrative (amounting to billions of dollars annually), it should be no surprise that organized crime is behind a lot if not most of it. I'm speaking not only of traditional crime organizations that also run prostitution, illegal drug and gambling operations, but also completely new organizations focused solely on credit-card trafficking (“carding”) and other electronic crimes.

College students and teenagers still fit into the equation, but in many cases, they’re working for scary people, for real money. The people writing the trojans, worms and viruses that do so much of the heavy lifting in these attacks are, in many cases, highly skilled programmers earning much more than the people who write anti-malware and firewall software!

This is our new security landscape. The situation is no more or less unwinnable than it was ten years ago, and sure enough, ecommerce and Internet traffic in general still are churning along more or less smoothly. But, we need to pay attention to these trends for that to continue to be the case.

So, how do these trends in the asset and attacker equation affect the defense equation?

Web Application Security and SSL/TLS Weaknesses

If Internet security is a war, Web applications surely constitute the front line. Nothing has changed the way we interact with computers, and the places from which we interact with them, like the World Wide Web—that is, the use of Web browsers to access data and even entire networks that are mediated by Web servers. In fact, the term World Wide Web is all but obsolete. The Web is now so ubiquitous, it’s become synonymous with the Internet and even to some extent with user interface.

Web browsers now do things that used to be done by entire operating systems. Whereas the primary function of Web browsers used to be to format and display data correctly (Web pages originally being, in real terms, brochures with text, images and links to other pages), for some time now, we’ve used our browsers to download and execute code transparently. This code can be as simple as a script that takes form data from us, such as an on-line order form, and transmits it back to the server. Or, it can be as complex as an entire remote-desktop application that lets us control a computer on the other side of the world.

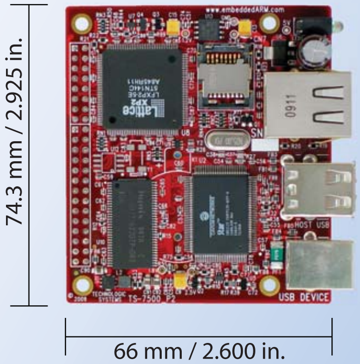
Most of the same things an attacker might attempt to subvert in an operating system, therefore, are now worth attempting against a Web browser. In the world of OS security, we worry about viruses—executables that end users might be tricked into running that change system behavior or steal data on the system. In the world of browser security, we worry about hostile Web sites—hostile Web content that can change browser behavior or steal data the browser is processing or has stored on the system.

And, that’s just on the client side of the Web application equation! On the server side, we worry not only about hostile Web sites, but also about flaws in our own Web applications that might allow attackers to gain unauthorized access to back-end systems and data, or to attack other users.

What about SSL/TLS? Doesn’t that provide a reliable means of cryptographically signing and verifying active content (code), authenticating transactions and preventing eavesdropping? Obviously, yes. It does so well enough for most of us to shop on-line, do on-line banking and so forth, with a reasonable level of safety and confidence. However, as I reported in my November 2009 DEFCON column, there has been a marked increase lately in man-in-the-middle attacks against SSL/TLS-protected transactions.

Some of these attacks exploit the ways commercially signed digital certificates are issued, maintained and verified by major issuers, such as VeriSign. That is, they exploit weaknesses in commercial public key infrastructures. Others exploit the ways Web servers and browsers handle SSL/TLS functions and the ways they alert (or don’t alert) end users of suspicious behavior.

The good news is the actual cryptosystems on which SSL/TLS is built remain sound. Most of these problems stem from the way Web server and browser developers implement them (less so Web application developers) and the way large Certificate Authorities manage certificates. On the one hand, the server/browser development and PKI communities have their work cut out for them in figuring out how to keep SSL/TLS mechanisms transparent enough for ordinary users to accept and



have success with, while fixing these new, serious security gaps. Even getting those communities to acknowledge their respective responsibilities and roles in fixing these issues is a big challenge, and it's not at all clear that they have done or will do so.

But, at least the suite of algorithms and other systems comprising TLS itself is sound. This is a solvable problem!

Cloud Computing

As Internet connectivity has gotten faster, cheaper and more ubiquitous, people have begun to question the need for relying on local computing power and storage, if so much of one's daily computing experience depends so heavily on Internet connectivity anyhow. If your major applications are all Internet applications, why not run them *from* over the Internet, on remote servers rather than your local CPU or on your local IT infrastructure?

Why not subscribe to a framework in which external providers host enormous farms of servers and storage arrays on which *anybody* can host virtual servers running massively multiuser on-line applications? Heck, why not just use applications written and maintained by the provider? Should end users even care where and how these applications are being run or who wrote them in the first place?

This is the promise of cloud computing—not just the major systems in your data center, but the data center itself—from the floor to which the racks are bolted upward to the applications running on top of it all—can become someone else's problem to maintain, for much cheaper than maintaining any of it yourself. All you need is a bunch of users with ordinary desktop systems (or Netbooks, for that matter) and Internet connectivity.

Maybe I've been a network engineer for too long, but I have a little trouble seeing the appeal of being *completely* dependent on network connectivity to do all my work. Even though I don't do much computing off-line nowadays, I certainly like to think I *could*. (I definitely would have written this article much faster without the distraction of an open browser window!)

My real problem with cloud computing, however, is the difficulty of protecting data that is not just flowing through, but being processed and stored by, applications owned and maintained by someone else on hardware and bandwidth completely outside my control. Frankly, I'm amazed that in an era when identity theft is the single-most quickly growing type of computer crime, any organization would be in such a big hurry to put such dense concentrations of its critical data in the hands of strangers.

Do I think cloud computing is a boondoggle? Not at all, but my attitude is the same as with IT outsourcing in general. It probably makes sense for certain applications used by certain organizations, but I think the whole concept is being grossly oversold, and I think people are overlooking substantial trade-offs and hidden costs, both operational- and security-related.

Malware

Malware has been with us a long time, and some of the things that scare us now, like polymorphic code that alters itself to thwart signature-based antivirus methods, actually have been around a while. What's changed recently is the emergence of "targeted malware": worms, trojans and viruses designed to attack specific parts of specific target organizations.

Targeted malware is probably the scariest new threat that we as security professionals and system/network administrators are faced with. By definition, it's always "zero-day". You never can hope your antivirus software provider has signatures for code that not only has never been released into the wild, but that also won't necessarily even *function* against anybody's network and systems but yours. Targeted malware almost is never written from scratch. In fact, it's frequently generated using sophisticated, slick "malware construction" software written by the aforementioned highly skilled, highly paid malware authors of the underworld.

But although you might think there's some potential for detecting common characteristics between hostile applications targeting different organizations but originating from the same development tools, these tools are in fact specifically designed to write code that evades detection. In fact, at a recent security conference, a forensics specialist whose presentation I attended commented that it's not uncommon for his team to fail to isolate fully the source of attacker activity on a compromised network beyond identifying infected systems. Much of the code he encounters nowadays is too deeply embedded into other applications, DLLs and even the kernel itself to be identified and isolated easily.

Equally scary is how it's propagated. You may think that firewalls, application proxies and other defenses on your network's perimeter should minimize the chance for worms to penetrate your internal systems in the first place. You may even be correct. But frequently, targeted malware is installed *directly* onto one or more internal systems at a target site by either a corrupted insider or a crook who's obtained a job at the target organization for the specific purpose of placing the malware.

It's already hard enough to ensure proper physical security, OS-level access controls and application-level authorization controls for systems that handle or store sensitive data. But to do so uniformly across all systems or local networks that merely *interact* with such systems, and may have been compromised by malware, is a much bigger problem.

Furthermore, even if the back end is well secured, what about targeted malware that harvests data from end users? Your customer service representatives who handle customer account information may be perfectly trustworthy, but what if their systems become infested with keystroke loggers that transmit customer information back to some criminal's servers, over an SSL-encrypted network stream that's nearly indistinguishable from ordinary Web surfing? It's easy to imagine scenarios in which data handled by your organization's end users might be harvested by bad guys, if they were able to achieve even a small foothold on even one system in your internal network.

Is the targeted malware threat unstoppable? To some extent, yes. In practical terms, it's a particular type of insider attack, and insider attacks can never be prevented completely. The good news is we already know how to manage insider threats: background checks, system/application/employee monitoring, granular access controls at all levels, good physical security and so forth. The more broadly and consistently we apply these varied, layered controls, the less likely it will be that even a given targeted attack can succeed, and the more limited the scope of damage it is likely to cause.

Like so much else in security, it's a game less about preventing attacks, than of increasing the cost and effort required for such an attack to succeed.

Virtualization

And, now we come to virtualization, which both on its own and in tandem with cloud computing is the focus of so much buzz and hype. Virtualization has unquestionably altered the way we think about computers. By making the notion of "computing hardware" almost completely abstract relative to operating systems and applications, virtualization can free us from certain types of physical and even geographical limitations, or more accurately, it can shift those limitations to a different part of the resource planning process.

Perhaps overly idealistically, I used to think virtualization could free us from the "winner take all" phenomenon in operating system security. On any system under attack, attackers frequently need to find only one vulnerability in one application to compromise the entire system completely. But what if the most vulnerable application on a given server is the *only* network listener on that system?

Suppose I need to run an SMTP relay using Sendmail, and I normally also would run a network time protocol (NTP) dæmon, the Secure Shell dæmon (sshd) and RealVNC on that same system. That's four different attack vectors on one system. But, what if I run Sendmail in its own virtual machine on that host, allowing access to it from the outside world, and for the four other dæmons running on the underlying host, accept connections only from the IP address of some internal access point?

Sure, I could achieve a similar thing without virtualization by using TCP Wrappers or a local iptables policy. But if all dæmons run on the same system, and attackers gain only a partial foothold via Sendmail, perhaps resulting in nonroot remote access, the attackers may be able to attack one or more of the three other dæmons to attempt to escalate their privileges to root. But, if those dæmons are running on the virtual Sendmail machine's host system, and configured to reject connection attempts from the Sendmail virtual machine, that second attack will fail.

Unless, that is, our assumptions about virtualization don't hold. This brings me to the dark underbelly of virtualization, which in our headlong rush to maximize hardware resource utilization, I fear may not be under close enough inspection.

We assume that one virtual machine can't see or gain access to the resources (disk space, memory and so on) used by other virtual machines running on the same host. Virtual machines are supposed to be isolated by, among other things, a hypervisor or monitor program. We also assume that it isn't feasible or possible for any userspace application running on a guest virtual machine to speak directly to any process or resource on the underlying host.

If you write hypervisor code, there are strong incentives for you to maintain these assumptions and write a secure hypervisor. Pretty much anything that can subvert hypervisor security will have a negative impact on system performance, availability and overall reliability. For example, a bug that allows one virtual machine to access another's memory, while potentially calamitous if discovered by an attacker, is at least as likely to result in one virtual machine's impairing another's performance by *unintentionally* overwriting its memory.

But recent history has shown that both theoretical and demonstrable attacks are possible against popular system virtualization environments, such as VMware (see the link to Michael Kemp's presentation, in Resources).

Does this mean we shouldn't use virtualization? Of course not. This is a powerful and useful technology. But it's also very new, at least in many contexts in which we're deploying it nowadays, and until hypervisor security is better understood and more mature, I *do* think we should be careful about which virtual machines we run on the same host. It seems prudent to me to colocate only systems handling similar data and representing similar levels of risk (for example, Internet-reachability) on the same host system.

In other words, we probably shouldn't rely on hypervisors to protect virtual machines from each other, more than we have to.

Conclusion

The explosive proliferation of new types of Web applications, cloud computing services and virtualization solutions are exposing our data, systems and networks in ever-bigger, ever-further-reaching ways. Targeted malware, man-in-the-middle attack techniques and similar threats against SSL/TLS, the involvement of organized identity theft rings, and other nasty trends on the attack side of the equation only make it harder for those of us concerned with security to protect these emerging applications, services and infrastructures.

But what is a crisis, if not a job for experts? Interesting times call for creative, technology-obsessed types like *Linux Journal*'s readers (and columnists), and I have no doubt that we, the geek community, are amply up to the challenge. So, here's wishing you a safe, productive and interesting (in the good sense, not just the scary sense) 2010!■

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

Michael Kemp's Presentation "Virtualization: There Is No Spoon" (from Bellua Cyber Security 2008 Conference): **www.bellua.com/conference/asia08.materials/bcs08-kemp.ppt**

The 2009 Verizon Business Data Breach Investigations Report (describes trends in the use of targeted malware and the involvement of organized crime in real-world security breaches): **www.verizonbusiness.com/resources/security/reports/2009_databreach_rp.pdf**

Dr hjkl Meets the Vimperator

KYLE RANKIN

If you want to be a futuristic Web-browsing machine, terminate that mouse, pick up the keyboard, and find out how Vimperator can transform Firefox into a modal half-vim, half-browser cyborg.

In November 2009, I wrote an entire column ("Dr hjkl and Mr Hack") devoted to programs with vi-style keybindings. In the column, I introduced the Vimperator plugin for Firefox and discussed how it worked, but at the time, I mentioned, "The Vimperator plugin is extensive enough to deserve a column of its own (in fact, e-mail me at lj@greenfly.net if you'd be interested in that.)" Well, I received a number of responses, so between that and my love for all things with vi keybindings, I think it's worth giving the Vimperator plugin the full column it deserves.

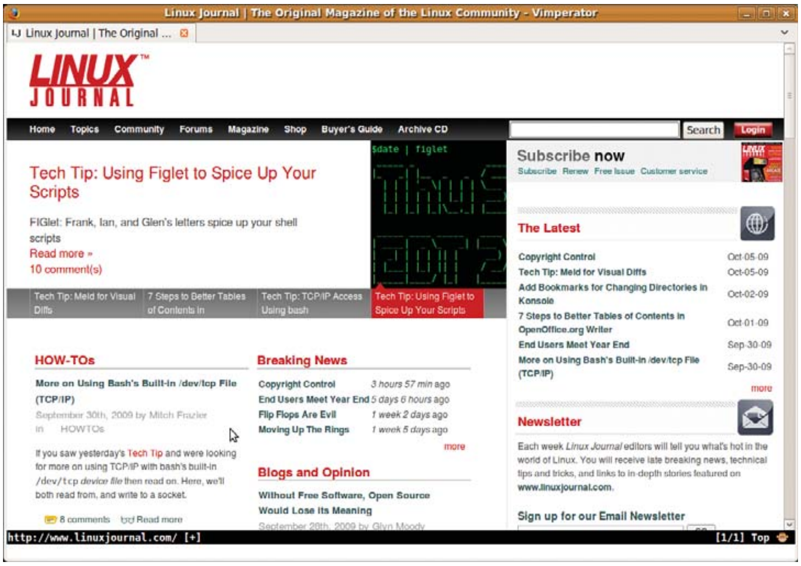
Hasta la Vista, Mousey

As I mentioned in my previous column, the main reason I love vi-style keybindings is that they keep your fingers on the home row (the asdfghjkl; row on your keyboard)—something quite important if you touch type. Once you get used to using hjkl to navigate documents, it and the rest of the keybindings become second nature. If you touch type with any decent speed, you realize how much it slows you down to reach for the mouse or even the arrow keys—particularly for something simple like clicking a link. Now, most Web browsers have some limited way to browse a page with a keyboard, but they almost always use the arrow and Page Up and Page Down keys, all of which are pretty far from the home row.

In the past, I've used a number of methods to add some level of vi-style keybindings to Firefox. At first, I used a custom configuration to my Firefox config, and later, I used the mozless extension. Both worked okay, at least for certain versions of Firefox, but they still were a limited version of the real thing. Well, the Vimperator plugin is the real deal. It goes far beyond simple keybindings and actually creates a modal interface with an incredible level of detail. You not only get hjkl navigation, but you also can open tabs and even record macros just like in vim. What's more, Vimperator was built with Web page navigation in mind, so there are keybindings available to make it easy to click on links and even hover over elements on the page—all from the keyboard.

Come with Me If You Want to Live

The first step is to install the Vimperator plugin. Visit **vimperator.org**, click on the Download Vimperator button on the page, and go through the typical Firefox plugin installation process. Once you start Firefox again, the first thing you will notice is that your menu bar is gone (Figure



Linux Journal | The Original Magazine of the Linux Community - Vimperator

LJ Linux Journal | The Original ...

LINUX™ JOURNAL

Home Topics Community Forums Magazine Shop Buyer's Guide Archive CD

Search Login

Tech Tip: Using Figlet to Spice Up Your Scripts

FIGlet: Frank, Ian, and Glen's letters spice up your shell scripts

Read more »

10 comment(s)

```
$date | figlet
```

Tech Tip: Meld for Visual Diffs
7 Steps to Better Tables of Contents in
Tech Tip: TCP/IP Access Using bash
Tech Tip: Using Figlet to Spice Up Your Scripts

Subscribe now

Subscribe Renew Free Issue Customer service

The Latest

Copyright Control	Oct-05-09
Tech Tip: Meld for Visual Diffs	Oct-05-09
Add Bookmarks for Changing Directories in Konsole	Oct-02-09
7 Steps to Better Tables of Contents in OpenOffice.org Writer	Oct-01-09
End Users Meet Year End	Sep-30-09
More on Using Bash's Built-in /dev/tcp File (TCP/IP)	Sep-30-09

more

HOW-TOs

More on Using Bash's Built-in /dev/tcp File (TCP/IP)

September 30th, 2009 by Mitch Frazier in HOWTOs

If you saw yesterday's Tech Tip and were looking for more on using TCP/IP with bash's built-in /dev/tcp *device file* then read on. Here, we'll both read from, and write to a socket.

8 comments Read more

Breaking News

Copyright Control	*3 hours 57 min* ago
End Users Meet Year End	*5 days 6 hours* ago
Flip Flops Are Evil	*1 week 2 days* ago
Moving Up The Rings	*1 week 5 days* ago

more

Blogs and Opinion

Without Free Software, Open Source Would Lose Its Meaning

September 28th, 2009 by Glyn Moody

Newsletter

Each week *Linux Journal* editors will tell you what's hot in the world of Linux. You will receive late breaking news, technical tips and tricks, and links to in-depth stories featured on www.linuxjournal.com.

Sign up for our Email Newsletter

```
http://www.linuxjournal.com/ [+]                                   [1/1] Top
```

Figure 1. Vimperator-Enabled Firefox without the Menu Bar

1)! Now, this might be fine once you get accustomed to Vimperator, but I found it a little jarring at first, so you might want to type `:set guioptions+=mT` to turn the menu bars back on for now. Notice that as with vim, you press the : key to enter command-line mode. Vimperator turns Firefox into a modal browser like vim that has a command-line mode (accessed when you press the : key) as well as a normal and insert mode. Also as with vim, when you get stuck in some strange mode, you generally can just press Esc a few times to get back to normal. If you find you want the menu bar back permanently, add the following to your ~/.vimperatorrc file:

```
set guioptions+=mT
```

This file acts like ~/.vimrc, so you can add any other Vimperator-specific settings here as well.

The basic navigation with Vimperator should be pretty familiar to you if you've ever used vim before, but in case you are still new to that kind of navigation, here's a quick list of keybindings:

- h — scroll left.
- j — scroll down one line.
- k — scroll up one line.
- l — scroll right.
- gg — move to the top of the page.
- G — move to the bottom of the page.
- / — enter search mode.
- n — move to the next match in your search.
- N — move to the previous match.
- Spacebar — move down one page.
- Shift-spacebar — move up one page.
- Esc — go back to standard navigation mode.
- F1 — show Vimperator help.

So for instance, if I wanted to use Vimperator to search for "Sarah Conner", I would press /, type in Sarah Conner and press Enter. Vimperator would jump to the first instance on the page. If the first Sarah Conner wasn't the right match, I would press n to move to the next match or N to go back to the previous match. If I wanted to start a new search from the top of the page, I could type gg to move back to the top, then / to enter search mode, and then type, for instance, "John Conner" and press Enter.

As with vim, you also can add numerical modifiers to any of these commands, so if you want to move down five lines instead of just one, you can press 5j. If you forget the keybinding for a particular function, just press F1 or type `:help` to see the full Vimperator help screen.

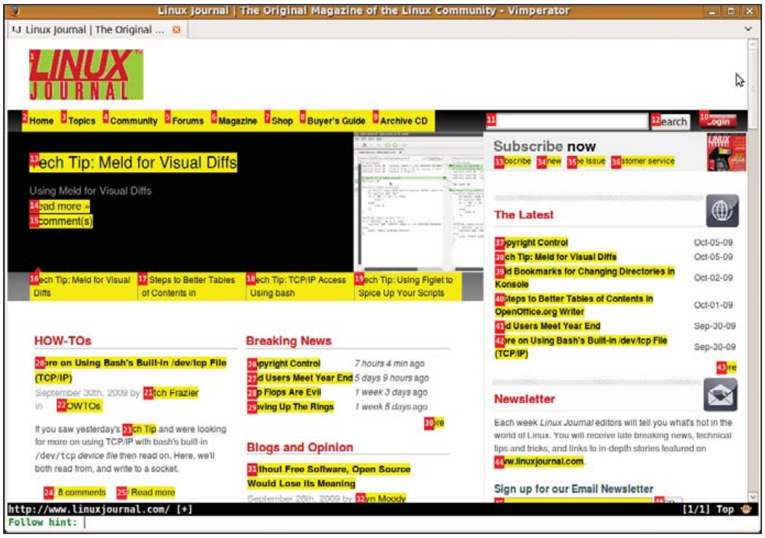
Vimperator would be useful even if it provided only the standard navigation keys, but it also adds a complete set of keys to access standard browsing functions. Here is a list of some of the standard ones:

- H — go back in the current tab's history.
- L — go forward in the current tab's history.
- gt — go to the next tab.
- gT — go to the previous tab.
- d — close the current tab.
- u — undo: open a previously closed tab (works with multiple previously closed tabs).
- r — reload the current page.
- R — reload the current page without the local cache.

Now, I've found that when I use tools like S5 for Web-based presentations, the keybindings it expects conflict with Vimperator. Luckily, Vimperator makes it easy to disable its keys temporarily. Simply press Ctrl-z, and all keybindings will go back to standard Firefox mode until you press Esc. I also use this mode when I browse Google Reader, because it already accepts vi-style key bindings to browse through RSS feeds. If you just need to enter one key that Vimperator won't intercept, you can press Ctrl-v, and after you press the key, Vimperator will go back to its normal mode.

Once you have the standard movement down, you might wonder, how do I actually open a new

Vimperator would be useful even if it provided only the standard navigation keys, but it also adds a complete set of keys to access standard browsing functions.



```
http://www.linuxjournal.com/ [+]                                   [1/1] Top
Follow hint:
```

Figure 2. LinuxJournal.com in Hint Mode

URL without a menu bar? Either press o (or type `:open`) followed by the URL you want to open to load that URL in your current tab, or press t (or type `:tabopen`) to type in a URL to open in a new tab. In addition to these basic keys, there also are a number of variations to them:

- T — open a :tabopen prompt, but fill in the URL with the URL of your current tab.
- O — create an :open prompt, but fill in the URL with the URL of your current tab.
- w — like :tabopen but only opens the URL in a new window.
- W — like T, it creates a :winopen prompt and fills out the URL with the URL in the current tab.
- p — open a URL based on the contents of the clipboard.

Once you type in a URL, you also can press the Tab key to trigger Tab-complete based on your browser history. Speaking of browser history, you still can access that and the other standard Firefox functions from command-line mode:

- `:bmarks` — access all of your Firefox bookmarks in command-line mode.
- `:history` — view your browser history.
- `:emenu` — access functions in the standard Firefox menu.
- `:dialog` — access other Firefox dialog windows; type `:help :dialog` for more information.

Heads-Up Link Displays

In my mind, the real power of Vimperator besides the standard keybindings is the fact that you can use the keyboard to open links, move to input boxes and even simulate mouse hovering. Vimperator calls this Hint mode, and to activate it, press the f key on any Web page. All of the “hintable” objects on the page, such as hyperlinks, text-entry boxes and drop-down menus will be highlighted with a number assigned to them (Figure 2). To select one of the highlighted items, you either can type in the number next to it and press Enter, or you can start typing part of the highlighted text. For instance, if you are reading a multipage article on the Web and see links to each page of the article along with a Next link, you could press f and then type `Next`. As you type, hints that no longer match drop away, and once there is only one match left, it automatically will load. When you use f, hints will open up in the current tab, but if you want to open the page in a new tab, simply start Hint mode with F instead of f. Like with other modes, you can press the Esc key to exit Hint mode.

The f and F keys activate a Quick Hint mode, but you also can activate an Extended Hint mode to enable other actions on a link beyond a left-mouse click. To enable Extended Hint mode, press the ; key, followed by a special key to set the type of action you want to perform, and finally type the number associated with a particular hint. Here is an abridged list of some hint modes you might want to use, but for the full list, check the Vimperator help page. Keep in mind that you will press the ; key before any of these keys:

- ; — pressing two ; keys in a row will focus a link and hover over it with a mouse; this is useful for activating JavaScript drop-down menus.
- s — save the destination of a link.
- f — focus a particular frame.
- y — yank the destination location for a link.
- Y — yank the text description of a link.

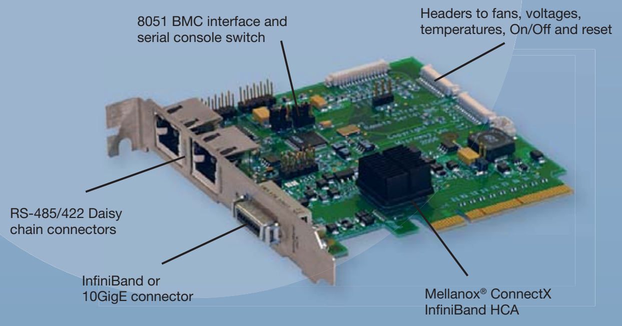
Believe me, I’ve barely scratched the surface of Vimperator here. It really reminds me of vim in the sense that I always feel like I’m using only 10% of the available features. As with vim though, Vimperator rewards you while you progress through its learning curve. I use Vimperator on all of my Firefox sessions, and it seems weird (and slow) to me now to browse Web sites with a mouse.■

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

Your Applications Will Run Faster With Next Generation Microway Solutions!

TriCom™ X

- QDR/DDR InfiniBand HCA
- ConnectX™ Technology
- 1µsec Latency
- Switchless Serial Console
- NodeWatch™ Remote Management


8051 BMC interface and serial console switch
Headers to fans, voltages, temperatures, On/Off and reset
RS-485/422 Daisy chain connectors
InfiniBand or 10GigE connector
Mellanox® ConnectX InfiniBand HCA

Teraflop GPU Computing

For Workstations and HPC Clusters

- NVIDIA® Tesla™ GPU with 240 Cores on One Chip
  - CUDA™ SDK
- NVIDIA® Quadro® Professional Graphics
- AMD® FireStream™ GPU
  - Stream SDK with Brook+

NumberSmasher®

Large Memory Scalable SMP Server

- Scales to 1 TB of Virtual Shared Memory
- Up to 128 CPU Cores
- 8U System Includes 32 Quad Core CPUs
- QDR 1 µsec Backplane

FasTree™ X

- Mellanox® InfiniScale™ IV Technology
- QDR/DDR InfiniBand Switches
- Modular Design
- 4 GB/sec Bandwidth per Port
- QSFP Interconnects
- InfiniScope™ Real Time Diagnostics

Call the HPC Experts at Microway to Design Your Next High-Reliability Linux Cluster or InfiniBand Fabric.

_508-746-7341_

Sign up for Microway's Newsletter at
www.microway.com

Microway®
Technology you can count on℠

DIRK ELMENDORF

Who Goes There? Adventures in Amateur Security

“Firewalls are a hardware solution to a software problem.”
—Someone at ShmooCon

The situation: I share an office with my brother. This office is in a suite of other offices that we share with another company. Sometimes I work from home; sometimes I go in. I haven’t been in the office for the past few weeks because of a motorcycle accident (from which I hope to be fully recovered by the time you read this—fingers crossed). That got me thinking, I have no idea if anyone has been in my office during the time I’ve been out. It would be great if I had some sort of basic security system that could tell me if people entered my office and, if possible, show me a picture of who they were.

The Front Door

The first thing I did when I moved in was upgrade the lock to my office. This actually was more about laziness than security. I got a keypad lock so I still could get in even if I forgot my keys (or needed someone else to get something if I wasn’t there). When I went shopping, there were a lot of locks from which to choose. The one that really caught my eye was the Schlage Wireless Keypad lock. This was a keypad lock, but it also included Z-Wave support. Schlage had an add-on version and a starter kit. The starter kit included a wireless hub and a lamp controller. It didn’t seem like a terrible deal until I realized the included hub required a monthly service fee to use it, which turned me off.

It would be great if I had some sort of basic security system that could tell me if people entered my office and, if possible, show me a picture of who they were.

I figured there had to be a way to control it from Linux. If I could do that, I could meet my first goal—alert me when someone enters. I’ve used X10 for a long time, and it always was pretty easy to script from Linux. I assumed Z-Wave would be the same, but it turns out, Z-Wave communication is a lot more complicated (I guess I shouldn’t be surprised, as it is much more advanced than X10).

Under Linux, there are two paths to follow to Z-Wave access. The first is the LinuxMCE (**www.linuxmce.com**). This project is a combination of media management, home automation, phone and security system, all built on top of Kubuntu. In order to access Z-Wave from it, you need a Z-Wave hardware dongle. Several supported dongles are listed on the Web site. I was especially interested in this path, because of rhouse from Fernand Galiana (**github.com/derailed/rhouse**). rhouse lets you script LinuxMCE from Ruby. It seemed like a solution right up my alley.

The second path for Z-Wave is the Vera from Mi Casa Verde (**www.micasaverde.com**). I first learned about this product in the May 2009 issue of *LJ* (**www.linuxjournal.com/article/10302**). As I already mentioned, I’ve used (struggled with) X10, and the Vera seemed like a solution that would be more reliable and more accessible to my other family members.

Choosing a path always is hard. The LinuxMCE path was tempting because it used the least amount of hardware and the most amount of software (which favors my skills). But, I ended up choosing the Vera, because of deployment issues. I don’t run Kubuntu, so I either would have to set up another box in my office or run a virtual machine on my workstation (and handle passing through the USB Z-Wave dongle). The Vera box uses little energy and, thus, does not generate much heat—something that is always a consideration here in south Texas. I decided to start with the Vera and figured I always could fall back on the LinuxMCE.

Setup

I turned on and set up the Vera. A handy guide on the Wiki (**wiki.micasaverde.com/index.php/Schlage_Lock**) walked me through the process of pairing the lock. The first step was upgrading my firmware. My Vera was running 1.0.434, but 1.0.602 was required for the lock, and the current version was 1.0.939. I went to the latest version, figuring I might get other improvements as a bonus.

If you have ever worked with X10, the pairing process for Z-Wave is a breath of fresh air. Instead of messing with small toggles, you simply pull the dongle out and press a button to put it in pairing mode. Then, you activate the Z-Wave device. Next, you put the dongle back into the Vera and start configuring your new device. This would all be true if you were talking to a normal Z-Wave device, like a lamp. It turns out, they take lock management a lot more seriously. As a result, the pairing process is much more involved. And, it's a little more complicated, because the Schlage does not emit a very strong signal (a result of it trying to conserve battery power and being encased inside the lock mechanism). That didn't prove to be too much of a problem, because I had a network and power port not very far from the door.

I did the secure pairing dance, which mostly meant starting the pairing process and then running to the door to punch in a programming code. After two attempts, I got a green light from the lock. I was able to see the lock on the Vera device screen. Things always should go this smoothly. I started adding and removing code. I clicked on lock and unlock. Nothing seemed to happen. It turns out that although the lock and Vera were talking, they still were exchanging encryption information—meaning that all my messing around was being queued up until it was done. After about ten minutes, the lock and Vera were fully in sync.

Here is where I realized the LinuxMCE path is not as clear as I thought. I did some research, and I was not able to find anyone who reported pairing with a lock under LinuxMCE successfully. I would have been really frustrated to have gotten Kubuntu, LinuxMCE and rhouse installed only to find out the lock was unsupported.

Once the job queue was cleared, I was able to add myself as a user to Vera. My profile on the Vera has an e-mail and cell-phone number so I can get SMS alerts. Vera supports the concept of "scenes". A scene allows you to tie together an event, a notification and one or more changes to Z-Wave devices. I created a scene called Door Opened, which was tied to the event that is generated when the lock reports being opened. I also told the scene to notify me via both e-mail and SMS.

Once this was set up, I was notified when the door opened. This was completely independent of entering a pin on the keypad, which meant that when I left the door unlocked, going in and out of the room generated alerts. The SMS can take some time, so by the time I realized how many alerts I had generated, I was getting a lot of them.

Who Went There

Knowing someone is in my office is a great improvement. In the event of an actual compromise, it would be better to have some information about who the intruder was. That brings me to the second part of the solution, the Webcam. I had one lying around and figured this would be a good use for it.

The Vera appears to be a rebadged ASSU WL-500gP. It is running OpenWRT with custom modifications to it. This turned out to be a good thing, because it means software compiled for OpenWRT works on the Vera. Mi Casa Verde does not officially support changes at this level, but it gives you full root access, so at least it doesn't get in the way.

The first step to doing this sort of thing is hooking up the Webcam to my normal workstation. I've learned from experience that if you cannot get it to work easily on a full Linux install, you have no shot at getting it to work on an embedded device. The Webcam plugged in, and I installed luvcview, a simple viewer program that lets you see what the Webcam sees. I ran luvcview and immediately was looking at a small picture of myself. This was awesome on two fronts. I'm pretty sure this cam didn't work under Linux the last time I tried it, and now I can move on to the hard stuff.

Getting shell access on the Vera is really easy. Go to Advanced→Net & wifi→Advanced configuration. It will ask you to set a root password. From that point on, you will be able to ssh in as root. The filesystem is a little confusing at first. Using df, the root filesystem appears to be completely full. The way the system is created, that is not actually true. In most cases, you can ignore that and simply untar things on to the root filesystem with no problems.

OpenWRT normally uses ipkg to manage packages, and that is broken on the Vera. The workaround is very straightforward though. You simply follow the same process for all packages. It turns out that the ipk package is just a set of nested tarballs. Here is the process for installing the gphoto2 package:

```
cd /tmp
wget
http://downloads.x-wrt.org/xwrt/kamikaze/snapshots/
 ➥brcm-2.4/packages/gphoto2_2.4.7-1_brcm-2.4.ipk

tar -xzvf  gphoto2_2.4.7-1_brcm-2.4.ipk
cd /
tar -xzvf /tmp/data.tar.gz
```

I was really excited, because there are two options for doing the image capture: gphoto2 and motion. gphoto2 is a command-line tool for controlling a normal digital camera. motion is a tool for controlling a Webcam and detecting motion.

What I really, really wanted was motion, which would provide an actual video of the person entering, but I ran into a classic version problem. The Vera/ASUS uses a Broadcom chipset for the onboard wireless. This is apparently flaky under the 2.6 kernel, so it is using a 2.4 kernel. The Webcam drivers for 2.4 are really limited. It turns out the uvc driver that allowed the Webcam to work on my workstation is available only in 2.6. I couldn't find a Webcam around the house that was supported with the drivers at my disposal.

So, that sent me on the hunt to get gphoto2 to work. It requires an actual digital camera. I had three different cameras from which to choose: a Canon SD1100, a Canon SD780 and a Canon EOS 400. You probably are noticing a theme here—they are all Canons. I love the little Powershot cameras. The last one is a DSLR that actually belongs to my wife (she's more serious about photography).

Here is where I learned that Canon has used its own protocol in the past, but apparently it's coming around. As a result, you need a recent version of gphoto2 in order to access the above cameras. In this case, I was incredibly lucky. It turns out the latest, greatest version was available and compiled already for me (**downloads.x-wrt.org/xwrt/kamikaze/snapshots/brcm-2.4/packages**).

> I'm not sure whether that was the root cause of my problem, but as a software developer, I feel it is my responsibility to blame the hardware first.

In order to make this work, you need three different packages: gphoto2, libgphoto2 and libgphoto2-drivers. The instructions above work for installing gphoto2. The libs take some extra steps:

```
cd /tmp
wget
http://downloads.x-wrt.org/xwrt/kamikaze/snapshots/
➥brcm-2.4/packages/libgphoto2-drivers_2.4.7-1_brcm-2.4.ipk
tar -xzvf libghoto2-drivers_2.4.7-1_brcm-2.4.ipk
tar -xzvf data.tar.gz
wget
http://downloads.x-wrt.org/xwrt/kamikaze/snapshots/
➥brcm-2.4/packages/libgphoto2-_2.4.7-1_brcm-2.4.ipk
tar -xzvf libghoto2_2.4.7-1_brcm-2.4.ipk
tar -xzvf data.tar.gz
```

This gives you the /tmp/usr/lib directory. Then:

```
cd /tmp/usr/lib/libgphoto2/2.4.7/
rm -f any_drivers_for_cameras_you_dont_have
```

In my case, I left canon.so, directory.so and ptp2.so. The last two are needed to talk to my camera. If you don't clear out the drivers directory, you will run out of space when you try to copy this on to the Flash portion of the Vera:

```
cd /usr/lib
cp -R /tmp/usr/lib/* .
```

Now you can hook up your camera. Typing `gphoto2 -a` should list all your camera's abilities. The most important ability is capture. The Powershots reported being able to capture images, but I was unsuccessful in actually getting them to do so. They require a special command to open the lens that did not work. I hooked up the EOS and got an IO error. After some research, I found I needed to format the memory card. Once that was done, I could trigger the camera from the command line. Thanks to an idea from wearetherock (**snipplr.com/view/19935/post-twitpic-with-curl**), I found out that posting a tweet with the picture was super easy. This solved two problems at once. The first is that the Flash memory in the Vera can not keep very many pictures around, and second, the system is more secure if I can store the picture off-site, safe from intruders.

The script is dead simple:

```
#!/bin/sh
cd /tmp
gphoto2 --capture-image-and-download --filename=now.jpg
 ➥--force-overwrite
curl -F "username=USERNAME" -F "password=PASSWORD"
 ➥-F "message=Intruder Cam" -F
media=@//tmp/now.jpg http://twitpic.com/api/uploadAndPost
```

(Replace USERNAME/PASSWORD with valid credentials.)

Now that I had the picture capture happening, I just needed to connect it to the open door event. I found some people were monitoring a logfile (which in recent firmware has changed to /tmp/log/cmh/LuaUPnP.log). That would be fine if I wanted a record of what happened. Instead, I want the camera to trigger on an event.

It turns out Mi Casa Verde has a solution for this. The latest firmware adds in Luup. This is a Lua-based interface to the system. It allows you to do some pretty advanced scripting. In my case, I only need to do some simple scripting.

I put my shell script in /root/upload.sh. The scene I already had created had a button for Luup scene. I don't actually know any Lua, but in this case, Lua expertise wasn't really required. I simply entered:

```
os.execute("/root/upload.sh >/dev/null 2>&1")
```

This told Vera to run my little shell script whenever that scene was triggered. So when the door was opened, the camera would take a picture and upload it to Twitter.

Now when I open the door, it takes a picture. If I spend some more time on this, I could add more logic so that the notification and picture happen only at certain times of day or night. This would be useful to cut down on the notifications I was getting every time I left the office to do something else like eat lunch.

Wrapping Up

I actually was really surprised at how far I was able to get. Barring some hardware issues (stupid dongle), I was able to wire up a system that monitored my door and notified me (and the entire world via Twitter). Long term though, I ran into some problems with my solution.

In the end, I was not able to convince my wife to sacrifice her camera permanently to improve the quality of my security system. I will have to find a cheaper digital camera.

I also ran into a number of problems getting my event to trigger properly. I thought it was the result of poor programming on my part, but I discovered I had a defective dongle (**forum.micasaverde.com/index.php?topic=1855.0**). I'm not sure whether that was the root cause of my problem, but as a software developer, I feel it is my responsibility to blame the hardware first. I still was waiting on a replacement when I wrote this article (more because I just submitted the request five minutes ago), but I will assume that the new dongle has showed up and everything works like a charm by the time you read this. If you get this far, and it does not work, contact me, and I'll tell you what I had to do to fix it.

There is one last thing to discuss—security. When I started this project, the goal was to increase the physical security of my office. Assuming everything I did always works, I now will receive a notification when someone enters my office (through the door). Assuming that I find a replacement camera, I also will have a picture that I can quickly check to see whether the person should be there. On the surface, it seems like things are more secure than before.

As I mentioned previously, I have a door with a keypad lock. To get in, you have to get a PIN or break down the door. Thanks to my new security system, there now is a third option. If you compromise the Vera, you can add your own PIN to the door. Then, you can enter the office with a minimum of fuss. There are a number of guides on making the Vera more secure, so maybe it is not actually a big deal. I guess this is the reason security is so difficult. I thought I was making things better by improving the monitoring of the room, but my implementation may have weakened my security on other fronts. For me, the trade-off is an acceptable one, but I'm mentioning it so you can make your own decision. I suppose that's why I'm so terrible at security—I am too happy to make the trade-off.■

Dirk Elmendorf is cofounder of Rackspace, some-time home-brewer, longtime Linux advocate and even longer-time programmer.

Black Duck Software's Black Duck Export

The raison d'être for Black Duck Software's new Black Duck Export 5.0 is to confront the issue of often undetected encryption algorithms found in the open-source software that developers integrate into their development process. A component of the Black Duck Suite, Export helps companies comply with export regulations by scanning software and identifying the presence of encryption algorithms that might affect a product's legal compliance in various markets. The new version 5.0 covers more than 450 different algorithms. Black Duck also offers a new companion publication, "The Guide to Encryption Export Compliance in an Open Source World", which can be downloaded at no cost from the company's Web site.

www.blackducksoftware.com

Instantiations' WindowTester Pro and WindowBuilder Pro

Instantiations concurrently released updates to two popular Eclipse-based GUI development tools, namely WindowTester Pro and WindowBuilder Pro, advancing to versions 4.0 and 7.2, respectively. The new features in WindowTester Pro v4.0 are designed to help developers easily automate recording, test generation, code coverage and playback of GUI interactions. Version 4.0 also provides improvements for test recording; fine-tuning of assertions functionality, including improved Swing assertions support; and improved Linux support and preliminary support for 64-bit Cocoa. WindowBuilder Pro's new features include code generation and parsing improvements to optimize performance, API support for customizing properties, enhanced JFace and RCP support, improved support for Riena and expanded data binding support to make it easier to tie a user interface to an underlying datastore.

www.instantiations.com

Panasas' ActiveStor

In its new ActiveStor Series 9 parallel storage system, Panasas claims to have "the highest-performance file storage system in the world", a system that combines solid-state drive (SSD) technology with traditional disk drives "to produce a system with breakthrough performance and consolidation capabilities". Panasas further says that its synchronized hybrid architecture produces both high-bandwidth performance and optimized IOPS and is capable of delivering an estimated 80,000 NFS operations per second, as well as 6 gigabytes per second of throughput. ActiveStor users can save money by consolidating a wider variety of applications and workloads in a single storage architecture, including high-performance clustered applications, single-client applications and technical and commercial applications running NFS and CIFS file protocols.

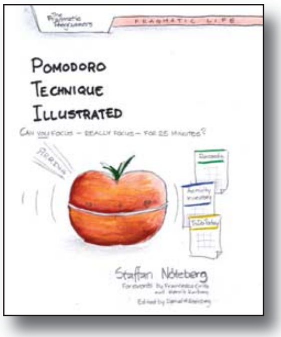
www.panasas.com

MontaVista Carrier Grade Edition Linux

Version 5.1 of MontaVista Carrier Grade Edition (CGE) Linux is now available, adding support for next-generation LTE and WiMAX 4G wireless networks. The product is a commercial-grade Linux development platform for network equipment developers working with off-the-shelf or custom hardware who require extensive support and want to build on open source. Other additions to CGE 5.1 include full integration of the OpenSAF high-availability middleware consistent with Service Availability Forum specifications and virtual routing and forwarding (VRF) capabilities for secure wireless networks. MontaVista also claims that CGE 5.1 remains the only carrier grade Linux to be compliant with Carrier Grade Linux, IPv6 and Linux Standard Base certifications.

www.mvista.com

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



Staffan Nöteberg's *Pomodoro Technique Illustrated* (Pragmatic Bookshelf)

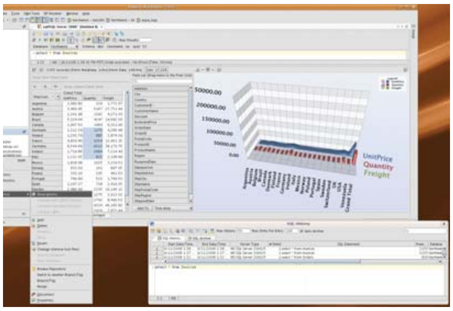
"Can you focus—really focus—for 25 minutes?" queries author Staffan Nöteberg as the subtitle of his new book, *Pomodoro Technique Illustrated* from the Pragmatic Bookshelf. We all know the pressure of needing to be productive—the to-do list is a mile long and you find yourself getting interrupted every other minute. You'd like to tell everyone to leave you alone, but most of the interruptions are coming from you! You think of a phone call you need to make or a Web site you need to check, and before you know it, you're answering e-mail, checking Twitter and finding a million other things to occupy your time. Author Nöteberg says that the Pomodoro Technique can put you back in charge of your day. You'll apply successful techniques from software engineering to identify what you should be doing today and to help you achieve your goals. Your mind won't wander when it is fully engaged in short bursts of focused activity. You'll learn to work less and accomplish more using nothing more than paper, pencil and a simple kitchen timer. This book is filled with advice on how to get started and how to tailor the method to your own needs.

www.pragprog.com

HPC Systems' HiPerDisk Storage System

The latest offering from HPC Systems is the HiPerDisk Storage System, the company's highest density SAS/SATA JBOD storage solution for enterprise applications. The HiPerDisk features 42 hot-swap drives in a 4U rackmount enclosure with up to 84TB of storage that will support high-capacity SAS or SATA drives individually or simultaneously, a feature that enables enterprise customers to standardize on one system for all three application data types: transactional, high-bandwidth and reference data. Designed for no active single point of failure, the HiPerDisk features single- or dual-controller modules, each consisting of three mini-SAS (4x) ports. A hot-swap redundant power supply and fans are provided for optimal enterprise reliability.

www.hpcsystems.com



AquaFold's Aqua Data Studio

Giving some overdue love to the database sector, AquaFold recently announced version 8.0 of Aqua Data Studio, a universal database administration software that simplifies the visualization and manipulation of multiple relational databases from within a single interface. The new v8.0 of Aqua Data Studio extends native support for the leading data warehousing and high-performance relational databases, including Teradata, DB2 for z/OS, Sybase IQ 15 and nCluster from Aster Data Systems. Thus, users will benefit from the features and functionality they are accustomed to when working with Aqua Data Studio's existing support tools for Oracle, DB2 for LUW, DB2 for iSeries, Microsoft SQL Server, MySQL, Sybase, Informix, Apache Derby and PostgreSQL. Another key feature is the extension of the Entity-Relationship Modeler, now with the ability to convert or translate an ER model from one database to another. Finally, users will enjoy the ability to export to ERX format, import object schemas into existing databases and compare schema objects of different models within the same or different databases.

www.aquafold.com

Ian Lawrence and Rodrigo Lopes' *Professional Ubuntu Mobile Development* (Wrox)

Publisher Wrox says that Ian Lawrence and Rodrigo Lopes' new book *Professional Ubuntu Mobile Development* is the world's first on the topic. The platform is quickly being adopted to provide a wide range of services on mobile devices, from Web to e-mail to messaging to GPS and more. This book is intended to show how to implement solutions for original equipment manufacturers and independent software vendors wishing to go to market using Ubuntu Mobile. The material also covers setting up Linux for mobile application development, the difference between developing on a mobile Internet device and a regular desktop environment, step-by-step tutorials and more good stuff.

www.wrox.com

Fresh from the Labs

Danger from the Deep

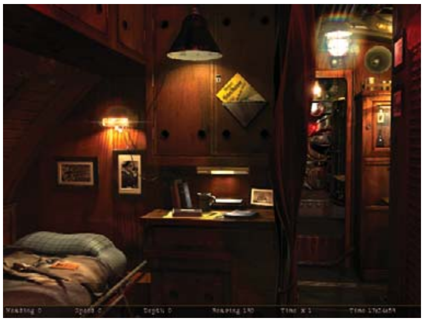
dangerdeep.sourceforge.net

If you remember last month's column, I was excited about a particularly cool-looking submarine simulator, *Danger from the Deep*. This month, I'm proud to feature it as one of the main projects. According to its Web site:

> *Danger from the Deep* (known as *dangerdeep* or *DftD*) is a free (as in free speech), open-source World War II German submarine simulator....This game is planned as tactical simulation and will be as realistic as our time and knowledge of physics allows. Its current state is alpha, but it is playable.
>
> *DftD* currently is being developed on Linux (i386 and AMD64) and Windows. There are binaries available for Linux (i386 and AMD64) and Windows (32-bit), and there are some old packages for Mac OS X. *Danger from the Deep* makes use of SDL/OpenGL and, thus, should be portable to other operating systems or platforms.
>
> *DftD* has even been reported to work on Windows (2000/XP/98), Linux (i386/x86-64/SPARC64) and FreeBSD (x86-64/SPARC64/IA64).

Installation In terms of hardware requirements, you'll need the following:

- An OpenGL 1.5-compliant graphics card (OpenGL 2.0 or greater is recommended).
- A fairly fast CPU (anything from 1.0GHz up should be okay).
- 256MB RAM (512MB is recommended).



Authenticity and romance are almost taken to extremes here, and touches like these captain's quarters really help you get into the feeling and atmosphere of the game.

The Web site provides a lovely binary installer that feels much like that of a commercial game. You can compile the game from source if you want, but would you really do that when you can simply click Next, Next, Next?

The binary installer runs most things by itself, but as is the usual way with these things, you need to flag it as executable. If you're running this through a file manager, you probably need to right-click and choose executable in whatever properties section is available, but I'll leave this part up to you. For those who prefer to run things from the command line, you can flag it as executable with the following:

```
$ chmod u+x dangerdeep-0.3.0-linux-installer.bin
```

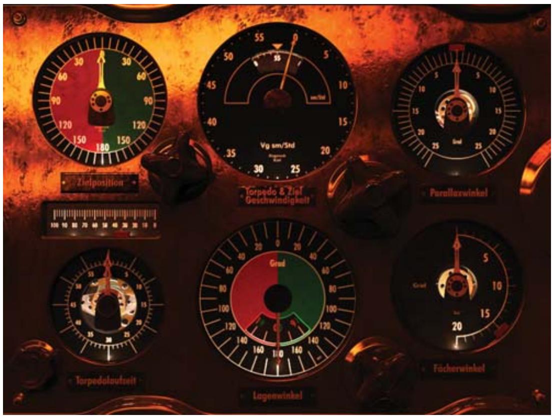
Then, run the installer with:

```
$ ./dangerdeep-0.3.0-linux-installer.bin
```

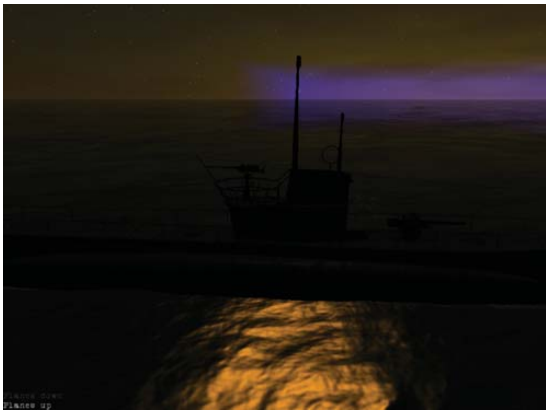
(Note: do this as root or sudo if you're installing it system-wide.)

The installer will go through the usual Next, Next, Next...Finish process, and once it's done, you can run the game with:

```
$ dangerdeep
```



Ah, authenticity! Don't speak German? Too bad! The controls stay that way, whether you like it or not (which I do).



Sometimes the environment can be quite beautiful, such as this sunset, which makes long hours in a sub much easier to bear!

Usage As soon as you're at the main screen, you'll be flooded by a wave of romantic nostalgia that never lets up! From the sepia-colored menus, German-labeled controls and wartime background imagery to the crackly WWII-era music, this game wears its heart on its sleeve.

Assuming you have a decent graphics card, you'll want to crank up the video resolution in the Options menu, as the graphics in this game actually are quite impressive at times.

Because this game is currently in alpha stage, the number of gaming options are limited. Nevertheless, several kinds of missions are available in single-player mode. For those wanting something to blast at from the get-go, try Historical Mission first under the menu Play single mission.

Before you embark on any voyages, however, I suggest that you print out the game's PDF manual, which is downloadable at the Web site. This game's controls are extensive (hey, you're piloting a submarine), and each key often is assigned to something different between other view modes. Here are a few basic universal controls to get you started though.

F1 through F12 all have a different view or function on the ship. Some keys bring you to actual locations within the submarine, such as the torpedo room or captain's quarters, and others bring up equipment, such as the ship's sonar. Of particular joy and amusement is the control screen on F1 and the torpedo screen on F6. The gauges and dials are in German, and as far as I know, there's no option to have them in English, so you'll have to find out what *HalbeFarht* and *Lagenwinkel* mean on your own.

Now, I'm sure at this stage you'll be keen to blast at something, so let's cover that. You may want to try surfacing for this, as it makes the above-water enemies easier to spot using more view modes, but it also shows off some of the cool graphics this game has.

Press the S key to surface, and when you're up there, have a look around with various different view modes (check out that deck for starters, and, yes, you can man that surface gun). If a target is somewhere in sight, press I to identify the target. Pressing the spacebar selects it, and T fires a torpedo.

I could go on and on, but it's been a privilege to see a project with such polish and passion as this one. In the hands of your average open-source team, a submarine simulator like this could be a bland, grey, blobby mess, and spending several hours with it might be a guaranteed cure for insomnia. This, on the other hand, is a colorful world of fine touches and authenticity that's guaranteed to bring a smile to your face.

Considering this game is in the alpha stage, it's an extraordinary effort with nice graphics, a brilliant installer and even a small soundtrack! Submarine buffs, do yourself a favor, and check out this game.

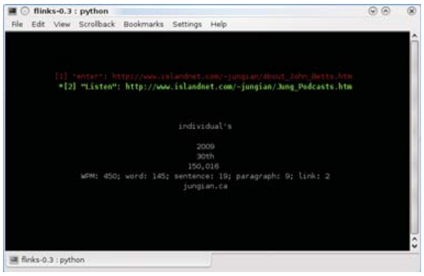
Flinks—Speed-Reading Web Browser

mbays.freeshell.org/flinks

I'm always on the lookout for original projects, and this particular application really took me by surprise. According to its Web site, "Flinks is a text-mode flashing word Web browser. It is intended for speed reading and/or skimming Web pages and text."

Martin Bays, the author of the project, was first inspired by a mobile-phone program that rapidly flashed words one after the other on screen, which promised reading speeds of 600–700 words per minute (WPM) after a few days of practice. Intrigued by the idea and put off by the thought that such a technically simple program cost $20, Martin set about making Flinks.

Installation Getting Flinks up and running is very easy, because no real installation is necessary, and its dependencies are very minimal. All you need is Python, version 2.4 or later, along with a working version of Lynx.



```
[1] "enter": http://www.islandnet.com/~jungian/About_John_Betts.htm
*[2] "Listen": http://www.islandnet.com/~jungian/Jung_Podcasts.htm


                         individual's

                             2009
                             30th
                            150,016
        WPM: 450; word: 145; sentence: 19; paragraph: 9; link: 2
                          jungian.ca
```

Although I can't really demonstrate the effect on printed page, Flinks rapidly flashes individual words of a Web page one after the other in rapid succession to help train your brain for speed reading.

Download the latest tarball from the Web site, extract it, and open a terminal in the new folder. At the command line, enter:

```
$ ./flinks
```

Usage Given the simple design, Flinks is best used with Web sites consisting of mostly text and easy navigation. A good example of this is Wikipedia, and as a result, Martin has chosen it as the default Web site upon opening Flinks.

To get going, press the g key to enter a URL (you don't have to enter in the "http://", something like "metallica.com" is fine). Once the page loads for a few seconds in the browser, press the spacebar to "play" the Web page. Pressing the spacebar again will pause the browser. At this point, a great deal of words starts flying at you, one after the other at a rapid speed. The effect is kind of startling at first but pretty darn cool.

Flinks is set to a default speed of 450 WPM, but if you want to speed up or slow down the word rate, the up and down arrow keys adjust it accordingly. The left and right arrow keys allow you to skip through sentences, and the / or ? keys let you search forward or backward within the text. Other basic navigation includes b to go back, u to go "unback" and q to quit the program.

In the end, this program is a great trip. Put on some Speed Garage (or conversely some Stoner/Prog rock for a time-warp effect) in the background while you try reading at 700 WPM, and you'll be in hyperspeed geek heaven. And, to quote Martin himself:

> But perhaps the most important plus of using Flinks is that having this almost direct jack from the computer to my brain makes me feel like I'm living in the future. And the future is in text mode, just as I always dreamed it would be.

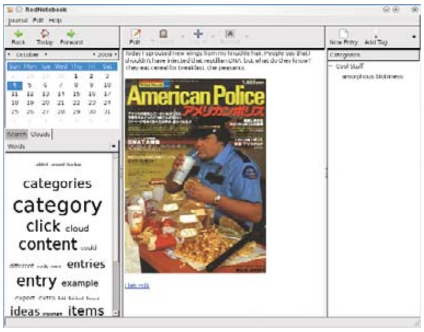
Sounds good to me.

RedNotebook—Advanced Diary Keeping

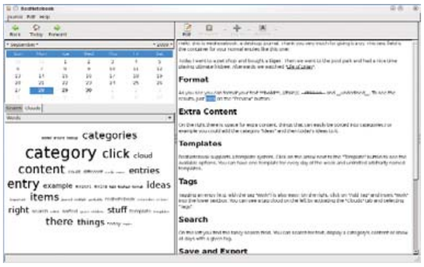
digitaldump.wordpress.com/projects/rednotebook

Finally, this month, we have RedNotebook, a nifty little diary application. According to its Web site:

> RedNotebook is a graphical diary and journal helping you keep track of notes and thoughts. It includes calendar navigation, customizable templates, export functionality and word clouds. You also can format, tag and search your entries.



RedNotebook lets you keep a daily journal with organizational categories, multimedia attachments and more.



RedNotebook has some unique organizational structures and search functions, such as its unique cloud interface.

RedNotebook's many features include the ability to add text, images or links to any day within the excellent calendar navigation; backup utilities; HTML exportation—the list goes on.

Installation Installing RedNotebook from source is quite easy, but there also are a number of different binaries available, so you might want to check those first on the Downloads page. In terms of library requirements, you'll need Python (2.5/2.6), PyYaml (>=3.05) and PyGTK (>=2.13). Depending on your distro, package names will be something along the lines of python, python-yaml and python-devel.

If you're going with the source, download the latest tarball from the Web site, extract it, and open a terminal in the new folder.

If your distro uses sudo, enter:

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

If not:

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

If Lady Luck is smiling, an entry for RedNotebook may appear in your main menu in the Office section.

Usage When RedNotebook starts, you should be at today's date on the calendar automatically. You can attach journal entries to each day on the calendar, and you can have text along with pictures, links and so on. Near the top right is the New Entry button. Click that, and you'll be presented with a small window with two fields. The first is to select or create a category (such as Todo, Cool Stuff and so on), and the second is for naming your entry.

Once this is done, the big pane in the center of the window is where you enter text and other material. Write the text for today, and in the top row of buttons is Insert. Use this to add any images, links, formatting or for numerous other options. During the editing process, you'll see each attachment only as bracketed text, but if you click Preview, you can see your work in progress. When you're done, click Save under the Journal menu, and you can create or browse any other journal entries on the calendar and come back to your entry at any time.

Also, well worth looking at are the Search and Clouds sections, which make navigating through your old entries easy and may save you some headaches in the future. There are many more features that are worth covering (especially the ability to encrypt journals), but I'm afraid I'm well out of space for this month!

Overall, this is an intuitive program with an easy installation that should appeal to someone looking for a good journal program that's both well designed and easy to use.■

John Knight is a 25-year-old, drumming- and climbing-obsessed maniac from the world's most isolated city—Perth, Western Australia. He can usually be found either buried in an Audacity screen or thrashing a kick-drum beyond recognition.

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

Rob Purdie
Project Manager
economist.com
amnesty.org
Lullabot-Powered
The most super powered sites in the world are created in Drupal, by you and Lullabot.
Lullabot™
New Lullabot Learning Series training DVDs at Lullabot.com

AN AMATEUR RADIO SURVIVAL GUIDE FOR LINUX USERS

AN OVERVIEW OF COMMON AMATEUR RADIO ACTIVITIES WITH INFORMATION ABOUT HOW TO PARTICIPATE USING A LINUX SYSTEM AND FREE SOFTWARE.

DAN SMITH

Maintaining a Linux-only household is getting easier every day. The large number of people working to port or re-implement desktop and general-use software from other operating environments helps keep the progress going. Enjoying a hobby that is mostly dominated by Windows users is, unfortunately, not nearly as easy. The Amateur Radio culture is one of experimentation and going against the mainstream, but the relatively small number of people pushing innovation are doing so on their platform of choice or comfort, which usually means Windows. As a result, many applications are not available for Linux.

In this article, I cover some of the basic tasks that a Linux user venturing into the world of Amateur Radio might be interested in doing without undergoing a significant lifestyle change. First, here's a quick vocabulary lesson for those not familiar with some common radio terms.

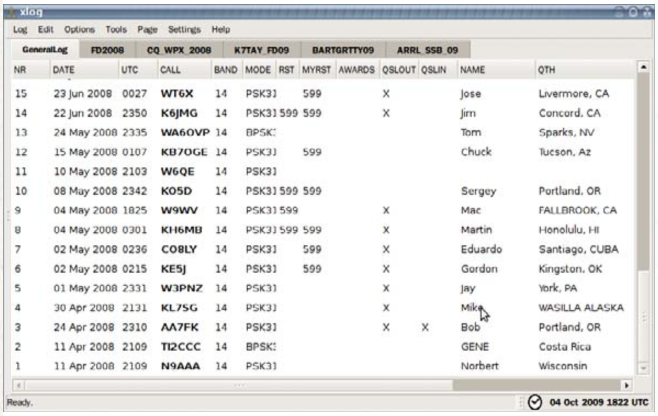
The HF (High Frequency) bands are frequencies between about 1.8MHz and 30MHz, starting just above the US AM broadcast radio band. Signals on these frequencies are capable of traveling around the globe, thanks to the upper ionosphere. If you want to talk to another country, these are the bands for you.

The VHF (Very High Frequency) and UHF (Ultra High Frequency) bands are frequencies above 30MHz up to about 3GHz. Signals in this range propagate in an increasingly line-of-sight manner and, thus, are mostly useful for local area (VHF) and very short range (UHF) communication. However, these frequencies also provide an opportunity for increased bandwidths and data rates, which is why Wi-Fi sits at the upper range (2.4GHz).

CONTEST LOGGING

One popular activity on HF is "contesting", which involves making long-distance contacts to achieve some sort of goal. This usually involves making as many contacts to different places as possible in a certain period of time. Because it is a contest, some sort of log is needed to record the contacts you make for later submission. Because Amateur Radio operators use call signs to identify each other, most logging software helps identify people you've already "worked" to avoid duplication.

The Xlog program for Linux provides basic contest log functionality, including duplicate checking. It also can interface to your radio via a serial port to record other bits of information about a contact automatically, such as mode, signal strength and frequency. Each contest specifies a different piece (or pieces) of information that must be exchanged between operators, so Xlog has some configurable fields to help with that task (Figure 1).



NR	DATE	UTC	CALL	BAND	MODE	RST	MYRST	AWARDS	QSLOUT	QSLIN	NAME	QTH
15	23 Jun 2008	0027	**WT6X**	14	PSK31		599		X		Jose	Livermore, CA
14	22 Jun 2008	2350	**K6JMG**	14	PSK31	599	599		X		Jim	Concord, CA
13	24 May 2008	2335	**WA6OVP**	14	BPSK3						Tom	Sparks, NV
12	15 May 2008	0107	**KB7OGE**	14	PSK31		599				Chuck	Tucson, Az
11	10 May 2008	2103	**W6QE**	14	PSK31							
10	08 May 2008	2342	**KO5D**	14	PSK31	599	599				Sergey	Portland, OR
9	04 May 2008	1825	**W9WV**	14	PSK31	599			X		Mac	FALLBROOK, CA
8	04 May 2008	0301	**KH6MB**	14	PSK31	599	599		X		Martin	Honolulu, HI
7	02 May 2008	0236	**CO8LY**	14	PSK31		599		X		Eduardo	Santiago, CUBA
6	02 May 2008	0215	**KE5J**	14	PSK31		599		X		Gordon	Kingston, OK
5	01 May 2008	2331	**W3PNZ**	14	PSK31				X		Jay	York, PA
4	30 Apr 2008	2131	**KL7SG**	14	PSK31				X		Mike	WASILLA ALASKA
3	24 Apr 2008	2310	**AA7FK**	14	PSK31				X	X	Bob	Portland, OR
2	11 Apr 2008	2109	**TI2CCC**	14	BPSK3						GENE	Costa Rica
1	11 Apr 2008	2109	**N9AAA**	14	PSK31						Norbert	Wisconsin

Ready.

04 Oct 2009 1822 UTC

Figure 1. The Xlog program records contacts you make with other stations.

HF DIGITAL MODES (PSK, RTTY AND OTHERS)

Another very popular activity on the HF bands is operating the slow-speed digital modes. Some of these modes, such as RTTY (radioteletype), predate modern digital computers. Others, like PSK31 (phase shift keying, 31 baud), are fairly recent inventions that use advanced signal processing to their advantage. Although external hardware (digital and analog) previously was used to operate these modes, it now is very common to use a modern soundcard to encode and decode the signal, much like a modem does for a telephone line.

The most common application for doing this sort of work on Linux today is called Fldigi. With a soundcard, serial port and some interfacing to your radio, you can transmit and receive these digital signals without any significant expense. The Fldigi software supports a large number of operating modes, allowing you to communicate via keyboard-to-keyboard text with other amateur operators around the world.

In addition to conversing directly in real time with other amateurs, you also can use Fldigi to record and report the signals it hears in an autonomous fashion. By leaving your radio on the standard PSK31 calling frequency, Fldigi will listen for and report the call signs and locations it hears to a public database. This is very valuable information when comparing the stations other locals are hearing, given the differences between your locations, antennas and so on. It also gives you an idea of what time of day signals from a particular part of the globe are reaching you, in case you want to contact someone in a specific place. If you're interested in this sort of operation, check out the live map (see Resources) to find out who is hearing whom right now.

Although it may seem quaint and obsolete, if you've never had a half-duplex text conversation at 31 baud with someone on the other side of the world, you don't know what you're missing—double that if you've ever done it with nothing more than a battery, a radio and a piece of wire hung in a tree!

VHF/UHF PACKET

When most people think about Amateur Radio, they picture a geek sitting by a big radio under an even bigger antenna clicking out letters and words in Morse code. If you still have this stereotype stuck in your head, perhaps the following discussion will clear things up. Although the HF bands are used for long-distance communications on low frequencies using large antennas, VHF is more about local communication using higher frequencies and smaller antennas. VHF and UHF are the bands used by everyone from taxi drivers to police radios for reliable local communication. Licensed amateur operators are allowed many more privileges on these bands, however, including the use of much higher power and automated stations, such as beacons, message-forwarding systems and data networks.

Wireless networking is a hot topic right now, especially when it involves the use of large transmitters to blanket a wide area with network service. Did you know that Amateur Radio operators have been doing this since the mid-1980s? Albeit at a much slower speed, amateur packet networks have been in place for a long time, providing very wide-area access to various types of networks, including the Internet.

Historically, amateur packet networks have operated using a protocol suite called AX.25, which provides both Layer 2 and Layer 3 functionality. More common today, however, is the use of AX.25 at Layer 2 with IP at Layer 3. Linux has had AX.25 (and obviously IP) support integrated into the kernel for quite a long time. With this support, a $150 radio and a $75 TNC (terminal node controller) interface, you can link your Linux box to a packet network using IP at about 1,200 baud. It's definitely slower than even the slowest Wi-Fi links on a bad day, but you can enjoy this link for tens of miles with only modest equipment and longer with a little effort. You won't want to do any serious Web surfing over this link, but it's actually not too bad for a Telnet or POP3 session. With a little more expensive hardware, you can move up to 9,600 or 19,200 baud, which is similar to the speed offered by many modern satellite phones.

Most distributions include AX.25 support in the kernel and require installation of only the userspace tools package(s) to get started. Extensive configuration is beyond the scope of this article, but the following steps are enough to get an IP link on the air.

First, configure the AX.25 port by putting something like the following in /etc/ax25/axports:

```
radio KK7DS 9600 255 2 MyRadio
```

Next, attach your serial TNC to the AX.25 interface and give it an IP address:

```
# kissattach /dev/ttyS0 radio 44.1.2.3
```

Configure the TNC parameters for transmit delay and so forth:

```
# kissparms -p radio -t 100 -s 100 -r 25
```

At this point, if the other end is set up similarly, you should be able to ping, Telnet or do whatever else you want over your IP-over-AX.25 interface. To bring the interface down, simply `killall kissattach` to disconnect.

One of the privileges we are not granted is encryption over the air. The reasons for this are well documented in Part 97 of the Federal Communications Commission's rules, so I won't go into them here. However, you might wonder how you can run any sort of a secure system without the use of modern encryption. Obviously, care must be taken, and privacy never can be achieved. However, clever use of a one-time-password scheme can provide some protection against casual password sniffing. I have effectively utilized the One Time Passwords in Everything (OPIE) tools by configuring the system to use pam_opie instead of pam_unix for authentication.

D-STAR: DIGITAL SMART TECHNOLOGIES FOR AMATEUR RADIO

Most of the technologies I have discussed so far originated in the previous century and, thus, are quite old. One newcomer to the scene is the D-STAR system. Describing both a new on-air mode as well as a system for interconnecting nodes over the Internet, D-STAR has brought some 21st-century functionality to the hobby.

The ever-growing D-STAR network consists of repeater systems that provide RF access for radios in a local area, and which are connected to each other via the Internet. Much like early cell-phone systems, calls can be routed between users on the system without needing to know which "cell" the other is in at any given moment. With the exception of the infrastructure (which runs exclusively on Linux, by the way), D-STAR is mostly a voice mode, so it doesn't present much of a problem for Linux users.

There are, however, three things that D-STAR provides that may prompt you to want to plug your Linux computer in to your radio. The first is slow-speed data transmission, which involves transmitting data alongside the voice traffic at slow speeds. This can be useful for position reporting (all D-STAR radios can be plugged in to a GPS directly) as well as transferring small files. For this, the D-RATS software (see Resources) can be used and is natively supported on Linux.

In addition to slow-speed data transfers, D-STAR has a high-speed transport that doesn't waste any space with voice traffic and provides 128kbps of throughput over coverage areas of tens of miles. Luckily, this system behaves just like an Ethernet bridge, which means Linux is an equal-opportunity player here as well.

Finally, the D-STAR network can be accessed without a radio at all, using a piece of hardware called a DVDongle. This dongle encapsulates the voice encoder used in D-STAR radios and, when coupled with software on a desktop computer, allows you to connect to the Internet-linked infrastructure and communicate by voice with radio users in another area. This is a marriage of VoIP-like functionality with radio that has become very popular among D-STAR enthusiasts. The good news is the software for this also runs natively on Linux!

WHAT ELSE IS OUT THERE?

I have covered only some of the more-common activities that Linux users looking to venture into the world of Amateur Radio are likely to be interested in. There are many other interesting areas of activity, and in almost all cases, there is a way to participate as a Linux user. For example:

- APRS (Automatic Packet Reporting System): location-aware services are starting to emerge as our mobile telephones gain the hardware, connectivity and software required to support them. As with many things, Amateur Radio operators are ahead of the curve here as well. For more than two decades, operators have been pairing radios and modems with GPS receivers to transmit their positions to others. Initially used for local-area positional awareness during public events, the system now is interfaced to the Internet, providing a way to track people from a Web browser using Google Maps. APRS is easy and inexpensive to use, and it provides features and functionality above even some so-called modern services. On a recent trip, I was more than 100 miles away from cellular coverage, but my friends were

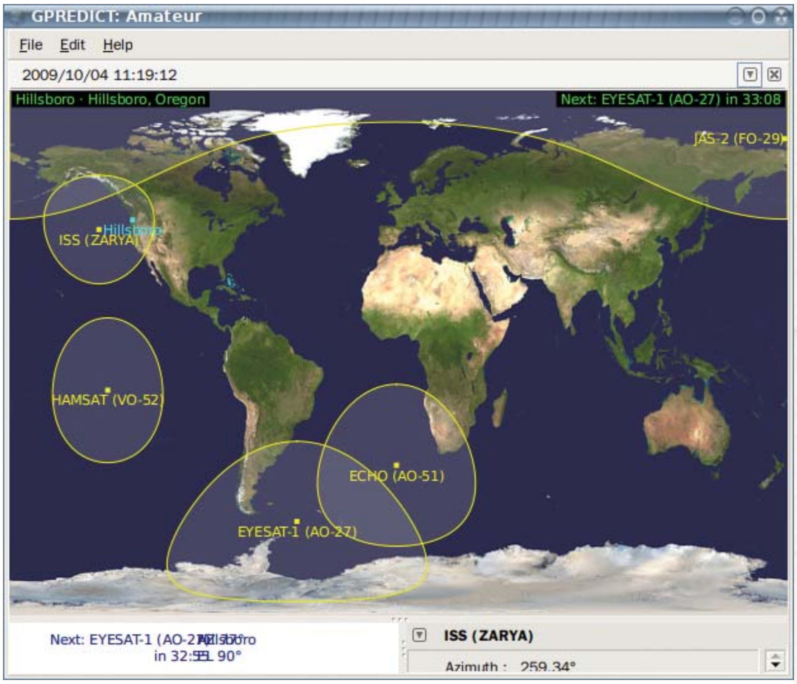


Figure 2. The GPredict program shows where amateur satellites are in real time.

able to track my position through the APRS system.

- Satellite communications: did you know that Amateur Radio operators have satellites in space for the sole purpose of facilitating communications? With an inexpensive radio and some creative antenna work, you can communicate with an amateur satellite for the purposes of digital exchange or even analog voice calls. Linux users are not excluded from this activity. If your distribution provides the gpredict package, take a moment to install and run it. You might be surprised to see an amateur satellite passing overhead as you read this (Figure 2).

- SDR (Software Defined Radio): historically, radios have been complex purpose-built analog devices that depend on a lot of filters, resonant circuits and other components. Increasing performance of these devices often means adding additional filter stages and, recently, embedding Digital Signal Processors (DSPs) to cut noise and block unwanted interference. A new generation of radio technology recently has emerged that uses very simple, wideband, general-purpose radio components and relies on the high performance of a modern PC to do all the hard work. The result is what is called a Software Defined Radio or SDR. These systems have incredible sensitivity and are extremely flexible. Several projects exist that are working to push the envelope of this new technology, but the most interesting to Linux users probably is the GNU Radio Project (see Resources).

Clearly, there are many natural interactions between the Open Source community and the Amateur Radio community, and you will be amazed and surprised at how many current projects have been inspired by Amateur Radio or adopted by it.■

Dan Smith, KK7DS, is a software engineer for IBM's Linux Technology Center in Beaverton, Oregon. He enjoys playing Amateur Radio on the weekends and is the author of the D-RATS software.

Resources

Fldigi: **www.w1hkj.com/FldigiHelp/Modes/index.htm**

Live Map: **psk.gladstonefamily.net/pskmap.html**

HAMSOFT: Linux Software for the Hamradio Community: **radio.linux.org.au**

GNU Radio: **gnuradio.org**

D-RATS: **d-rats.com**

Xastir

Open-Source Client for the Automatic Packet Reporting System

What do search and rescue, Amateur Radio and Linux have in common?

In the early 1990s, Bob Bruninga, an instructor at the United States Naval Academy in Annapolis, devised an interesting stunt: he wanted to track the Army/Navy game football on its travels from Annapolis to Philadelphia, about 150 miles away. To do this, Bruninga stuffed a small electronics package into a football helmet consisting of a GPS receiver, an Amateur Radio transmitter and a radio modem. At the time, GPS receivers were quite expensive (and a novelty), and cell phones still were quite new and incapable of doing data. Bruninga's real innovation, however, was to plot the received position reports on a computer map display and automatically track the position of the midshipmen carrying the game ball (see Resources).

That was one of the first uses of the Automatic Position Reporting System (APRS)—originally, Amateur Position Reporting System. APRS has evolved considerably in the decades since. The protocol has been greatly enhanced to include automated weather station reports, status messages and two-way text messaging. APRS communications systems have evolved beyond simple transmitters and receivers to sophisticated networks that encompass Amateur Radio satellites, digipeater (digital repeater) automatic relay systems and Internet gateways. APRS technology is not confined to Amateur Radio; it is used in numerous commercial applications as well.

APRS is intended to provide a situational awareness display or tactical display. Everything of note should be displayed easily on the map with additional detail and messaging available at a click. In emergency operations, a quick glance reveals what resources are available and where.

Note: APRS is a registered trademark of APRS Software and Bob Bruninga, WB4APR.

CURTIS E. MILLS, STEVE STROH AND LAURA SHAFFER MILLS

Introducing Xastir

Imagine a map display on your laptop with moving symbols representing the current positions of your ham radio friends and acquaintances, while they can see your real-time position on their displays. Imagine being able to "instant-message" any of them as well. These are a small subset of Xastir's capabilities, and they can be accomplished with a small amount of equipment and an entry-level Amateur Radio license (see sidebars).

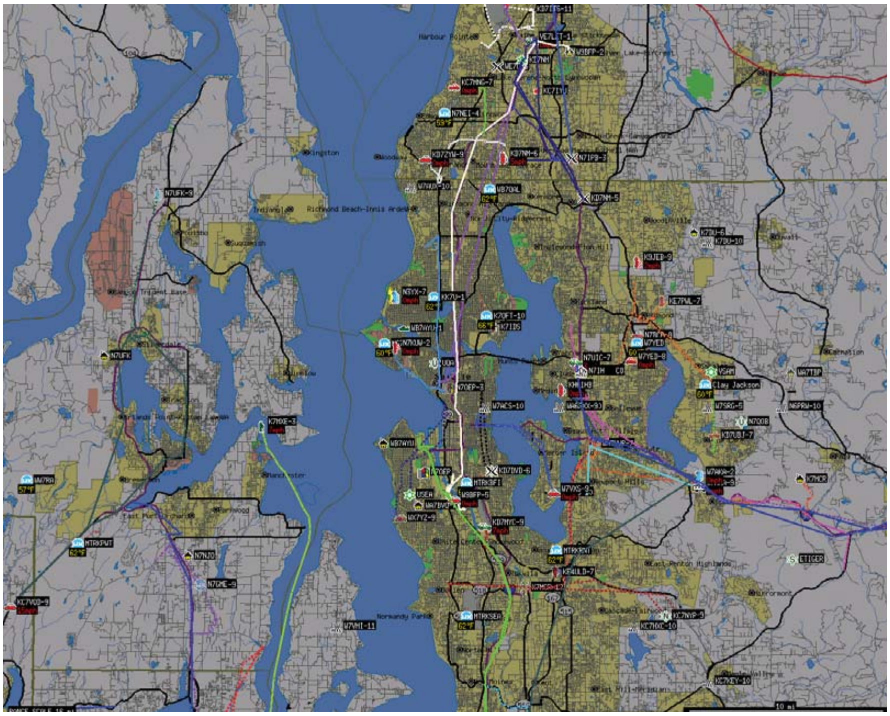


Figure 1. Xastir Centered on Seattle, Washington (Maps Courtesy of US Census Bureau Tigermap Server)

Xastir is open-source software, which aims to be compliant with and interoperable with the APRS protocol. Frank Giannandrea, KC2GJS, wrote the first version of X Amateur Station Tracking and Information Reporting (XASTIR) for Linux and released it under the GPL license. As with many open-source projects, even though Frank has retired from the project, a team of developers has continued and considerably extended Frank's original work. Xastir is arguably one of the most capable APRS implementations and has an active user community.

Xastir can be compiled and run on Linux, FreeBSD, Mac OS X, Solaris, HP/UX and even Windows. Today Xastir can display objects and their associated status (and messages) in real time on Internet-based or local maps, enable two-way messaging between stations and many more functions. Some Xastir users are involved in search and rescue (SAR), others in helping out at public service events or Amateur Radio emergency organizations, but many use it just for fun.

You can get started in Xastir with Internet-based maps and data streams. Advanced users have radio interfaces connected to laptops or touchscreen trunk-mount PCs in their vehicles where it gives a tactical display of nearby stations while providing mobile mapping and letting everyone know their current position. Xastir also can speak via the Festival speech synthesizer.

Note: although Xastir is described here as Amateur Radio software, nothing specifically ties its use to Amateur Radio. There are alternative radio systems that require no licensing and allow data transmissions between stations. Networks can be deployed using such systems and Xastir; however, you would lose access to the rich set of interconnected networks that exist in the Amateur Radio APRS system.

Linux APRS Bits

Linux is special and not just for the reasons most of you already know. Linux was the first operating system to have the Amateur Radio AX.25 packet protocol actually built in. Currently, it's a pluggable kernel module, but in the early days, we had to compile custom kernels to enable it.

AX.25 is based on ITU-T X.25, with a few additions that let us store and forward or digipeat (digitally repeat) a packet through multiple stations on RF (radio frequencies). This allows us to create a network from multiple radio stations on the same frequency. To the computer, an AX.25 port looks like any other networking port, so you must guard against sending out broadcast packets from some common Linux applications that can trash our low bit-rate radio channels. Firewalls or specific configuration of each errant dæmon are the fixes.

Once you set up an AX.25 interface, you can share it across multiple applications, run special digipeater or server software, or run multiple protocols across it. See the AX.25 HOWTO on the Xastir Wiki (see Resources) or the older AX.25 HOWTO for setup details.

Note: a simpler method for APRS only is to use the Serial KISS TNC interface within Xastir to talk directly to a serial port, USB→Serial adapter, or soundmodem terminal node controller (TNC). This method does not let you share the TNC across multiple applications, however.

For the TNC, there are several options. Software only (mostly?) options include the soundmodem driver on Linux/Solaris or AGWPE on Windows.

INFO: What are these odd groups of up to six letters and numbers after people's names? Amateur Radio call signs! Like television and radio stations, each ham has a unique call sign representing a transmitting station.

See the soundmodem HOWTO on the Xastir Wiki for details (see Resources).

Most operators prefer a hardware TNC over the software solutions above. We recommend KISS mode for long-term reliability. The alternate TNC mode, command mode, was designed for human interaction and, thus, is prone to timing errors and commands getting lost or misinterpreted. KISS is a much simpler protocol; the computer does most of the protocol processing work instead of the TNC.

Internet APRS Bits

An entire tiered network of servers moves packets around the Internet and connects the various radio networks together, called the APRS-IS (APRS Internet Services). With proper gateways, two APRS stations anywhere in the world can exchange two-way messages. Alternative networks exist for Citizen's Weather (CWOP) data and for emergency-minded or weather-minded folks who wish to see more objects (Firenet.us). Firenet is one of Curt's interests; he contributes earthquake, weather, fire and river gauge scripts, also plotting positions of steam-train excursions.

Home stations can have a radio interface, can connect as clients of APRS-IS or Firenet, and optionally may choose to be a one-way or two-way igate (Internet gateway) between the Internet and radio. One-way igates (RF→INET) are welcome any time, but two-way igates are best coordinated so that they don't compete for limited airtime in one area. The typical two-way igate passes only APRS messages and ACKs/NAKs for stations it has heard recently on the local frequency. Mobile travelers can send and receive messages from their current locations, as the igates will route the messages appropriately. Xastir can function as such an igate, but if you need more advanced igate functions, other applications may be better suited.

Other important pieces of the infrastructure are Internet databases with Web front ends. See the Resources section of this article to find out more, specifically the info.aprs.net link, and click on APRS Internet System—it'll be well worth your time.

Xastir Major Features

Compared to other APRS clients, Xastir really excels in the map department with 125 types of maps, including topo maps, street maps, aerial photographs, weather alerts and radar images. It can use on-line or local map sources, or combinations of both. Map stackups

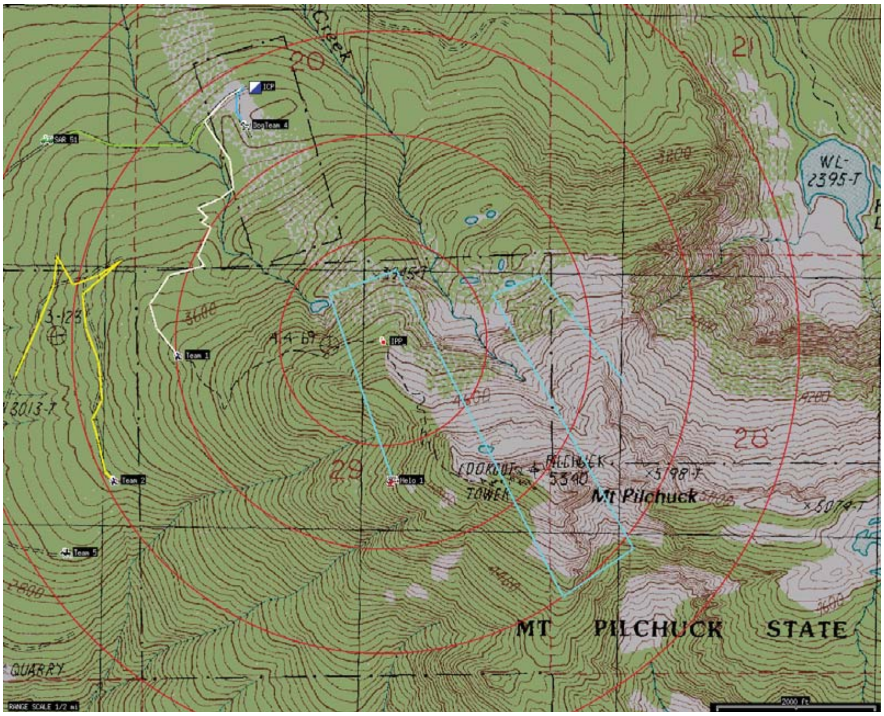


Figure 2. Xastir Topo Maps, Simulated SAR Mission, Mt. Pilchuck (Maps Courtesy of USGS)

Glossary

Amateur Radio has been doing digital communications literally since there has been wireless communications. We first used CW (Continuous Wave) or On/Off keying with Morse code. Abbreviations quickly became commonplace. After WWII, surplus teletypes (the heavy, loud electromechanical monsters you see in old movies) became available at reasonable prices and ham radio RTTY (radio teletype) was born, and keyboard shortcuts and abbreviations became even more commonplace.

We use a lot of abbreviations—Q-codes during voice and especially CW conversations (CW or Continuous Wave means Morse code). Even the word conversation has an abbreviation: QSO.

Quite a few of our abbreviations have roots in the Morse code world, spilling over into our e-mail and instant messaging. If we type "hi hi" to you, that means laughter. "QTH" equals location. This is similar to instant-messaging abbreviations that many of you use with SMS messaging or e-mail.

Here are some other common terms:

- **APRS:** Automatic Position Reporting System.
- **AX.25:** the packet radio link-level protocol we use.
- **digipeater:** single-frequency store-and-forward digital repeater for radio.
- **KISS mode:** Keep It Simple Stupid protocol. A simplified protocol used in a TNC for computer control (yes, that really is the name of the protocol).
- **soundmodem:** software to turn a soundcard into a TNC.
- **TNC:** Terminal Node Controller or radio modem.
- **tracker:** a GPS/radio/TNC that emits positions while moving. It may or may not have two-way messaging or a map display.

can be custom-designed by the user in nearly any combination.

We do not support or condone use of proprietary map formats, but there are plenty of free maps from which to choose. You also can grab screen captures of maps you own and easily georeference them for use with Xastir. For those coming from the Windows side, UI-View maps can be converted with our inf2geo.pl script for use with Xastir. This helps users outside North America immensely, as they often lack access to the free maps we enjoy here.

For contiguous topographic maps from USGS DRG maps, Xastir topo code crops the white borders and stretches the image into a rectangular shape while drawing. Maps are stitched together into a contiguous view.

Xastir supports direct or network connection to weather stations, GPSes and TNCs. Several coordinate grids/systems are supported: UTM/UPS, MGRS, USNG, Maidenhead grid and multiple latitude/longitude formats. A Coordinate Calculator is included that converts between many of these formats.

Menus can appear in any of seven real languages and five "not so real" languages. Curt took a bit of heat from other developers for adding those last five "languages", but you know beer-fueled coding sometimes yields interesting results! All kidding aside, he was asked to support Talk Like A Pirate Day (September 19th every year). He found pirate-speak REGEX plus several other "languages", so he added five in total. He's proud to be a supporter of Talk Like a Pirate Day, the only international holiday created from a sports injury!

Dig in to the menus, and you'll find some SAR-specific features, as well as features useful for public-service event tracking. Xastir is used for SAR, marathons, bicycle, motorcycle and car events.

Though Xastir and APRS are not well suited or intended for this application, one ham, when his vehicle was stolen, was able to tell police the car's current location, direction and speed. It does take some work though to explain to the police how you know this.

How to Use Xastir

First and foremost, sign up on the Xastir users mailing list (see Resources). Many helpful experienced users and quite a few of the developers hang out there. Posting requires a subscription, so sign up before you need it. Lurking and "dumb" questions are fine. Everybody learns or relearns a bit, and newbies are not harassed there.

Check the Xastir home page (see Resources), specifically the "XASTIR Documentation" and "Text docs distributed with Xastir" sections for useful documentation. Of course, look in the Xastir source main directory or the

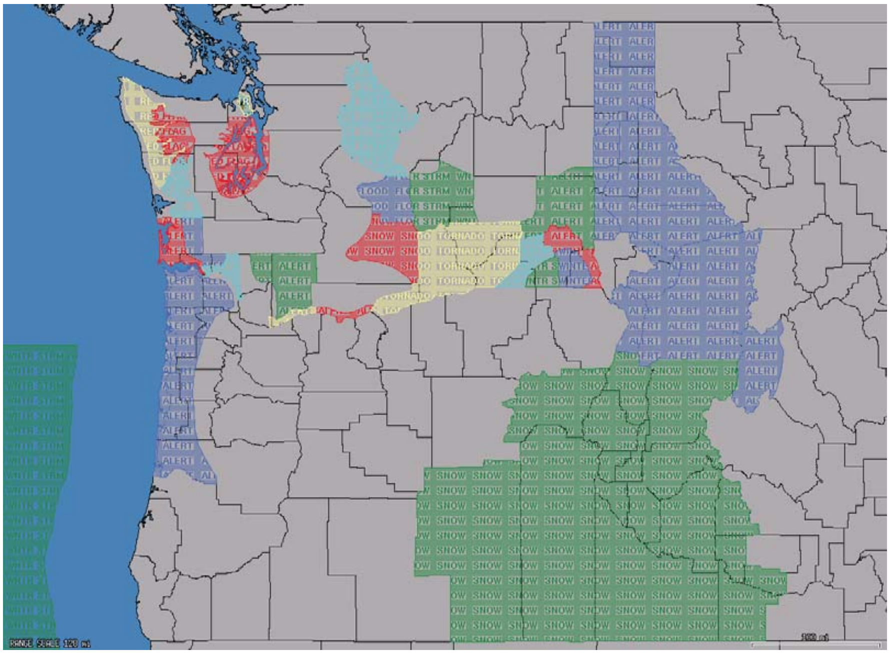


Figure 3. Xastir Weather Alerts (County Map Courtesy of National Weather Service)

Getting an Amateur Radio License

Many people think of Amateur Radio operators as old guys sitting in the basement in front of a radio with glowing tubes, tapping on a Morse code key.

The good news is that Morse code is no longer a requirement in the US. Tubes have been replaced by transistors, integrated circuits, embedded processors and software defined radio (SDR). We have satellites in orbit, had digipeaters on the international space station and (now expired) MIR, and do EME and EVE signal bounces (Earth→Moon→Earth and Earth→Venus→Earth). It's not your grandpa's ham radio anymore!

License types have been streamlined to three: Technician Class, General Class and Amateur Extra Class. Each license grants additional privileges, mostly in the form of additional allowed frequencies. The entry-level Technician Class allows you to participate in the APRS network described in this article along with many additional privileges.

Many barriers to getting a license have been removed. Tests are no longer given by the FCC field offices, but by volunteer examiners. In large urban areas, there may be multiple places to take a test each month. Practice tests are available on-line (try **www.qrz.com** and others). For more information about Amateur Radio licensing, or where to find testing in your area, go to **www.arrl.org** and click on Licensing.

Note: the information above is US-centric. Each country has a regulatory body governing Amateur Radio and often a national Amateur Radio organization. Contact your national organization or regulatory body for specific details regarding Amateur Radio licensing in your country.

Equipment Needed for Xastir/APRS

1) Minimum requirements: a live Internet feed, Linux (or another UNIX/UNIX-like operating system) and Xastir.

2) Adding Amateur Radio to the mix: Linux, a soundcard, the soundmodem package, a two-meter ham radio, a valid ham radio license and Xastir. It is possible to configure this setup with or without the Linux AX.25 library.

Note that it's completely legal to own Amateur Radio equipment without an Amateur Radio license, as long as you use the equipment *only* to receive. Transmitting on Amateur Radio frequencies without a license is blatantly illegal, and the community is capable (and motivated!) to hunt down troublesome rogue transmissions, and the FCC has a long tradition of prosecuting miscreants.

3) A more reasonable station: Linux plus a hardware TNC in KISS mode, a two-meter radio and a valid license. Again, one can configure with or without the Linux AX.25 library.

To build a portable or mobile tracker, many options exist that don't involve carrying around a full-blown laptop. Go to **info.aprs.net**, and click on Hardware to see loads of options.

default install location of /usr/local/share/doc/xastir/ for the text docs as well.

Run Xastir for the first time, and it takes you to the File→Configure→Station dialog to enter your call sign and location and to choose a symbol that will appear on others' maps. Some symbols have special significance, so if you have a question, please ask on the list.

Visit the File→Configure→Defaults dialog next. Note that antenna height is not height above sea level (ASL), but height above average terrain (HAAT). Curt once lived at 150 feet ASL but –450 feet HAAT due to surrounding hills, but the protocol doesn't allow for negative HAAT; perhaps they figured no ham would live under such conditions!

If you have Internet connectivity, go to Map→Map Chooser, click on Online/Tigermap.geo, then click OK. If you're located in or near the US and zoomed in sufficiently, you should see Tigermap street maps appear shortly. If you don't have connectivity or are missing libraries to fetch maps, stick to built-in vector map formats like ESRI Shapefile. A map of the world in ESRI Shapefile format is included.

Hint: start Xastir from an xterm to catch error and warning messages. Also, check File→Configure→Timing→Internet Map Timeout to ensure it's adequate.

You probably will want to see other stations, weather alerts and the like. Go to Interface→Interface Control. Click Add, then Internet Server, then Add. You should see a new Internet Server entry on the Interface Control dialog. Click on that entry and then Properties. Enter server and port information into the form. Try rotate.aprs.net port 14580 with filter parameters of m/600, which causes reports for all stations within 600km to appear on the map. Enter a passcode to log in to the server, and see the callpass program that comes with Xastir. Send a position to the server if using the range filter, or the server won't know which stations to send you.

Run this script as root to fetch Shapefile maps for weather alerts:

```
xastir/scripts/get-NWSdata
```

For mouse functions, turn on emulate third button in your OS if you have a two-button mouse, as Xastir can use all three buttons of a three-button mouse. See the text docs distributed with Xastir for details on each function. Toggle buttons at the top of the screen change how the mouse buttons work, specifically Measure, Draw and Move. The cursor changes when in these modes to give visual verification. While we're at it, a few defines in xastir/src/main.c allow swapping mouse buttons and touchscreen operation, for those who aren't afraid to compile their own.

A few special keys include HOME (centers the map on your house or car), PageUp/PageDown (zoom keys), arrow keys (pan keys). Toggle buttons at the top of the screen provide most of these functions also.

Transmit Paths

The AX.25 protocol has multiple slots in the header of each packet for digipeater call signs, which APRS uses for a flooding protocol. One packet floods an area out to X digipeaters in all directions. Think of it as a time-to-live count or as a distance limit for each packet.

Currently accepted digipeater paths include WIDE2-2 or WIDE1-1,WIDE2-1 for home stations and WIDE1-1,WIDE2-1 for mobile stations. Either path will give you two hops outward in all possible directions. Never use WIDE1-1 as anything but the *first* digipeater call sign in the path; it triggers home station fill-in digis as well as mountain-top digis.

By following the above (currently only North American) recommendations, you can travel between RF networks without reconfiguring your tracker. High-altitude balloon or airplane-mobile trackers likely will need to reduce their paths to prevent interfering with many RF networks at once. Rocketry enthusiasts often run high packet rates on alternate frequencies due to their short flights (five or ten seconds); this avoids interference with the "normal" APRS network on 144.39MHz.

For "tin-foil hat" types, there are special path aliases for the end of your path to keep packets from being gated to the Internet: NOGATE and RFONLY. These are respected by most igate software.

Please stay abreast of the current national recommendations for your country—the above recommendations may not apply. If you're not part of the solution (and up to date), you're part of the problem, which definitely applies here!

Tactical Calls for Special Events

A very useful recent addition to Xastir is the capability to assign tactical call signs to stations and to transmit them across RF or the Internet to other APRS clients (Xastir and APRS+SA are the only two clients that currently decode and display tactical call assignments).

We added this capability just prior to the Seattle Marathon in November 2008 and used it to great effect at that event. It allowed us to assign call signs, such as LEAD and TAIL, to APRS stations helping in the event, without having to change their tracker configurations. This aids reliability when we're not tweaking tracker settings before the event. The old "if it ain't broke, don't fix it" method really does work when dealing with mobile trackers. Don't touch a working tracker unless you have to.

Xastir also was used to good effect in the New York City Marathon. Officials love it for its instant and eye-catching tactical display. You'll see them hanging around the Xastir display when they're not otherwise engaged.

SAR tactical call signs provide similar advantages, assigning TEAM 1, HELO 2, DOGTEAM 3 and similar monikers to teams in the field. Special features have been added to Xastir to support SAR operations.

Resources

Xastir Home: **www.xastir.org**

Mailing Lists: **lists.xastir.org/mailman/listinfo**

SourceForge Project: **sourceforge.net/projects/xastir**

Soundcard Wiki Page: **www.xastir.org/wiki/index.php/HowTo:SoundModem**

AX.25 Wiki Page: **www.xastir.org/wiki/index.php/HowTo:AX.25**

SAR Page: **www.eskimo.com/~archer/xastir/SAR.html**

APRS Main Pages: **aprs.org**

APRS Wiki: **info.aprs.net**

ARRL: **www.arrl.org**

Practice Tests: **www.qrz.com**

Naval Academy Radio Club's 2008 Army/Navy Football Run *LIVE* Tracking Page: **www.aprs.org/football.html**

Photo of the Original GPS Tracker Installed in the Football Helmet for Runners Who Traveled from an Annapolis Pep Rally to the Stadium in New York: **www.usna.edu/Users/aero/bruninga/foot.html**

APRS Tracking 1993 Football Run: **www.usna.edu/Users/aero/bruninga/football.html**

Xastir's Future

From its meager beginnings as a Linux-specific application, Xastir has grown significantly in capability but is still tied to OpenMotif or Lesstif for its widget set. Xastir is a large multithreaded and multiprocess program. Developers have talked about splitting Xastir into an SQL database back end plus a dæmon with multiple clients to handle user interface functions. Such clients could use different widget sets, such as Qt, GTK+ or WxWidgets, with correspondingly better integration with modern-day window managers. Rewriting to use Qt without the X11 layers underneath might allow for running on Qtopia embedded devices. In any case, whether we refactor Xastir, it has a bright future as its user base continues to grow and useful features are added continually. See you on the mailing list!

Note: the Open Group, Motif, Making Standards Work, OSF/1, UNIX and the "X" device are registered trademarks, and TOGAF and Boundaryless Information Flow are trademarks of The Open Group in the US and other countries.■

Curtis E. Mills, WE7U, became active in packet radio in the mid-1980s, creating a receive-only station from a radio, a single-chip interface circuit and custom assembly code. He's active in search and rescue and contributes to Firenet, Xastir, SmartPalm and gpsbabel development. Other interests are hiking and bow hunting when his kids aren't running him ragged. He's employed as an engineer at Fluke Corporation. Reach him at archer@eskimo.com or as WE7U-3 on APRS. He also can be found on Xastir, NWAPRS and APRSSIG lists.

Steve Stroh, N8GNJ, had his first experiences with TCP/IP networking via Amateur Packet Radio on the Puget Sound Packet Radio TCP/IP network (WETNet) in the late 1980s. From hanging out with that bad crowd of techies, Steve became a sysadmin (on systems lesser than Linux) and in 1997, began writing about broadband wireless Internet access based on his practical experience with wireless gained from being a ham. He's looking forward to some winter projects, including diving deep into open-source wireless mesh networks, embedded ARM-based Linux systems, running IPv6 over Amateur Packet Radio and getting a number of radios back on the air after a long absence. Steve can be reached at steve@stevestroh.net.

Laura Shaffer Mills studied engineering, but prefers writing software in any convenient language. She hopes to learn something new from every project and particularly enjoys solving problems that no one else had found. She relaxes with wire harp or tatting, since both are quite rare. Laura lives with her husband and three daughters and can be found at www.redwriteblue.com.

Example: Seattle Marathon

For the Seattle Marathon, I (Curt) sit in the net control vehicle with the "route" and "start/finish" radio guys. Xastir is running on a Linux laptop with Internet access, along with a remote LCD display for the radio operators to view the current event map. Bob Donnell, KD7NM, has a similar station several hundred feet away at the first-aid station. Trackers are assigned tactical call signs like SAG 1 prior to the event or at the beginning of their assignment.

I listen for runner pick-up or aid requests from the radio operators and send instant messages to Bob. Bob then dispatches vehicles (SAG wagons) via voice radios to pick up the runners. Positions of all SAG wagons appear on both of our maps.

SAG vehicles can't cross the course, so it's tricky getting them to the correct locations. The map display allows Bob to dispatch the vehicle that would be quickest to pick up the runners—you must know how to route vehicles when Seattle is bisected by this event.

We place trackers on the lead vehicles for both the full and half marathons—problematic due to the choice of electric vehicles, which so far haven't completed the mission. If these trackers become unavailable, we shift to monitoring voice reports on the route network and create objects to represent them on the map. The trailing vehicle is driven by a ham who usually can keep its tracker running, although we have the same fallback plan.

Event organizers use the trailing runner information to adjust support needs at the finish line, and we use the information to keep the volunteer hams up to date on the approach of the final runners. The volunteers then share this information with the rest of the marathon volunteers in their area. This helps ensure that we don't leave anyone behind on the course at the end of the day and lets us know when to open streets and when to pick up ham volunteers and equipment.

Rolling Your Own with Digital Amateur Radio

Fldigi to FLDigiROL: an Amateur Radio operator's digital journey to open source.

GARY L. ROBINSON

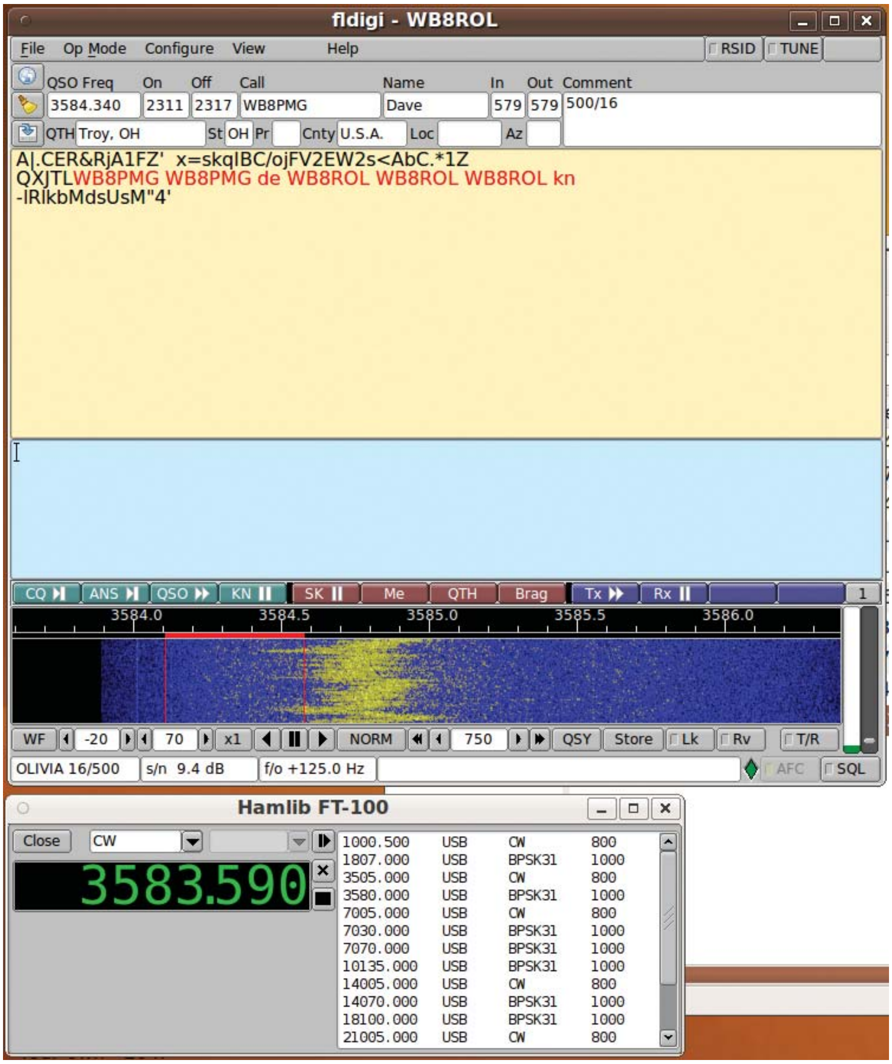
Amateur Radio operators are generally free-thinking individualists who don't mind getting their hands dirty to get something done right. Many of us do not think twice about buying a brand-new radio for hundreds or even thousands of dollars and popping the lid on it to see if we can modify it to make it better. You do not have to look hard to find myriad articles on how to modify different pieces of Amateur Radio equipment. So, it is not surprising that we might feel the same way about the software we use.

Open-source software and Amateur Radio are a natural fit. Few operators ever would buy a piece of radio gear if it came with a license that said they could *not* modify it, and it's natural to see why a lot of us navigate toward open source in general and Linux in particular. My personal computing journey started with DOS in 1990, OS/2 in 1993, Windows in 1998 and Linux since 2000. In the true Amateur Radio tradition, I taught myself how to write batch files in DOS, then started tinkering with Pascal. From there, it went to C and eventually, C++. Then, after learning how to use those languages, I took college-level classes to relearn them the right way. It's almost an Amateur Radio tradition to do things backward sometimes and without the manual first.

I started looking at Linux after IBM killed off its OS/2 operating system in the late 1990s. It took more than a few years until I felt comfortable with Linux, but in the past four years, it has been my primary operating system. For the past two years, it has become my *only* operating system on my three desktop machines and my new Netbook.

Open source on Linux already had supplied me with most of the software I needed, with one exception—a suitable Amateur Radio program for digital soundcard modes. Several programs were available, but compared to a few of the Windows offerings, they were feature-poor, and their user interfaces were not as friendly.

I left Windows on my Amateur Radio



QSO Freq 3584.340 On 2311 Off 2317 Call WB8PMG Name Dave In 579 Out 579 Comment 500/16

QTH Troy, OH St OH Pr Cnty U.S.A. Loc Az

```
A|.CER&RjA1FZ'  x=skqIBC/ojFV2EW2s<AbC.*1Z
QXJTLWB8PMG WB8PMG de WB8ROL WB8ROL WB8ROL kn
-lRlkbMdsUsM"4'
```

-20 70 750

OLIVIA 16/500 s/n 9.4 dB f/o +125.0 Hz

CW 3583.590

```
1000.500   USB   CW       800
1807.000   USB   BPSK31   1000
3505.000   USB   CW       800
3580.000   USB   BPSK31   1000
7005.000   USB   CW       800
7030.000   USB   BPSK31   1000
7070.000   USB   BPSK31   1000
10135.000  USB   BPSK31   1000
14005.000  USB   CW       800
14070.000  USB   BPSK31   1000
18100.000  USB   BPSK31   1000
21005.000  USB   CW       800
```

Figure 1. Original Fldigi v 3.10 with Frequency Control Dialog

computer in a dual-boot configuration with Linux until a few years ago when I discovered an excellent digital program called Fldigi (Figure 1) written by Dave Freese, W1HKJ; Stelios Bounanos, M0GLD; and Leigh Klotz, WA5ZNU. It had all the digital modes I was interested in, and it worked very well. Prior to this, I used the Amateur Radio Deluxe DM780 program on my Windows XP partition for most digital contacts. DM780 also is a very fine program and is written by Simon Brown, HB9DRV. He uses open-source code (some of which came from Fldigi) in the digital decoding DLL files for his program, but part of the program is proprietary and, like many programs, I feel DM780 is starting to suffer from feature overload, and in its quest to do everything, it is beginning to get a bit bloated. So, I was excited to find Fldigi for Linux and started using it immediately.

I used it for several months and was fairly satisfied with it. However, I found some things I did not like and missed a few features from DM780 that Fldigi did not have. (Did I mention that we Amateur Radio operators are a picky bunch?) Fldigi was 95% on the road to where I wanted to be, but that last 5% itched my hide a little bit.

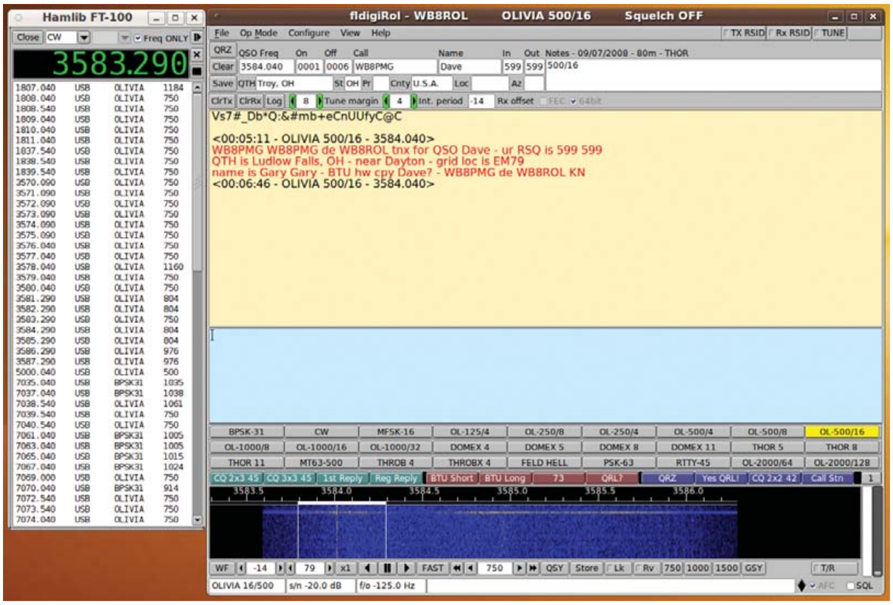
I finally decided to download the open-source code for Fldigi and see what it would take to get that last 5% that would make me happier. I simply could have contacted the authors of Fldigi and sent them ideas and suggestions on how I felt Fldigi could be improved, but everyone's ideas differ as to how software should work and what to expect from it. What might make me happy may not be important to others, or it might be a low priority on a long list. Open source makes it possible for you to have it your way, if you don't mind learning a little and working hard. So, I assembled a list of about 25 (mostly minor) things I thought needed to be changed or added to the Fldigi program.

The documentation for the Fldigi source code was easy to understand, and I rapidly set up a build environment to work on it, including dependencies listed in the Fldigi documentation along with additional code required for development. The user interface part of the program is based on the Fast Light Toolkit (FLTK). FLTK is a lightweight set of libraries that supplies all of the modern graphical controls and window elements most modern computer users would expect to see. I downloaded the documentation for the FLTK and familiarized myself with it as well.

I initially decided to import the whole Fldigi project into a KDevelop project—an easy-to-use IDE that I was already familiar with, which can be used for almost any type of project, not just KDE programs. I could just as easily have used any of a dozen other such environments or a simple text editor and a standalone debugger program. I elected to call my revised version of Fldigi FLDigiROL (Figure 2), adding the suffix of my Amateur Radio call sign (WB8ROL) to the end of the original name.

Many things I wanted to change or add to Fldigi involved the program's graphical interface. The interface is almost ideal for general operation, but parts of it bothered me. I felt I spent too much time accessing menus, submenus and tabbed dialog boxes to change modes and mode parameters as I operated my digital station. Menus and dialog boxes really help organize things and reduce clutter, but I thought that if I had to access certain items continually, perhaps they should be right on the main window. My philosophy always has been that software should do the work and not make me work any harder than I have to.

Amateur Radio digital operation can be challenging, because there are a number of different modes. Fldigi has ten different digital modes, but many of them can be configured in multiple



Hamlib FT-100

Close CW ☑ Freq ONLY

3583.290

```
1807.040   USB   OLIVIA   1184
1808.040   USB   OLIVIA   750
1808.540   USB   OLIVIA   750
1809.040   USB   OLIVIA   750
1810.040   USB   OLIVIA   750
1811.040   USB   OLIVIA   750
1837.540   USB   OLIVIA   750
1838.540   USB   OLIVIA   750
1839.540   USB   OLIVIA   750
3570.090   USB   OLIVIA   750
3571.090   USB   OLIVIA   750
3572.090   USB   OLIVIA   750
3573.090   USB   OLIVIA   750
3574.090   USB   OLIVIA   750
3575.090   USB   OLIVIA   750
3576.040   USB   OLIVIA   750
3577.040   USB   OLIVIA   750
3578.040   USB   OLIVIA   1160
3579.040   USB   OLIVIA   750
3580.040   USB   OLIVIA   750
3581.290   USB   OLIVIA   804
3582.290   USB   OLIVIA   804
3583.290   USB   OLIVIA   750
3584.290   USB   OLIVIA   804
3585.290   USB   OLIVIA   804
3586.290   USB   OLIVIA   976
3587.290   USB   OLIVIA   976
5000.040   USB   OLIVIA   500
7035.040   USB   BPSK31   1035
7037.040   USB   BPSK31   1038
7038.540   USB   OLIVIA   1061
7039.540   USB   OLIVIA   750
7040.540   USB   OLIVIA   750
7061.040   USB   BPSK31   1005
7063.040   USB   BPSK31   1005
7065.040   USB   BPSK31   1015
7067.040   USB   BPSK31   1024
7069.000   USB   OLIVIA   750
7070.040   USB   BPSK31   914
7072.540   USB   OLIVIA   750
7073.540   USB   OLIVIA   750
7074.040   USB   OLIVIA   750
```

fldigiRol - WB8ROL   OLIVIA 500/16   Squelch OFF

File Op Mode Configure View Help   □ TX RSID □ Rx RSID □ TUNE

QRZ	QSO Freq	On	Off	Call	Name	In	Out	Notes - 09/07/2008 - 80m - THOR
Clear	3584.040	0001	0006	WB8PMG	Dave	599	599	500/16

Save QTH Troy, OH St OH Pr Cnty U.S.A. Loc Az

ClrTx ClrRx Log 8 Tune margin 4 Int. period -14 Rx offset □ FEC ☑ 64bit

```
Vs7#_Db*Q:&#mb+eCnUUfyC@C

<00:05:11 - OLIVIA 500/16 - 3584.040>
WB8PMG WB8PMG de WB8ROL tnx for QSO Dave - ur RSQ is 599 599
QTH is Ludlow Falls, OH - near Dayton - grid loc is EM79
name is Gary Gary - BTU hw cpy Dave? - WB8PMG de WB8ROL KN
<00:06:46 - OLIVIA 500/16 - 3584.040>
```

OLIVIA 16/500 s/n -20.0 dB f/o -125.0 Hz

Figure 2. FLDigiROL v 3.10.r31 with Its Frequency Control Dialog, Button Bar and Other New Controls

ways, which can be considered submodes, all of which can be decoded only if you can identify the mode and its parameters before the station stops transmitting. For example, my favorite digital mode, Olivia, can be configured 40 different ways. Even an experienced digital operator often will find it difficult to identify these modes accurately and in a timely manner.

FEW OPERATORS EVER WOULD BUY A PIECE OF RADIO GEAR IF IT CAME WITH A LICENSE THAT SAID THEY COULD *NOT* MODIFY IT, AND IT'S NATURAL TO SEE WHY A LOT OF US NAVIGATE TOWARD OPEN SOURCE IN GENERAL AND LINUX IN PARTICULAR.

Several automatic features in Fldigi help operators determine a digital signal's mode when it is received, but they are not all very effective, especially under adverse band or weak signal conditions. Quite often, you just have to guess and change modes and configurations, as rapidly as possible, and hope you eventually guess the right one before the station quits transmitting. If you have to go through multiple dialog tabs and menu items to accomplish this, it will slow you down and greatly decrease your chance of success.

I decided to add three programmable button bars to the main screen of FLDigiROL. I made them similar to the macro button bar that Fldigi already had, so the existing code was helpful as a model. Each button could be set to a specific digital mode or submode, giving me a total of 27 buttons that could be visible all at the same time. This would hardly be enough buttons for all the possible modes available, but it easily could cover most of the commonly used ones. This would let me switch from one specific mode with specific parameters to another one with a single click of the mouse and greatly speed up the process (Figure 3).

I added an additional panel just below the contact (QSO) entry fields with several controls to clear the transmit (TX) quickly and receive (RX) text fields, access the log book, set important Olivia-MT63-DominoEX mode parameters on the fly, and a few other important things that demanded to be on the main screen.

I also added some more user feedback code in the form of both pop-up dialog boxes and information written to the title bar and status bar. I felt that Fldigi did not always let me know what was going on when I clicked some buttons and menu items. The QSO SAVE button, for example, did not give any audible or visual indication that anything occurred when you saved a contact. Often, I would forget if I had saved a contact and would have to check the log to make sure the information was there. I also added code so that if I tried to save the same contact more than once, the program would know it and notify me.

Another thing I really missed from the DM780 program was the auto log lookup feature, which let me know if I had talked to another Amateur Radio station before I entered the call into the CALL entry field. So, I spent more than a few hours writing and perfecting a logbook lookup routine for FLDigiROL that did all of that and also would display the date and frequency (band) information for the previous contact in the label control above the NOTE text field. Later, I added code so that I could simply put my mouse cursor over the call sign in the RX text control and see the same information before I even entered the call sign into the CALL entry field.

I made several cosmetic changes to the FLDigi internal contact log book and also altered the way it saved the contacts to disk. Fldigi did not save log entries directly to disk when I clicked the QSO SAVE button. I discovered that fact when Fldigi crashed on me once after finishing three consecutive contacts and finding out the hard way that it saved the contact information to disk only when the program was closed in a

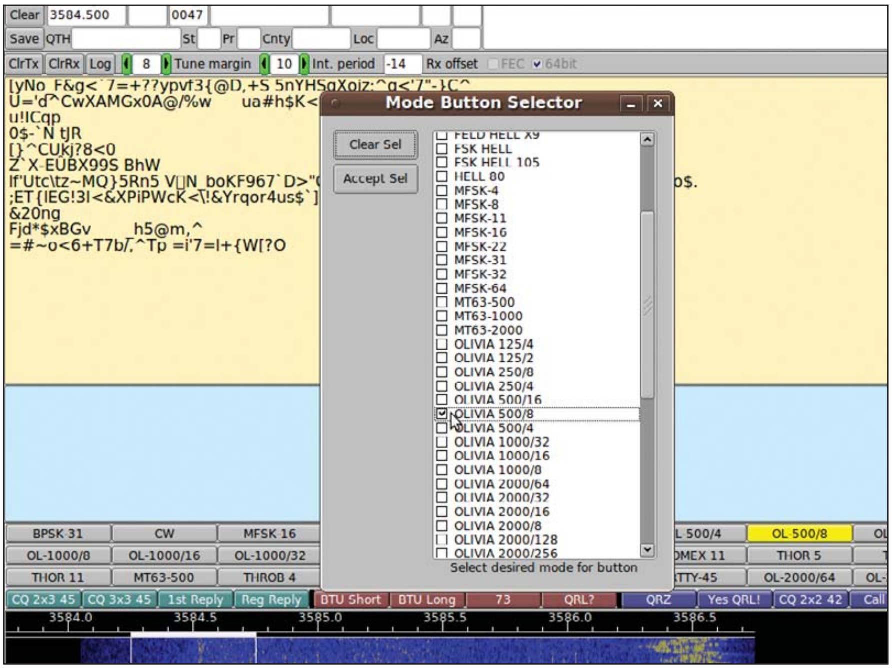


Figure 3. The Programmable Button Bar Dialog for FLDigiROL

normal fashion. I changed the code so the log book would save the information immediately to the hard disk file and also made the UPDATE button in the log book dialog do the same after I edited an existing entry.

I initially spent a couple months adding features, mods and playing with the code to get it closer to where I wanted it to be. Eventually, during the past year, I added a dozen more small enhancements and had a lot of fun in the process. It was an experience that made me respect the original authors even more than I already did. Because I also documented all my code additions and changes, I was able to download newer versions of Fldigi as they were released and apply my mods in just a matter of days. Some of the things I added to FLDigiROL have since been added to the original Fldigi program. I suspect that most of those items were going to be added anyway and were just waiting in the queue for a long time, but I would like to think my efforts also may have resulted directly or indirectly in a few of them being adopted.

I installed Cygwin, a Windows-based cross-compiler environment, in a Windows virtual machine and eventually managed to make a Windows version of FLDigiROL for a few operators who requested it. However, since the release of version 3.13 of Fldigi, the authors have started supporting the use of MinGW, another cross-platform compiler environment that can be hosted easily in Linux. I still use my virtual machine to test the compiled Windows version of FLDigiROL, but I now use MinGW on my main distro and compile it in Linux. MinGW seems to work much better than Cygwin did for me, and it was a lot easier to set up.

My experience with the Fldigi open-source code has inspired me to try another project. My next goal is to take the Fldigi code and make yet another version—a lite version of Fldigi for new or inexperienced digital operators. I plan to take out most of the configuration settings and simplify the program so that new operators can get it up and running within minutes and not have to do much more than enter call sign, name and home location (QTH). They still would need to select a soundcard and set up their interface COM or USB port(s), but they would not have to worry about a lot of advanced mode or configuration settings. I want to make it simple enough that it will not overwhelm new operators, so they easily can try some of the more advanced modes like Olivia, MT63, DominoEX and THOR. I plan to hard-code a lot of arbitrary default setting choices for them. When they get to the point where they start to feel more at ease with some of the advanced modes and want to have more control, they can move on to the regular release of Fldigi. I want Fldigi-Lite to be a little more like the older versions of the classic DigiPan program, except that it will have more digital modes than just PSK (phase shift keying—another mode).

The power of open-source software has made a big difference in my Amateur Radio and computing life. It's given me a measure of control over the software I use, and it has allowed me to do a lot more than just make suggestions or complain about commercial programs. It's given me the opportunity to make software personal, to learn a lot at the same time and to contribute not just ideas but actual code to the user community. I hope to see more of my fellow Amateur Radio developers (and those who would like to learn how to program) support the Open Source movement and continue to help Linux lead the way. It's part of what open source is all about—sharing and improving software, allowing choice and engendering freedom in general.■

Gary L. Robinson has been a ham operator for more than 46 years and a programmer since 1993. He is semi-retired and lives with his wife and a small herd of cats in the tiny village of Ludlow Falls, Ohio. Please feel free to send him comments at grobin1949@gmail.com, or catch him on the air for a digital chat.

Resources

Fldigi: **www.w1hkj.com/Fldigi.html**

Ham Radio Deluxe: **ham-radio-deluxe.com**

FLTK: **fltk.org**

KDevelop: **www.kdevelop.org**

FLDigiROL Source Code: **home.roadrunner.com/~rolswana**

Cygwin: **cygwin.com**

MinGW—Minimalist GNU for Windows: **mingw.org**

DigiPan Download Page: **digipan.net**

What’s New in Firewall Builder 3.0

Firewall Builder helps automate the process of configuring Linux, BSD and Cisco firewalls. Firewall Builder 3.0 comes with support for IPv6, branching rules and a lot more. It will save you time and help avoid errors. VADIM KURLAND

Mick Bauer introduced Firewall Builder to *Linux Journal* readers in 2003 with his article “Using Firewall Builder”. A lot of time has passed since 2003, the project has evolved, the appearance of the GUI has changed and many features have been added.

Open-source firewall implementations have gained momentum in recent years and now offer a viable alternative to commercial systems. These implementations include iptables (Netfilter) on Linux, PF on OpenBSD and FreeBSD, and ipfilter and ipfw on FreeBSD. These systems provide very respectable feature sets and good performance, but they provide only command-line access and plain-text configuration files. The syntax often is rather complex and different between the different tools, and definitely different from commercial firewalls. Administrators have to understand the internal structure and logic of the given firewall system in order to be able to design and maintain a configuration with the required level of confidence and reliability. For example, for iptables, you need to understand internal packet flow in the Netfilter engine to choose chains and targets correctly. Using the chain INPUT instead of FORWARD can mean the difference between a working service and a broken service. A simple error like this can cause the server behind the dedicated firewall to become inaccessible and at the same time create a hole allowing access to the firewall itself.

Complex configuration languages and the need for the administrator to be aware of the internal packet flow in the packet inspection engine make management of these firewalls difficult.

Many firewall appliances based on iptables or PF offer a Web-based GUI interface that helps configure them, but these interfaces tend to reflect the structure and ideas of the underlying configuration language closely.

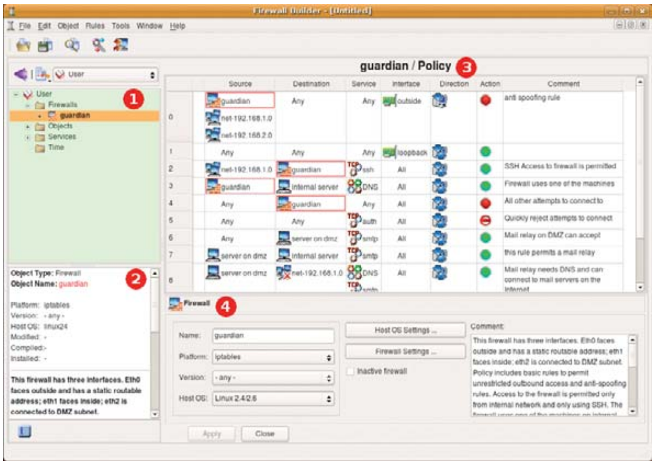
Complex configuration languages and the need for the administrator to be aware of the internal packet flow in the packet inspection engine make management of these firewalls difficult. This is especially so in multivendor installations where the administrator manages router access lists, several dedicated firewalls and perhaps local firewall rules on servers. Constant switching between different configuration languages leads to errors. Coordinating changes across multiple devices in the network becomes difficult and risky. When a new service is added, administrators need to implement the same policy rule using different languages for different devices. Administrators have to become experts in all platforms in order to be able to make the same change everywhere. An error leads either to service downtime or a hole in the security policy, and both can be very costly.

Modern IT organizations most often operate on a network built with products from different manufacturers. Even in installations that standardize on one kind of OS for servers, there is usually the need to configure access lists on routers and rules on dedicated firewalls, which often are products of a different vendor.

One of the ways to help the situation is to switch to automated generation of the configuration. Several commercial systems and open-source projects try to tackle this problem. Automatic generation of the configuration has many advantages over manual processes. Automated systems treat configuration as a combination of standardized blocks, or objects, that present information in an abstract way. When administrators work with such an abstracted system, they do not need to switch between the configuration paradigms of different vendors in order to implement the same rule; they just work with the same abstract firewall all the time. Firewall Builder implements this approach to the generation of firewall configuration.

Firewall Builder is a GUI firewall configuration and management tool. It helps administrators design and build complex firewall rules for several open-source and commercial hardware and software platforms. With Firewall Builder, administrators can create configurations for iptables, PF, ipfilter, ipfw, Cisco IOS extended access lists and Cisco ASA (PIX). Administrators can configure policies for servers, dedicated firewalls and routers, all from the same GUI running on a workstation or laptop. Once all rules have been created or modified, administrators can use the same program to update the configuration of all devices using SSH for secure connections.

Firewall Builder is an open-source project hosted on


Firewall Builder - [Untitled]
File Edit Object Rules Tools Window Help
User
User
Firewalls
guardian
Objects
Services
Time
1
2
3
4

guardian / Policy

	Source	Destination	Service	Interface	Direction	Action	Comment
0	guardian net-192.168.1.0 net-192.168.2.0	Any	Any	outside		●	anti spoofing rule
1	Any	Any	Any	loopback		●	
2	net-192.168.1.0	guardian	ssh	All		●	SSH Access to firewall is permitted
3	guardian	internal server	DNS	All		●	Firewall uses one of the machines
4	Any	guardian	Any	All		●	All other attempts to connect to
5	Any	Any	auth	All		⊖	Quickly reject attempts to connect
6	Any	server on dmz	smtp	All		●	Mail relay on DMZ can accept
7	server on dmz	internal server	smtp	All		●	this rule permits a mail relay
8	server on dmz	net-192.168.1.0	DNS smtp	All		●	Mail relay needs DNS and can connect to mail servers on the internet

Object Type: Firewall
Object Name: guardian

Platform: iptables
Version: - any -
Host OS: linux24
Modified: -
Compiled: -
Installed: -

This firewall has three interfaces. Eth0 faces outside and has a static routable address; eth1 faces inside; eth2 is connected to DMZ subnet.

Firewall

Name: guardian
Platform: iptables
Version: - any -
Host OS: Linux 2.4/2.6

Host OS Settings ...
Firewall Settings ...
□ Inactive firewall

Comment:
This firewall has three interfaces. Eth0 faces outside and has a static routable address; eth1 faces inside; eth2 is connected to DMZ subnet. Policy includes basic rules to permit unrestricted outbound access and anti-spoofing rules. Access to the firewall is permitted only from internal network and only using SSH. The

Apply Close

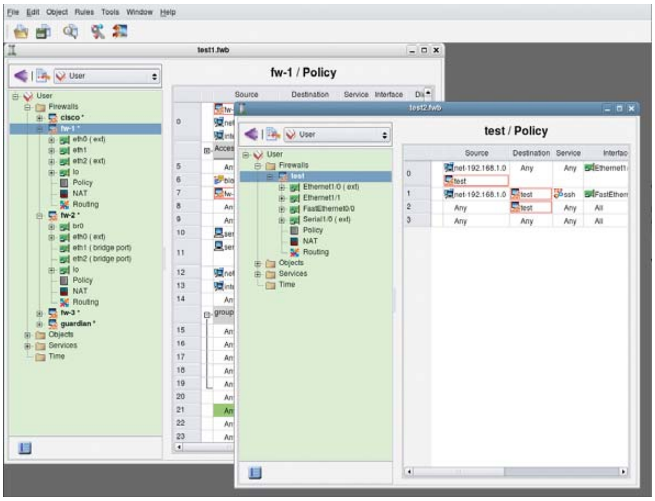
Figure 1. Main GUI Elements

SourceForge at **https://sourceforge.net/projects/fwbuilder**. It comes with most major Linux distributions, such as Debian, Ubuntu, Fedora, OpenSUSE and Mandriva. You can install it using your favorite package manager—just look for the package fwbuilder and its dependencies. Because these distributions work on their own schedules, which are not coordinated with Firewall Builder releases, the packages they offer usually are one or two revisions behind the latest released package. Because of this, we offer our own repositories of deb and rpm fwbuilder packages. Packages in these repositories are signed with our GPG key for verification. We offer two repositories: stable and testing. The stable repository provides the latest stable release of Firewall Builder, and the testing repository is used to distribute packages to the beta testers while we work on the next version. Instructions for how to set up your system to use our repositories are available on our Web site (**www.fwbuilder.org/docs/firewall_builder_packages.html**).

Once the program has been installed, you can launch it from a command line using its name fwbuilder or via a main GNOME or KDE menu under System→Administration.

Firewall Builder is a rather old project; it has been around since 2001. During this time, the program has evolved into a mature, production-ready tool used by hundreds of system administrators every day. The most recent version, 3.0, was released in fall 2008, and since then, it has seen a few maintenance releases that provided bug fixes and minor improvements. This article focuses on the features that were introduced in Firewall Builder 3.0.

The main window of the program (Figure 1) includes an objects tree on the left (1), brief information about the object selected in the tree (2), current firewall policy view (3) and a dialog panel where administrators can edit object parameters (4). All servers, firewalls, subnets, individual addresses and services are represented by objects in the tree. Each object has attributes that describe its parameters; the combination of these attributes is different depending on the object type. Objects are arranged in the tree so that each object type has its own folder or branch.



test / Policy

	Source	Destination	Service	Interfac
0	net-192.168.1.0 test	Any	Any	Ethernet1
1	net-192.168.1.0	test	ssh	FastEthen
2	Any	test	Any	All
3	Any	Any	Any	All

Figure 2. Main Window with Two Data Files Opened at the Same Time

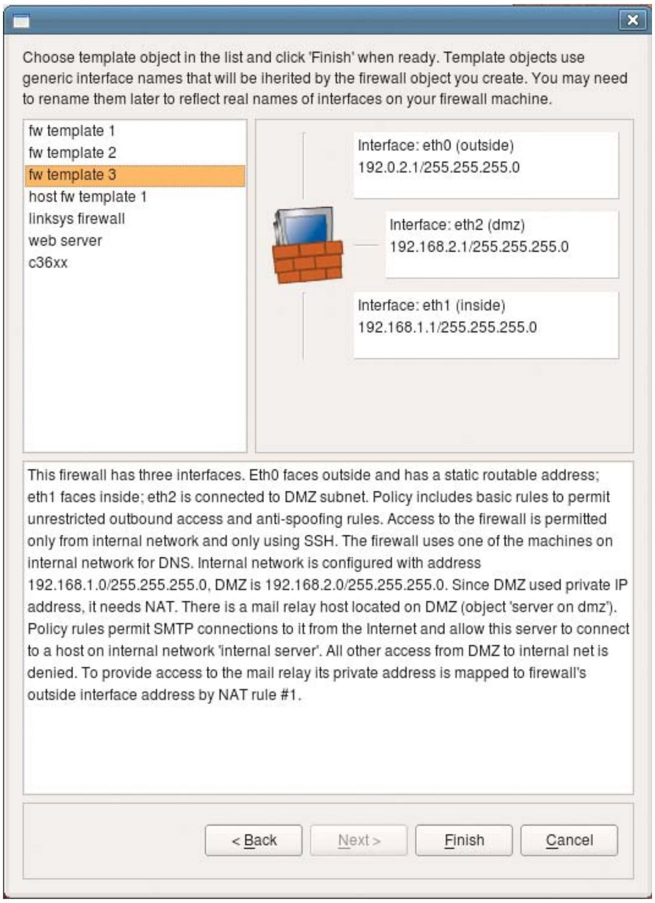
Expanding on this concept, Firewall Builder 3.0 allows you to open multiple data files at once (Figure 2) and move objects between them using drag-and-drop or copy/paste operations.

You can expand one data file so that it occupies the whole window, or you can arrange several data files so they are all visible at the same time.

Once objects are created to represent a network, you can create a firewall object and start building security policy rules. Firewall Builder 3.0 comes with a library of templates representing some typical firewall configurations. The simplest way to get started is to create a new firewall from one of these templates and then make changes to reflect your network's addresses and configuration.

Administrators can configure policies for servers, dedicated firewalls and routers, all from the same GUI running on a workstation or laptop.

Figure 3 shows the dialog where you can choose which template to use to create a new firewall object. This view shows the creation of a new firewall object with three interfaces that will have the IP addresses shown in the dialog. Most likely, your network uses a different addressing schema, and these addresses don't match what you have.



Choose template object in the list and click 'Finish' when ready. Template objects use generic interface names that will be iherited by the firewall object you create. You may need to rename them later to reflect real names of interfaces on your firewall machine.

- fw template 1
- fw template 2
- fw template 3
- host fw template 1
- linksys firewall
- web server
- c36xx

Interface: eth0 (outside)
192.0.2.1/255.255.255.0

Interface: eth2 (dmz)
192.168.2.1/255.255.255.0

Interface: eth1 (inside)
192.168.1.1/255.255.255.0

This firewall has three interfaces. Eth0 faces outside and has a static routable address; eth1 faces inside; eth2 is connected to DMZ subnet. Policy includes basic rules to permit unrestricted outbound access and anti-spoofing rules. Access to the firewall is permitted only from internal network and only using SSH. The firewall uses one of the machines on internal network for DNS. Internal network is configured with address 192.168.1.0/255.255.255.0, DMZ is 192.168.2.0/255.255.255.0. Since DMZ used private IP address, it needs NAT. There is a mail relay host located on DMZ (object 'server on dmz'). Policy rules permit SMTP connections to it from the Internet and allow this server to connect to a host on internal network 'internal server'. All other access from DMZ to internal net is denied. To provide access to the mail relay its private address is mapped to firewall's outside interface address by NAT rule #1.

Figure 3. Creating Firewall Object Using a Template

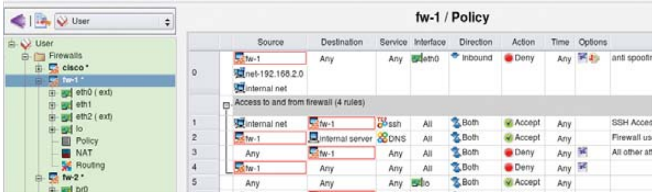
You can use the "search and replace" function in the GUI to replace them. Other templates that come with the program represent a firewall with one or two interfaces, a Cisco router, a server and a small residential firewall based on Linksys hardware. Firewall objects created from templates come not only with all the interfaces already configured, but they also come with a small set of rules to implement a basic security policy. Of course, you should edit these rules to adapt the configuration to your security policy requirements. The rules created by the templates serve as an illustration of how practical rules can be built in Firewall Builder and as a starting point for your own policy design.

Template objects provide a quick way to create new firewall configurations, but they are limited in number and variety. If no template is close to your configuration, you can create a firewall object and all its interfaces manually. In addition to that, if your firewall machine runs the SNMP dæmon, you can have Firewall Builder "discover" its configuration automatically.

Now that the firewall object and objects for its interfaces have been created, you can add some rules. Figure 4 shows a fragment of a firewall policy.

As in the previous versions of Firewall Builder, policy rules have fairly typical columns: source, destination, service, interface, direction, action, time, options and comment. An address object placed in the source rule element means the packet's source address must match it in order to match the whole rule. The same goes for the destination. A service object defines ports for TCP and UDP protocols or type and code for ICMP messages. When a service object appears in the service rule element, the packet must match attributes of the object in order to match the rule. You can place objects in rules by dragging them from the tree or by using a copy/paste operation.

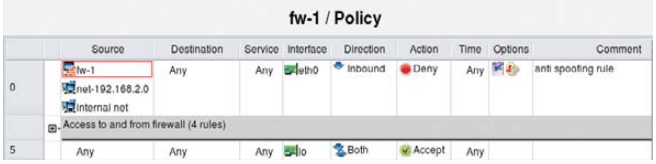
Firewall Builder 3.0 adds support for rule grouping.



fw-1 / Policy

	Source	Destination	Service	Interface	Direction	Action	Time	Options	
0	fw-1 net-192.168.2.0 internal net	Any	Any	eth0	Inbound	Deny	Any		anti spoofi
	Access to and from firewall (4 rules)								
1	internal net	fw-1	ssh	All	Both	Accept	Any		SSH Acces
2	fw-1	internal server	DNS	All	Both	Accept	Any		Firewall us
3	Any	fw-1	Any	All	Both	Deny	Any		All other at
4	fw-1	Any	Any	All	Both	Deny	Any		
5	Any	Any	Any	lo	Both	Accept	Any		

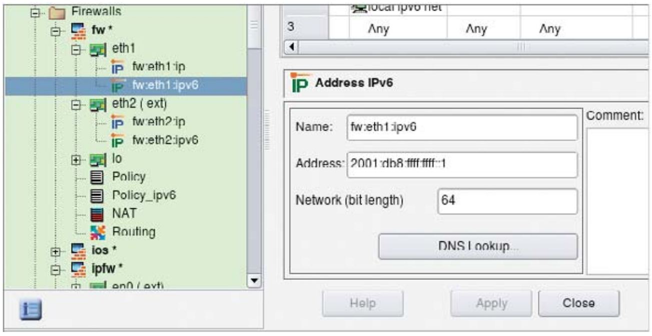
Figure 4. Example of Policy Rules



fw-1 / Policy

	Source	Destination	Service	Interface	Direction	Action	Time	Options	Comment
0	fw-1 net-192.168.2.0 internal net	Any	Any	eth0	Inbound	Deny	Any		anti spoofing rule
	Access to and from firewall (4 rules)								
5	Any	Any	Any	lo	Both	Accept	Any		

Figure 5. Example of a Collapsed Group of Rules

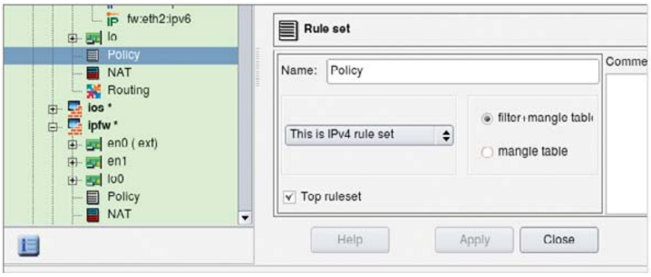


Name: fw:eth1:ipv6

Address: 2001:db8:ffff:ffff::1

Network (bit length) 64

Figure 6. Interface eth1 has an IPv6 address.



Name: Policy

This is IPv4 rule set

● filter+mangle tabl

○ mangle table

☑ Top ruleset

Figure 7. Example of a Policy Object Configured as IPv4-Only

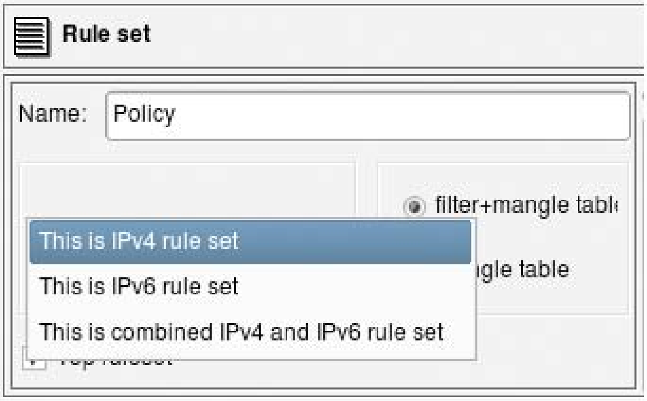
Figure 4 shows a group of rules with the title “Access to and from firewall” that consists of four rules. These rules form a group that can be collapsed so that only its title is visible (Figure 5).

You can greatly improve the readability of the whole rule set if you arrange rules in groups. This helps maintain a good visual reference to the internal logic when several rules work together to implement some part of the security policy. You also can collapse groups you are not working with at the moment to make more rules fit on the screen at once.

Firewall Builder 3.0 comes with a library of templates representing some typical firewall configurations.

With IPv4 address space depletion on the horizon, IPv6 roll-out is starting to pick up speed. Firewall Builder 3.0 comes with support for IPv6. The program provides several new types of address and service objects that represent IPv6 addresses, networks and services. An example of an interface with an IPv6 address is shown in Figure 6.

In Firewall Builder 3.0, an interface object can have several different types of child objects: a MAC address, an IPv4 address and an IPv6 address. Policy and NAT rule set objects now have an attribute that defines which address family they should match. Figure 7 shows an example of an IPv4-only policy.



Name: Policy

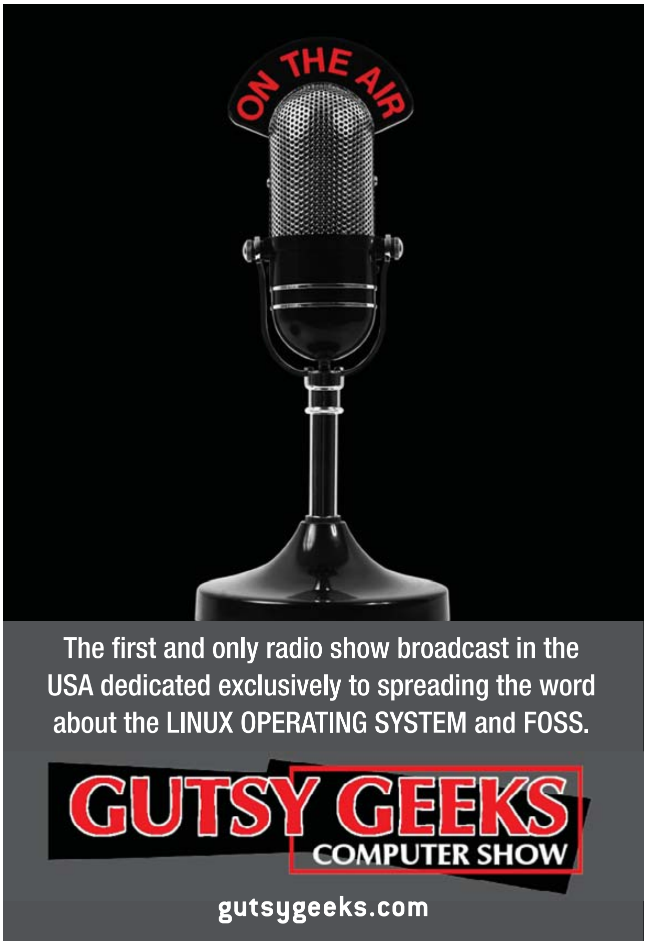
● filter+mangle tabl

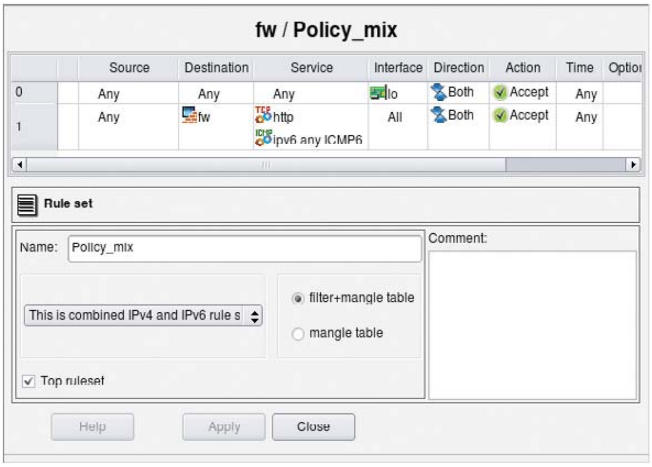
This is IPv4 rule set
This is IPv6 rule set
This is combined IPv4 and IPv6 rule set

Figure 8. Different Address Family Configuration Options for a Policy or NAT Object

The drop-down list of address family options is shown in Figure 8. The address family options are:

- “IPv4-only rule set”: only addressable objects with an IPv4 address will be used in the rules. If an object with an IPv6





fw / Policy_mix

		Source	Destination	Service	Interface	Direction	Action	Time	Optio
0		Any	Any	Any	lo	Both	Accept	Any	
1		Any	fw	http ipv6 any ICMP6	All	Both	Accept	Any	

Rule set

Name: Policy_mix

Comment:

This is combined IPv4 and IPv6 rule s

● filter+mangle table

○ mangle table

☑ Top ruleset

Figure 9. Example of Mixed IPv4+IPv6 Rules

address appears in rules, it is ignored. IPv6-only services, such as ICMPv6, also are ignored. TCP and UDP services are used, as they apply for both IPv4 and IPv6 rules.

- “IPv6-only rule set”: only objects with IPv6 addresses are used, and those with IPv4 addresses are ignored. IPv6-only services, such as ICMPv6, are used, but IPv4-only services, such as ICMP, are ignored. TCP and UDP services are used, as they apply to both IPv4 and IPv6 rules.

- “Mixed IPv4- and IPv6-only rule set”: the compiler makes two passes over the same rules, first to produce an IPv4 configuration and then to produce an IPv6 configuration. On each pass it uses only address objects with addresses matching the address family of the pass. This is the best configuration for transitional configurations when IPv6 rules are gradually added to existing IPv4 configuration. Note that if you add IPv6 addresses to an interface of a firewall or a host object used in the rules, the compiler will use IPv4 addresses of the interface on the IPv4 pass and the new IPv6 address of the same interface on the IPv6 pass. This principle also applies to the mixed groups of addresses and services.

Another important new feature in Firewall Builder 3.0 is support for branching rules and multiple rule sets.

Let’s see how the program converts some simple rules in the combined IPv4+IPv6 rule set into an iptables script. The rules are shown in Figure 9.

The first rule uses the interface object of the firewall that represents its loopback interface, and the second uses the object representing the firewall itself and two service objects: “http” and “ipv6 any ICMP6”. The rule that permits any protocols on loopback should be similar for IPv4 and IPv6, but the rule that permits http and any ICMP6 will look different. Here is what the generated iptables script looks like:

```
# ================ IPv4
#
# Rule Policy_mix 0 (lo)
#
iptables -A INPUT  -i lo -m state --state NEW -j ACCEPT
iptables -A OUTPUT  -o lo -m state --state NEW -j ACCEPT
#
# Rule Policy_mix 1 (global)
#
iptables -A INPUT -p tcp -m tcp --dport 80 \
    -m state --state NEW -j ACCEPT

# ================ IPv6
#
# Rule Policy_mix 0 (lo)
#
ip6tables -A INPUT  -i lo -m state --state NEW -j ACCEPT
ip6tables -A OUTPUT  -o lo -m state --state NEW -j ACCEPT
#
# Rule Policy_mix 1 (global)
#
ip6tables -A INPUT -p tcp -m tcp --dport 80 \
    -m state --state NEW -j ACCEPT
ip6tables -A INPUT -p ipv6-icmp -m state \
   --state NEW -j ACCEPT
```

IPv4 rules are loaded using the iptables command-line utility, while IPv6 rules are loaded using ip6tables. The rule that permits anything on loopback produces very similar iptables commands. The difference is only in the choice of the command-line tool used to add them. The rule that permits http and any ICMP6, when compiled for IPv4, generates only an iptables command to match TCP port 80. This is because ICMP6 service makes sense only in the IPv6 context, so it is dropped from the rule when it is compiled for IPv4. When this rule is compiled for IPv6 though, both services are used and we get two iptables commands, one for TCP port 80 and another for protocol ipv6-icmp.

Mixed IPv4/IPv6 rule sets can be especially useful when IPv6 is added to an existing IPv4 network. Because most firewalls and routers require different syntax for IPv6 ACL and rules, you have to implement a second rule set for IPv6, carefully trying to copy existing IPv4 rules to preserve the general structure and meaning of the security policy. Things become even more complicated after that, because every change in the policy should now be reflected in two sets of ACL or firewall rules. Keeping these synchronized quickly can turn into a major task that can increase the probability of human error and network outages significantly.

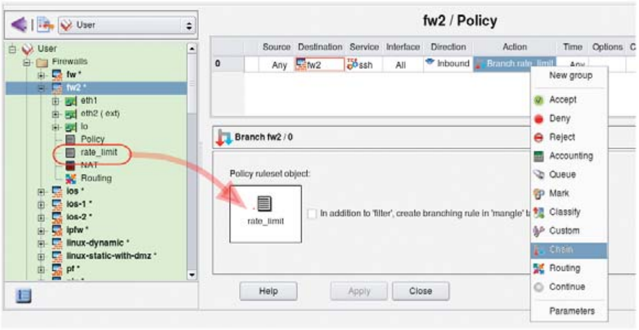


Figure 10. How to Create a Branch in the Rule Set

Mixed IPv4+IPv6 rule sets in Firewall Builder help solve this problem, because two iptables scripts are generated automatically from the same set of rules.

Another important new feature in Firewall Builder 3.0 is support for branching rules and multiple rule sets. Firewall objects now can have several sets of policy rules; each set is represented by a Policy object in the tree. Figure 10 shows firewall object fw2 that has two Policy rule sets that appear as objects in the tree: one has the name Policy, and the other has the name rate_limit. When the firewall object platform is set to iptables, rules in each of these additional Policy rule sets are placed in a chain with a name that is the same as the name of the rule set object. If the target firewall platform is set to PF, rules are placed in an anchor with the same name.

To create a branch in your rules, choose the Chain rule action, and drag another policy object into its editor panel, as shown in the screenshot.

This configuration translates into an iptables command with the target option `-j rate_limit` that passes control to that chain.

Multiple rules can use the Chain action to pass control to the same branch rule set if necessary. Abstraction of branching rules in Firewall Builder opens access to the very powerful mechanism of chains and anchors in the underlying firewall software. This can be used for rule optimization or to build dynamic policies where a branch rule set is populated by an external script at runtime.

Firewall Builder 3.0 comes with a large Users Guide in PDF and HTML formats, full of examples and step-by-step instructions. The Guide includes the Firewall Builder Cookbook that describes many typical problems and solutions.

As you can see, Firewall Builder 3.0 comes with many new features and improvements that make the upgrade worth your while. The main driver in the development is bug reports and feature requests that come from users, so please visit our Web site at **www.fwbuilder.org**, and help us make Firewall Builder even better.■

In addition to being the author of Firewall Builder starting in 1999, Vadim Kurland is a longtime network engineer for various Silicon Valley companies. Before that, he worked in software development and ISP operations.

Advertiser Index

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

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

Advertiser	Page #	Advertiser	Page #
1&1 Internet, Inc. www.oneandone.com	C2	Lullabot www.lullabot.com	45
Aberdeen, LLC www.aberdeeninc.com	5	Microway, Inc. www.microway.com	C4, 35
Archie McPhee www.mcphee.com	79	Mikro Tik www.routerboard.com	3
ASA Computers, Inc. www.asacomputers.com	25	Natuba Natuba.com	78
Cari.net www.cari.net	43	Polywell Computers, Inc. www.polywell.com	13, 79
Cloud Connect www.cloudconnectevent.com	69	PyCon us.pycon.org	49
Devicesoft Organization, LLC www.devicesoft.org	79	Saint Arnold Brewing Company www.saintarnold.com	78
Digi-Key Corporation www.digi-key.com	78	Scale www.socallinuxexpo.org	7
Emac, Inc. www.emacinc.com	61	SharePoint TechCon www.sptechcon.com	71
EmperorLinux www.emperorlinux.com	21	Silicon Mechanics www.siliconmechanics.com	19, 23
Gecad Technologies/Axigen www.axigen.com	78	Technologic Systems www.embeddedx86.com	27
Genstor Systems, Inc. www.genstor.com	39	Trusted Computer Solutions www.TrustedCS.com/SecurityBlanket	1, 79
Gutsy Geeks www.gutsygeeks.com	63	Usenix Association www.usenix.com/events	9
iXsystems, Inc. www.ixsystems.com	C3	Utilikilts www.utilikilts.com	79
Logic Supply, Inc. www.logicsupply.com	29		

ATTENTION ADVERTISERS

April 2010 Issue #192 Deadlines

Space Close: January 25; **Material Close:** February 2

Theme: Software Development

BONUS DISTRIBUTIONS: FOSE, SxSW

Call **Joseph Krack** to reserve your space
+1-713-344-1956 ext. 118, e-mail joseph@linuxjournal.com

Implement Port-Knocking Security with knockd

Instead of closing ports, make them disappear.

FEDERICO KEREKI

When dealing with computer security, you should assume that hackers will be trying to get in through any available doors your system may have, no matter how many precautions you might have taken. The method of allowing entrance depending on a password is a classic one and is widely used. In order to "open a door" (meaning, connect to a port on your computer), you first have to specify the correct password. This can work (provided the password is tough enough to crack, and you don't fall prey to many hacking attacks that might reveal your password), but it still presents a problem. The mere fact of knowing a door exists is enough to tempt would-be intruders.

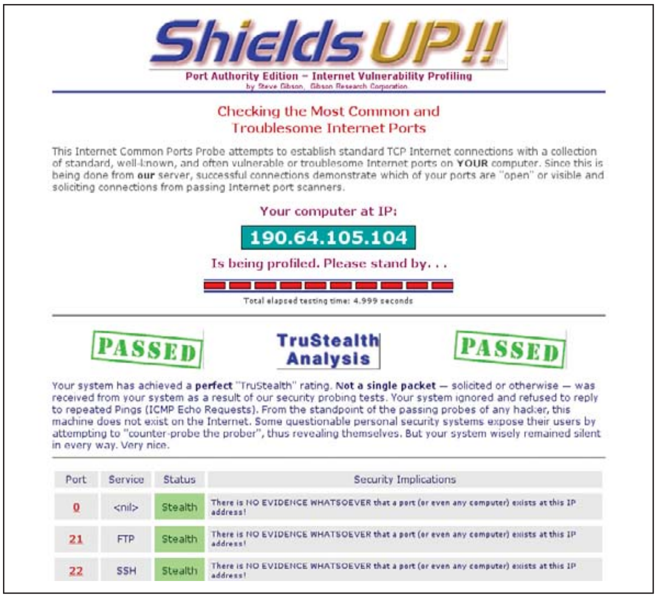
So, an open port can be thought of as a door with (possibly) a lock, where the password works as the key. If you are running some kind of public service (for example, a Web server), it's pretty obvious that you can't go overboard with protection; otherwise, no one will be able to use your service. However, if you want to allow access only to a few people, you can hide the fact that there actually is a door to the system from the rest of the world. You can "knock intruders away", by not only putting a lock on the door, but also by hiding the lock itself! Port knocking is a simple method for protecting your ports, keeping them closed and invisible to the world until users provide a secret knock, which will then (and only then) open the port so they can enter the password and gain entrance.

Port knocking is appropriate for users who require access to servers that are not publicly available. The server can keep all its ports closed, but open them on demand as soon as users have authenticated themselves by providing a specific knock sequence (a sort of password). After the port is opened, usual security mechanisms (passwords, certificates and so on) apply. This is an extra advantage; the protected services won't require any modification, because the extra security is provided at the firewall level. Finally, port knocking is easy to implement and quite modest as far as resources, so it won't cause any overloads on the server.

In this article, I explain how to implement port knocking in order to add yet another layer to your system security.

Are You Safe?

Would-be hackers cannot attack your system unless they know which port to try. Plenty of port-scanning tools are available. A simple way to check your machine's security level is by running an on-line test, such as GRC's ShieldsUp (Figure 1). The test results in Figure 1 show that attackers wouldn't even know a machine is available to attack, because all the port queries were ignored and went unanswered.



Shields UP!!
Port Authority Edition – Internet Vulnerability Profiling
by Steve Gibson, Gibson Research Corporation

Checking the Most Common and Troublesome Internet Ports

This Internet Common Ports Probe attempts to establish standard TCP Internet connections with a collection of standard, well-known, and often vulnerable or troublesome Internet ports on **YOUR** computer. Since this is being done from **our** server, successful connections demonstrate which of your ports are "open" or visible and soliciting connections from passing Internet port scanners.

Your computer at IP:

190.64.105.104

Is being profiled. Please stand by. . .

Total elapsed testing time: 4.999 seconds

Your system has achieved a **perfect** "TruStealth" rating. **Not a single packet** — solicited or otherwise — was received from your system as a result of our security probing tests. Your system ignored and refused to reply to repeated Pings (ICMP Echo Requests). From the standpoint of the passing probes of any hacker, this machine does not exist on the Internet. Some questionable personal security systems expose their users by attempting to "counter-probe the prober", thus revealing themselves. But your system wisely remained silent in every way. Very nice.

Port	Service	Status	Security Implications
0	<nil>	Stealth	There is NO EVIDENCE WHATSOEVER that a port (or even any computer) exists at this IP address!
21	FTP	Stealth	There is NO EVIDENCE WHATSOEVER that a port (or even any computer) exists at this IP address!
22	SSH	Stealth	There is NO EVIDENCE WHATSOEVER that a port (or even any computer) exists at this IP address!

Figure 1. A completely locked-up site, in "stealth" mode, doesn't give any information to attackers, who couldn't even learn that the site actually exists.

Another common tool is nmap, which is a veritable Swiss Army knife of scanning and inspection options. A simple `nmap -v your.site.url` command will try to find any open ports. Note that by default, nmap checks only the 1–1000 range, which comprises all the "usual" ports, but you could do a more thorough test by adding a `-p1-65535` parameter. Listing 1 shows how you can rest assured that your site is closed to the world. So, now that you know you are safe, how do you go about opening a port, but keep it obscured from view?

Secret Handshakes, Taps and Knocks

The idea behind port knocking is to close all ports and monitor attempts to connect to them. Whenever a very specific sequence of attempts (a knock sequence) is recognized, and only in that case, the system can be configured to perform some specific action, like opening a given port, so the outsider can get in. The knock sequence can be as complex as you like—for example, a simple list (like trying TCP port 7005, then

Listing 1. The standard nmap port-scanning tool provides another confirmation that your site and all ports are completely closed to the world.

```
$ nmap -v -A your.site.url

Starting Nmap 4.75 ( http://nmap.org ) at 2009-10-03 12:59 UYT
Initiating Ping Scan at 12:59
Scanning 190.64.105.104 [1 port]
Completed Ping Scan at 12:59, 0.00s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 12:59
Completed Parallel DNS resolution of 1 host. at 12:59, 0.01s elapsed
Initiating Connect Scan at 12:59
Scanning r190-64-105-104.dialup.adsl.anteldata.net.uy (190.64.105.104)
 [1000 ports]
Completed Connect Scan at 12:59, 2.76s elapsed (1000 total ports)
Initiating Service scan at 12:59
SCRIPT ENGINE: Initiating script scanning.
Host r190-64-105-104.dialup.adsl.anteldata.net.uy (190.64.105.104)
 appears to be up ... good.
All 1000 scanned ports on r190-64-105-104.dialup.adsl.anteldata.net.uy
 (190.64.105.104) are closed

Read data files from: /usr/share/nmap
Service detection performed. Please report any incorrect results
 at http://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 2.94 seconds
```

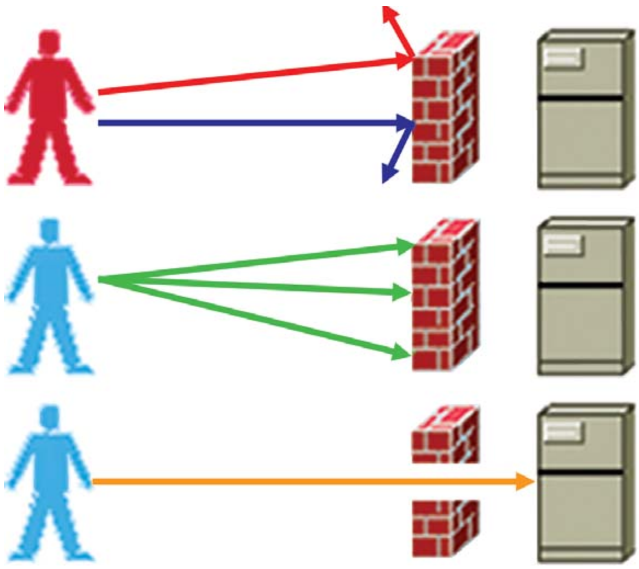


Figure 2. Would-be attackers (top) are simply rejected by the firewall, but when a legit user (middle) provides the correct sequence of "knocks", the firewall (bottom) allows access to a specific port, so the user can work with the server.

Listing 2. A simple /etc/knockd.conf file, which requires successive knocks on ports 7005, 7007 and 7006 in order to enable secure shell (SSH) access.

```
[opencloseSSH]
  sequence      = 7005,7006,7007
  seq_timeout   = 15
  tcpflags      = syn
  start_command = /usr/sbin/iptables -s %IP% -I input_ext 1
                ➥-p tcp --dport ssh -j ACCEPT
  cmd_timeout   = 30
  stop_command  = /usr/sbin/iptables -s %IP% -D input_ext
                ➥-p tcp --dport ssh -j ACCEPT
```

TCP port 7006 and finally, TCP port 7007) to a collection of use-only-once sequences, which once used, will not be allowed again. This is the equivalent of "one-time pads", a cryptography method that, when used correctly, provides perfect secrecy.

Before setting this up, let me explain why it's a good safety measure. There are 65,535 possible ports, but after discarding the already-used ones (see the list of assigned ports in Resources), suppose you are left with "only" 50,000 available ports. If attackers have to guess a sequence of five different ports, there are roughly 312,000,000,000,000,000,000,000 possible combinations they should try. Obviously, brute-force methods won't help! Of course, you shouldn't assume that blind luck is the only possible attack, and that's why port knocking ought not be the only security measure you use, but just another extra layer for attackers to go through (Figure 2).

On the machine you are protecting, install the knockd dæmon, which will be in charge of monitoring the knock attempts. This package is available for all distributions. For example, in Ubuntu, run `sudo apt-get install knockd`, and in OpenSUSE, run `sudo zypper install knockd` or use YaST. Now you need to specify your knocking rules by editing the /etc/knockd.conf file and start the dæmon running. An example configuration is shown in Listing 2. Note: the given iptables commands are appropriate for an OpenSUSE distribution running the standard firewall, with eth0 in the external zone; with other distributions and setups, you will need to determine what command to use.

You probably can surmise that this looks for a sequence of three knocks—7005, 7006 and 7007 (not very safe, but just an example)—and then opens or closes the SSH port. This example allows a maximum timeout for entering the knock sequence (15 seconds) and a login window (30 seconds) during which the port will be opened. Now, let's test it out.

First, you can see that without running knockd, an attempt to log in from the remote machine just fails:

```
$ ssh your.site.url -o ConnectTimeout=10
ssh: connect to host your.site.url port 22: Connection timed out
```

Next, let's start the knockd server. Usually, you would run it as root via `knockd -d` or `/etc/init.d/knockd start`; however, for the moment, so you can see what happens, let's run it in debug mode with `knock -D`:

```
# knockd -D
config: new section: 'opencloseSSH'
config: opencloseSSH: sequence: 7005:tcp,7006:tcp,7007:tcp
config: opencloseSSH: seq_timeout: 15
config: tcp flag: SYN
config: opencloseSSH: start_command:
           /usr/sbin/iptables -s %IP% -I input_ext 1
                              -p tcp --dport ssh -j ACCEPT
config: opencloseSSH: cmd_timeout: 30
config: opencloseSSH: stop_command:
           /usr/sbin/iptables -s %IP% -D input_ext
                              -p tcp --dport ssh -j ACCEPT
ethernet interface detected
Local IP: 192.168.1.10
```

Now, let's go back to the remote machine. You can see that an ssh attempt still fails, but after three knock commands, you can get in:

```
$ ssh your.site.url -o ConnectTimeout=10
ssh: connect to host your.site.url port 22: Connection timed out
$ knock your.site.url 7005
$ knock your.site.url 7006
$ knock your.site.url 7007
$ ssh your.site.url -o ConnectTimeout=10
Password:
Last login: Sat Oct  3 14:58:45 2009 from 192.168.1.100
...
```

Looking at the console on the server, you can see the knocks coming in:

```
2009-09-03 15:29:47:
     tcp: 190.64.105.104:33036 -> 192.168.1.10:7005 74 bytes
2009-09-03 15:29:50:
     tcp: 190.64.105.104:53783 -> 192.168.1.10:7006 74 bytes
2009-09-03 15:29:51:
     tcp: 190.64.105.104:40300 -> 192.168.1.10:7007 74 bytes
```

If the remote sequence of knocks had been wrong, there would have been no visible results and the SSH port would have remained closed, with no one the wiser.

Configuring and Running knockd

The config file /etc/knockd.conf is divided into sections, one for each specific knock sequence, with a special general section, options, for global parameters. Let's go through the general options first:

- You can log events either to a specific file by using `LogFile=/path/to/log/file`, or to the standard Linux log files by using `UseSyslog`. Note: it's sometimes suggested that you avoid such logging, because it enables an extra possible goal for attackers—should they get their hands on the log, they would have the port-knocking sequences.

- When knockd runs as a dæmon, you may want to check whether it's still running. The `PidFile=/path/to/PID/file` option specifies a file into which knockd's PID (process ID) will be stored. An interesting point: should knockd crash, your system will be safer than ever—all ports will be closed (so safer) but totally unaccessible. You might consider implementing a cron task that would check for the knockd PID periodically and restart the dæmon if needed.

- By default, eth0 will be the observed network interface. You can change this with something like `Interface=eth1`. You must not include the complete path to the device, just its name.

Every sequence you want to recognize needs a name; the example (Listing 2) used just one, named openclosessh. Options and their parameters can be written in upper-, lower-

If You Are behind a Router

If you aren't directly connected to the Internet, but go through a router instead, you need to make some configuration changes. How you make these changes depends on your specific router and the firewall software you use, but in general terms you should do the following:

1) Forward the knock ports to your machine, so knockd will be able to recognize them.

2) Forward port 22 to your machine. Although in fact, you could forward any other port (say, 22960) to port 22 on your machine, and remote users would have to `ssh -p 22960 your.site.url` in order to connect to your machine. This could be seen as "security through obscurity"—a defense against script kiddies, at least.

3) Configure your machine's firewall to *reject* connections to port 22 and to the knock ports:

```
$ /usr/sbin/iptables -I INPUT 1 -p tcp --dport ssh -j REJECT
$ /usr/sbin/iptables -I INPUT 1 -p tcp --sport 7005:7007 -j REJECT
```

The command to allow SSH connections would then be:

```
$ /usr/sbin/iptables -I INPUT 1 -p tcp --dport ssh -j ACCEPT
```

And, the command for closing it again would be:

```
$ /usr/sbin/iptables -D INPUT -p tcp --dport ssh -j ACCEPT
```

or mixed case:

- `Sequence` is used to specify the desired series of knocks—for example, `7005,7007:udp,7003`. Knocks are usually TCP, but you can opt for UDP.

- `One_Time_Sequences=/path/to/file` allows you to specify a file containing "use once" sequences. After each sequence is used, it will be erased. You just need a text file with a sequence (in the format above) in each line.

- `Seq_Timeout=seconds.to.wait.for.the.knock` is the maximum time for completing a sequence. If you take too long to knock, you won't be able to get in.

- `Start_Command=some.command` specifies what command (either a single line or a full script) must be executed after recognizing a knock sequence. If you include the %IP% parameter, it will be replaced at runtime by the knocker's IP. This allows you, for example, to open the firewall port but only for the knocker and not for anybody else. This example uses an iptables command to open the port (see Resources for more on this).

- `Cmd_Timeout=seconds.to.wait.after.the.knock` lets you execute a second command a certain time after the start command is run. You can use this to close the port; if the knocker didn't log in quickly enough, the port will be closed.

- `Stop_Command=some.other.command` is the command that will be executed after the second timeout.

- `TCPFlags=list.of.flags` lets you examine incoming TCP packets and discard those that don't match the flags (FIN, SYN, RST, PSH, ACK or URG; see Resources for more on this). Over an SSH connection, you should use `TCPFlags=SYN`, so other traffic won't interfere with the knock sequence.

For the purposes of this example (remotely opening and closing port 22), you didn't need more than a single sequence, shown in Listing 2. However, nothing requires having a single sequence, and for that matter, commands do not have to open ports either! Whenever a knock sequence is recognized, the given command will be executed. In the example, it opened a firewall port, but it could be used for any other

functions you might think of—triggering a backup, running a certain process, sending e-mail and so on.

The knockd command accepts the following command-line options:

- -c lets you specify a different configuration file, instead of the usual /etc/knockd.conf.
- -d makes knockd run as a dæmon in the background; this is the standard way of functioning.
- -h provides syntax help.
- -i lets you change which interface to listen on; by default, it uses whatever you specify in the configuration file or eth0 if none is specified.
- -l allows looking up DNS names for log entries, but this is considered bad practice, because it forces your machine to lose stealthiness and do DNS traffic, which could be monitored.
- -v produces more verbose status messages.
- -D outputs debugging messages.
- -V shows the current version number.

In order to send the required knocks, you could use any program, but the knock command that comes with the knockd package is the usual choice. An example of its usage is shown above (`knock your.site.url 7005`) for a TCP knock on port 7005. For a UDP knock, either add the -u parameter, or do `knock your.site.url 7005:udp`. The -h parameter provides the (simple) syntax description.

Conclusion

Port knocking can't be the only security weapon in your arsenal, but it helps add an extra barrier to your machine and makes it harder for hackers to get a toehold into your system.■

Federico Kereki is an Uruguayan Systems Engineer, with more than 20 years' experience teaching at universities, doing development and consulting work, and writing articles and course material. He has been using Linux for many years now, having installed it at several different companies. He is particularly interested in the better security and performance of Linux boxes.

Resources

The knockd page is at **www.zeroflux.org/projects/knock**, and you can find the knockd man page documentation at **linux.die.net/man/1/knockd**, or simply do `man knockd` at a console.

For more on port knocking, check **www.portknocking.org/view**, and in particular, see **www.portknocking.org/view/implementations** for several more implementations. Also, you might check the critique at **www.linux.com/archive/articles/37888** and the answer at **www.portknocking.org/view/about/critique** for a point/counterpoint argument on port knocking.

Read **en.wikipedia.org/wiki/Transmission_Control_Protocol** for TCP flags, especially SYN. At **www.faqs.org/docs/iptables/tcpconnections.html**, you can find a good diagram showing how flags are used.

Port numbers are assigned by IANA (Internet Assigned Numbers Authority); see **www.iana.org/assignments/port-numbers** for a list.

To test your site, get nmap at **nmap.org**, and also go to GRC's (Gibson's Research Corporation) site at **https://www.grc.com**, and try the ShieldsUp test.

Check **www.netfilter.org** if you need to refresh your iptables skills.

SharePoint Comes Back to San Francisco!

Attend

SPTechCon
The SharePoint Technology Conference

Feb. 10-12, 2010 → San Francisco

"Wide range of topics—good depth—plus it was fun!"
Patrick McMahon, Web/IS Architect, SEPATON Inc.

"Excellent content and speakers. Knowledgeable, real-world examples gave me solutions to various SharePoint problems."
Laura Diorio, Web Systems Manager, Porter Novelli

Choose From 80+ Classes & Workshops

Check Out Our All-Star Faculty!

Shane Young, Randy Drisgill, Todd Klindt, John Ross, Michael Noel, Bob Mixon, Heather Solomon, Andrew Connell, Bill English, Dux Raymond Sy

Joel Oleson, Errin O'Connor, Laura Rogers, Mauro Cardarelli, Mark Miller, Matt Passannante, Mark Rackley, Jason Dearinger, Eric Harlan, Shadeed Eleazer

Phillip Wicklund, Dan Usher, Joshua Haebets, Peter Serzo, Nicola Young, Mark Ferraz, Karuana Gatimu, Steven Fowler, Paul J. Swider, Jennifer Mason

SPECIAL KEYNOTES

Ted Pattison
Co-founder, Critical Path Training

Tom Rizzo
Director of SharePoint, Microsoft

DIAMOND SPONSOR

AvePoint®
Unleashing the Power of SharePoint™

PLATINUM SPONSORS

StoragePoint

commvault

QUEST SOFTWARE
Smart Systems Management

ESRI

KWizCom

GOLD SPONSORS

fpweb.net

MAGIC SOFTWARE

CORASWORKS

MIMOSA SYSTEMS

CASAHL

SILVER SPONSORS

rssbus

azaleos

ComponentOne

For more information, go to **www.sptechcon.com**

REGISTER NOW for Early Bird Rates! SAVE $$$!

A BZ Media Event

February 10-12, 2010

Hyatt Regency San Francisco Airport

Simple Virtual Appliances with Linux and Xen

Use Xen and Linux to make your own ready-to-use software virtual appliances. Create a DNS server, a Web server, a MySQL server—whatever you need, ready to go when you need it.

MATTHEW HOSKINS

Everyone is familiar with hardware appliances in one form or another. It could be a wireless access point at home or a DNS server appliance in the data center. Appliances offer a prebuilt software solution (with hardware) that can be deployed rapidly with minimal hassle. When you couple the "appliance" concept with virtualization, you get virtual appliances—a prebuilt software solution, ready to run on your own hardware with minimal work.

In this article, I provide a hands-on introduction to constructing a simple virtual appliance by assembling readily available components. The framework can be used to build a wide range of appliances.

What Is a Virtual Appliance?

Virtual appliances share many attributes in common with their hardware cousins. In general, both types of appliances have a small footprint, use an embedded or "thin" OS, are single-purpose, provide easy backup and restore, and are Web-managed. Most important, they come ready to rock and roll with minimal configuration. Virtual appliances have the additional benefit of being hosted on your own hardware, so you can host multiple virtual appliances on a single physical host.

Many Linux-based virtual appliances are constructed with an extremely thin OS. This can make installing common software complicated due to dependencies, especially for a beginner. For this example, I decided to use an off-the-shelf free distribution, specifically CentOS, because it uses tools most people are used to. However, we'll cut it to the bone as much as possible.

Collecting the Parts

We are going to build our virtual appliances using the Xen hypervisor, because it's free and comes with most Linux distributions these days. In my examples, I am using CentOS 5.3 for both the host and appliance. The host needs the Virtualization option selected during install, or you can retro-fit an existing Linux system by installing the xen and kernel-xen packages. I chose Xen because it's easy; alternatively, you could use VMware, KVM or any other hypervisor.

You can install CentOS directly from the Internet if you have a good connection, or download it to a local Web or NFS server. In this example, I point to **mirror.centos.org** for the install sources and to a local NFS server for the kickstart config.

We will use the Webmin package to provide Web-based management of our appliance. Webmin has been around for a long time and will provide our appliance with a lot of functionality, like complete Web-based management and simple backup/restore. I downloaded the webmin-1.480-1 RPM from **www.webmin.com** for our appliance. Everything else will be provided by standard CentOS packages.

Installing CentOS

To create a minimal CentOS install for our appliance, we will use a custom kickstart with the --nobase option set. One of the most important concepts of good system management is repeatability—a fully automated kickstart install is repeatable and self-documenting. Our entire OS installation will fit quite comfortably in a 2GB virtual disk and 256MB of memory. We are creating our appliance under /xen, which is a standard location for Xen virtual machines (also known as guests). If you choose another location, make sure either to disable SELinux or adjust your settings. Wherever you put Xen, the disk images need the system_u:object_r:xen_image_t context set.

First, let's create an "appliance-base" guest, which will be used like a template. All the files for this guest will be stored in /xen/appliance-base/. Start by logging in to the Xen host as root and create the virtual disk. Then, grab the Xen vmlinuz and initrd files from the install media:

```
xenhost$ mkdir -p /xen/appliance-base
xenhost$ cd /xen/appliance-base
xenhost$ dd if=/dev/zero of=appliance-base.img \
            oflag=direct bs=1M seek=2048 count=1
1+0 records in
1+0 records out
1048576 bytes (1.0 MB) copied, 0.071271 seconds, 14.7 MB/s
xenhost$ cd /xen
xenhost$ wget \
  http://mirror.centos.org/centos/5.3/os/i386/images/xen/initrd.img
xenhost$ wget \
```

Listing 1. Xen Configuration for Install: appliance-base.install.cfg

```
# Xen Configuration for INSTALL of appliance-base
kernel  = "/xen/vmlinuz"
ramdisk = "/xen/initrd.img"
extra   = "text ks=nfs:192.168.200.10:/home/matt/ks.cfg"
name    = "appliance-base"
memory  = "256"
disk    = ['tap:aio:/xen/appliance-base/appliance-base.img,xvda,w',]
vif     = ['bridge=xenbr0,mac=00:16:3e:00:00:01',]
vcpus   = 1

on_reboot = 'destroy'
on_crash  = 'destroy'
```

Listing 2. Kickstart Configuration: ks.cfg

```
# Kickstart Configuration for MINIMAL CENTOS
install
text
reboot

url --url http://mirror.centos.org/centos/5.3/os/i386/
lang en_US.UTF-8
langsupport --default=en_US.UTF-8 en_US.UTF-8
keyboard us

skipx
network --device eth0 --bootproto dhcp

# The password is "password"
rootpw --iscrypted $1$h5ebo1pm$OHL3De9oalNzqIG1BUyJp0
firewall --disabled
selinux --permissive
authconfig --enableshadow --enablemd5
timezone America/New_York

bootloader --location=mbr
clearpart --all --initlabel
part /boot --fstype ext3 --size=100
part pv.2 --size=0 --grow
volgroup VolGroup00 --pesize=32768 pv.2
logvol /     --fstype ext3 --name=LogVol00 \
             --vgname=VolGroup00 --size=1024 --grow
logvol swap  --fstype swap --name=LogVol01 \
             --vgname=VolGroup00 --size=256
%packages --nobase
coreutils
yum
rpm
e2fsprogs
lvm2
grub
sysstat
ntp
openssh-server
openssh-clients
%post
```

```
  http://mirror.centos.org/centos/5.3/os/i386/images/xen/vmlinuz
```

You have just created a 2GB virtual disk for your appliance. Now, create an appliance-base.install.cfg file, and a ks.cfg file as shown in Listings 1 and 2. Be sure to substitute your CentOS URL or a mirror on the Internet. The last three bytes of the MAC address in the .cfg file are made up; just make sure all your Xen guests are unique.

Now, all you have to do is boot up the Xen guest and watch your appliance's OS install. The install will be fully automated; simply execute the following command and sit back:

```
xenhost$ xm create -c /xen/appliance-base/appliance-base.install.cfg
```

After the install completes, it will shut down the Xen guest and drop back to a shell prompt. Next, still in the same directory, create an appliance-base.cfg, as shown in Listing 3, which will be used to run the appliance in normal mode.

Listing 3. Xen Configuration: appliance-base.cfg

```
# Xen Configuration for appliance-base
name    = "appliance-base"
memory  = "256"
disk    = ['tap:aio:/xen/appliance-base/appliance-base.img,xvda,w',]
vif     = ['bridge=xenbr0,mac=00:16:3e:00:00:01',]
vcpus   = 1

bootloader ="/usr/bin/pygrub"
on_reboot   = 'restart'
on_crash    = 'restart'
```

Boot up the Xen guest again using the new config:

```
xenhost$ xm create -c /xen/appliance-base/appliance-base.cfg
```

And now, you're ready to start installing services.

Installing Web Management

Let's get this guest ready to be an appliance. When the guest is completely booted, log in as root. The password is "password" (this is somewhat of a de facto standard for virtual appliances). Execute the following commands to update fully; then, install Webmin and all its dependencies:

```
appliance-base# wm=http://sourceforge.net/projects/webadmin/files
appliance-base# yum -y update
appliance-base# yum -y install perl wget
appliance-base# wget $wm/webmin/webmin-1.480-1.noarch.rpm/download
appliance-base# rpm -Uvh webmin-1.480-1.noarch.rpm
appliance-base# chkconfig webmin on
```

Finally, add the following snippet of code to the bottom of the /etc/rc.local file:

```
appliance-base# echo "" >> /dev/console
appliance-base# echo "" >> /dev/console
appliance-base# echo "Connect to WEBMIN at: http://$(ifconfig eth0 |
                           grep 'inet addr:' |
                           awk '{ print $2; }' |
                           cut -d: -f2):10000/" >> /dev/console
appliance-base# echo "" >> /dev/console
appliance-base# echo "" >> /dev/console
```

This will output the current IP address for eth0 to tell the user how to connect to Webmin for the first time. This, of course, assumes that the appliance is booting up on a DHCP network. Often a virtual appliance is booted initially with DHCP and then configured via the Web with a static address.

Customizing and Installing Services

At this point, we have a generic virtual appliance ready to customize. To make a MySQL server appliance, run `yum install mysql-server`. To make a DNS appliance, run `yum install` `bind bind-utils`. To make a LAMP appliance, run `yum install httpd php mysql-server`. Reboot, or click Refresh Modules inside Webmin, and you will be presented with Web management for whatever you installed. Webmin supports a very wide range of software right out of the box, and even more with extension modules available on the Webmin Web site.

For our example, let's make a simple MySQL database server appliance. To customize your base appliance, run the following commands inside the VM:

```
appliance-base# yum -y install mysql-server
appliance-base# /etc/init.d/mysqld start
Initializing MySQL database:  Installing MySQL system tables...
OK
appliance-base# mysql_secure_installation

NOTE: RUNNING ALL PARTS OF THIS SCRIPT IS RECOMMENDED FOR ALL MySQL
      SERVERS IN PRODUCTION USE!  PLEASE READ EACH STEP CAREFULLY!

Enter current password for root (enter for none):
OK, successfully used password, moving on...

Setting the root password ensures that nobody can log into the MySQL
root user without the proper authorization.

Set root password? [Y/n] Y
New password: password
Remove anonymous users? [Y/n] Y
Disallow root login remotely? [Y/n] n
Remove test database and access to it? [Y/n] Y
Reload privilege tables now? [Y/n] Y

All done! If you've completed all of the above steps, your MySQL
installation should now be secure.

Thanks for using MySQL!
```

Packaging and Deploying the Appliance

Next, let's package up the appliance and then go through the motions of deploying it as mysql.example.com. To package up the appliance, simply tar up the disk image and configuration:

```
xenhost$ cd /xen/appliance-base
xenhost$ tar -cvzf appliance-base.img appliance-base.cfg
xenhost$ mkdir /xen/mysql.example.com
xenhost$ cd /xen/mysql.example.com
xenhost$ tar -xvzf /xen/appliance-base.tar.gz
xenhost$ mv appliance-base.cfg /etc/xen/auto/mysql.example.com.cfg
xenhost$ vim /etc/xen/auto/mysql.example.com.cfg
```

Listing 4. /etc/xen/auto/mysql.example.com.cfg

```
name   = "mysql.example.com"
memory = "256"
disk   = ['tap:aio:/xen/mysql.example.com/appliance-base.img,xvda,w',]
vif    = ['bridge=xenbr0,mac=00:16:3e:00:00:02',]
vcpus  = 1

bootloader = "/usr/bin/pygrub"
on_reboot  = 'restart'
on_crash   = 'restart'
```

Edit the Xen configuration file /etc/xen/auto/mysql.example.com.cfg as shown in Listing 4. Set the name, the path to the disk image, and give this guest a unique MAC address. Placing the configuration under /etc/xen/auto means the appliance will be started automatically when the Xen host boots.

Start the new appliance using the following command:

```
xenhost$ xm create /etc/xen/auto/mysql.example.com.cfg
xenhost$ vm console mysql.example.com
```

Examine the console output as the guest boots; the last bit of output will have the DHCP-assigned IP, thanks to your rc.local additions. Point a Web browser at the URL shown; by default, Webmin listens on TCP port 10000. Once logged in as root, you will be able to manage your MySQL appliance. Webmin will allow you to set a static IP, maintain YUM updates, create additional users, configure firewall rules, create and maintain MySQL databases and tables, and configure automated system and MySQL backups.

Conclusion

Using these simple steps and readily available components, you can create a thin virtual appliance to do almost anything. Because its a virtual machine, you can move it between physical computers and deploy it multiple times with ease.

As I stated in the introduction, all of these steps could have been done with VMware virtualization products. VMware is certainly the most widely deployed technology and has its own tools for creating virtual appliances, including an on-line "Appliance Marketplace" for sharing prebuilt appliances. No matter whether you use VMware or Xen, virtual appliances are a simple way to deploy preconfigured services with minimal hassle. If you are a software author, it allows you to hand your customers a "known working configuration" every time.■

Matthew Hoskins is a UNIX/Storage and Virtualization Administrator for The New Jersey Institute of Technology where he maintains many of the corporate administrative systems. He enjoys trying to get wildly different systems and software working together, usually with a thin layer of Perl (locally known as "MattGlue"). When not hacking systems, he often can be found hacking in the kitchen. Matt can be reached at matthoskins@gmail.com.

Resources

CentOS Linux: **www.centos.org**

Webmin: **www.webmin.com**

VMware Virtual Appliance Marketplace: **www.vmware.com/appliances**

KVM | Clutter | FPGAs | Android | Coreboot | RSpec | SRANDOM
LINUX JOURNAL
LINUX KVM for Low-Level System Programming
Bare-Metal Hacks with FPGA PROGRAMMING
coreboot
Unchain Your BIOS with the Ultimate Hack
PLUS: PUT ANDROID ANYWHERE
Spam-Fighting Tricks for Mutt and Postfix
Develop OpenGL Apps with Clutter
Laptop Hardening Tips
LAZARUS a Powerful IDE for Cross-Platform Development
Qt Making Apps Right on All Platforms
Build Rich Desktop Apps with TITANIUM
CHROME
Open-Source Compliance Best Practices
Looking at AppArmor in Ubuntu
Use tircd to Connect to Twitter over IRC
Bitlbee | KOffice 2.0 | Ksplice | Rootkits | squidGuard
Defend against /dev/mem Attacks
Using Fixtures and Factories in Rails Applications
KERNEL CAPERS
KOffice 2.0
Point/Counterpoint Kyle and Bill Debate the Merits of Twitter
Conkeror | Android | Squid | Kindle 2 | Indamixx | BUG
Hacks and Apps for the BUG
Keeping Track of Multiple Servers
LINUX

KYLE RANKIN

BILL CHILDERS

Education vs. Experience

Should you get certified?

The past few columns, we've talked about everything from laptops to filesystems. Our Point/Counterpoint this time around touches on a subject somewhat less technical, but no less important. This month, we debate education versus experience—in today's workplace, which carries more weight?

KYLE: So the fact of the matter is, even if you get a formal Computer Science education, typically, you still get only a basic foundation. Most CS programs seem to be geared more toward programming than system administration, and even when they try to stay up to date, it's just hard to keep the academic curriculum up to the pace of technology. When you get into the workforce, you find you have to learn specific aspects of technology *somehow*, either through on-the-job training (if you can get it) or specific certifications. Although I think Bill and I both agree that nothing can quite beat experience, how can you get the experience if you can't get hired? That's where things like certifications can come in.

I have seen "certified" people who couldn't administer their way out of a paper bag.

BILL: Maybe it's me, but as a hiring manager, I gloss right over certifications. Seriously. I give them a cursory glance, then jump right to the experience and skills list. Certification may prove useful in some cases, but I prefer to have people who are generalists on my staff.

KYLE: I guess that was lucky for me, because I didn't have any certs when Bill looked over my résumé. I think when it comes to certifications, it really depends on what the position is and what the certification is. If you've been in the industry for any period of time, you inevitably will run into someone who has a certification but doesn't seem to know the first thing about the topic. Does that mean the cert is bad? Not necessarily. It just highlights how difficult it is to force someone actually to learn material rather than cram information long enough to pass a test.

BILL: Exactly. I have seen "certified" people who couldn't administer their way out of a paper bag. I've seen Cisco-certified people who have no understanding of DNS, for example. Kyle's point is well taken—a cert may be just an indicator of a person who can cram and test well.

KYLE: I think formal training really comes in handy when you need to get ramped up on technology that has a high cost of entry (for instance, some enterprise virtualization products or SAN infrastructure). Most people don't have the funds to buy a bunch of expensive SAN gear or software licenses so they can build up experience at home. In those cases, formal training can be a good way to kickstart people onto a technology before they actually have to deploy it. I've also seen it come in handy when you hire people who have experience in 90% of what you need. Getting them formal training or certification can be a quick way to ramp them up on the remaining 10%, so they can learn the rest on the job.

BILL: Kyle, you're a bad example. When I hired you, I'd already known you for some time and had a good idea what you were capable of. However, there's another employee I hired without certs and much experience, but based on his résumé, I gave him a shot at an interview, and he knocked me out. He ended up rounding out our team quite nicely, and we had a good, successful run for quite some time.

KYLE: I think this is one of those rare times when it's a good thing to be a bad example.

BILL: Your point about getting certified in expensive technology is well taken; however, I remember a certain bloke who had no formal SAN training and did quite well for a long time administering, and later architecting, an EMC SAN solution. I'm of the opinion that good generalists, with a good head on their shoulders and a solid thought process and troubleshooting methodology can figure anything out. I've watched many seasoned "certified" technicians lock up when they are forced into uncharted waters by some esoteric issue.

KYLE: Heh, well I think that guy was lucky to have a good work environment to get up to speed. I have to agree with Bill that when I see certifications on a résumé, I always take them with a grain of salt. Trust but verify. That's why most technology positions have some sort of technical vetting process—a certification might get you in the door, but you still have to prove you know your stuff once you get there.

BILL: That guy also was lucky to have a manager who valued a good work environment. But I digress on that. Back to the topic, a good technical interview is a great vetting process, and one that I live by.

KYLE: I think the same thing goes for most education. It really depends on how you approach it. If you look at a cert as a quick way to get a piece of paper and get a job, you will miss out. On the other hand, if you actually use it as an opportunity to learn and remember some new technology, especially if you couple it with plenty of actual experience, I think it can be very valuable. You know, we've gone this whole time without specifically mentioning Linux certs. What do you think of those, Bill?

BILL: I put less stock in them in general. It shows me people were interested enough to get the cert, but that really doesn't give me a clue as to their abilities. I'd much rather see a list of skills and accomplishments, and then ask people about the most interesting thing they've done in the last six months, either via a phone interview or in person. Hearing people get excited about technology they've grappled with and been successful with gives me an idea of how vested they are in their job.

KYLE: I think I agree with you here. I actually think sometimes some Linux certs can hurt. Unlike with other certifications, there's nothing really stopping anyone from building their own Linux network and learning at home. When it comes to Linux knowledge, I *almost* put equal stock in personal and professional experience, and those are the most important things to me besides just having a good fundamental knowledge of Linux, networking and so on. The challenge with Linux certs is that Linux is a fast-moving technology. Not to single anyone out, but as an example, the LPIC cert still had IRQ settings as a fundamental part of the curriculum until recently. I can't remember the last time I needed to tweak that. To their credit, this spring the entire course was revamped, and now it more accurately reflects what you would need to know on a modern Linux system. It's actually a good course now as a result. It shows though that if you are shopping around for Linux certs, be sure to check the last time the course was updated.

BILL: That's a good point. Often the curriculum will lag behind the current technology, particularly when dealing with open-source technologies, such as Linux. Other OSS technologies also suffer from this as well, which is another reason why I'm an advocate of staying on top of technology yourself.

KYLE: To summarize, I think (and I think Bill will mostly agree with me here) that certifications can be a good way to get ramped up on a new technology, but they are no substitute for actual experience. Not all certs are equal. Do research on how frequently they are updated, and if you do take formal training, don't just cram to pass a test. Use the class as a catalyst to learn the technology in a way that you will remember far after the test is over.

> If you look at a cert as a quick way to get a piece of paper and get a job, you will miss out.

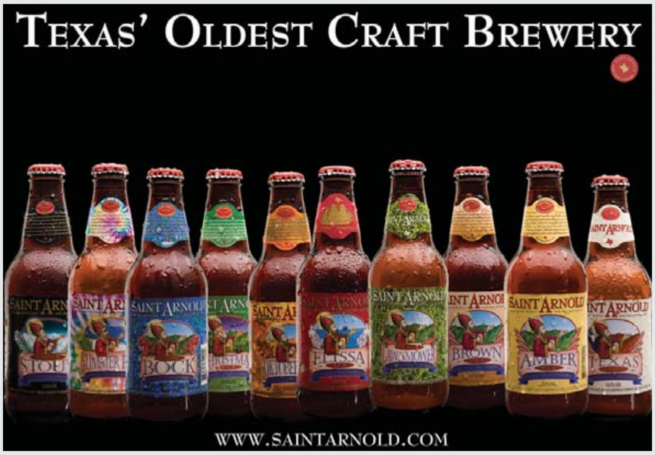
BILL: You can't go wrong with experience. Certs don't hurt, but in my book, they don't add as much value as many people think. Experience, a solid grounding in the fundamentals, a great work ethic and excitement about what you're doing are things I look for when I'm hiring a new employee. Unless you want to be something very specialized, certs are a minor differentiator in my book. I'd rather hire someone who's active in a LUG, contributes to projects like Fedora or Ubuntu and has some track record in the community. If you've contributed and been active in open source, you rate higher on my "hiring manager" radar than someone who's attended a lot of courses.■

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

Bill Childers is an IT Manager in Silicon Valley, where he lives with his wife and two children. He enjoys Linux far too much, and he probably should get more sun from time to time. In his spare time, he does work with the Gilroy Garlic Festival, but he does not smell like garlic.

INNOVATION ON THE GO
ORDER YOUR BEAGLE BOARD FROM DIGIKEY.COM
AVAILABLE EXCLUSIVELY AT DIGI-KEY
beagleboard
only $149⁰⁰
LOW-COST, NO FAN, SINGLE-BOARD COMPUTER
Digi-Key® CORPORATION
www.digikey.com

axigen MAIL SERVER
The Mail Server
for IT Professionals
www.axigen.com


Texas' Oldest Craft Brewery
SAINT ARNOLD
WWW.SAINTARNOLD.COM

Natuba™
HAS AN API THAT EVEN
SHAWN POWERS
IS COMFORTABLE USING.
IT'S CALLED "EMAIL."
Shawn Powers
Linux Journal Associate Editor
Follow Shawn on Natuba!
Natuba.com/shawnp0wers
RUNS ON LINUX
Try it on your phone!
Share your photos on Natuba.com today! Expose yourself™
© 2009 Hush Labs, LLC. All rights reserved.

Now Data Gets Personal

Know thyself. DOC SEARLS

The main problem with data is that it's easy to copy. In fact, sending it from one place to another is essentially an act of replication. The mv command is alien to most people's experience of the Internet. They may use it every day (hardly realizing it) within their own filesystems, but between separate systems on the Net, their experience is that of replication, not relocation.

This alone makes control of one's data problematic, especially when the first-person possessive voice isn't quite right. "My" data often isn't. For example, take profile or activity data kept by a service provider. It's from you and about you, but you don't own it, much less control it. Transaction data is created by a buyer and a seller together, but the canonical form of the data is what's kept by the seller and provided (by copying) in the form of a bill or displayed on an encrypted personal Web connection. Sellers don't go much further than that. The idea of sharing that information in its raw form, either during a transaction or later on request by the buyer, is alien at best to most sellers' IT and legal departments. As John Perry Barlow put it in "Death From Above" (way back in 1995, **w2.eff.org/Misc/Publications/John_Perry_Barlow/HTML/death_from_above.html**), "America remains a place where companies produce and consumers consume in an economic relationship which is still as asymmetrical as that of bomber to bombee." In fact, this is still true of the whole business world.

Yet, internetworking of that world brings a great deal of symmetricality to it, imposed by the architecture of the Internet and its growing suite of protocols. The bank that used to occupy the most serious building on Main Street—or a skyscraper in a big city—is now but one location among a trillion on the Web. Yours is another. The word "domain" applies to both of you, even if your bank's "brand" is bigger than yours. Of your own sense of place and power on the Net, the words of William Cowper apply (**www.bartelby.com/41/317.html**): "I AM monarch of all I survey; / My right there is none to dispute..."

Yet, as William Gibson famously said, "the future is here but not evenly distributed". Bomber/bombee power asymmetries persist in the B2C (business-to-consumer) world of everyday retailing. When you buy something, the transaction data in most cases comes to you only in the form of a receipt from the seller and a bill from the credit-card company. Neither is offered in formats that allow you to gather data on the spot or later over a secure Net connection—not easily, anyway.

If we could collect that data easily, our self-knowledge and future purchases would be far better informed. In fact, collected data could go far beyond transaction alone. Time, date, location, duration, sequence—those are obvious ones. How about other bits of data, such as those involved in dealings with airlines? For example, your "fare basis code" (HL7LNR, or some other collection of letters and numbers) contains piles of information that might be useful to you as well as the airline, especially as you begin to add up the variables over time.

A marketplace is no better than the knowledge and practices that buyers and sellers both bring to it. But, while the Net opens many paths for increasing knowledge on both sides, most of the knowledge-gathering innovation has gone into helping sellers. Not buyers.

Today, that's changing. More and more buyers (especially the geeks among them) are getting around to helping themselves. In particular, two new development categories are starting to stand out—at least for me. One is self-tracking, and the other is personal informatics.

Compared to its alternative (basically, guessing), self-tracking is "know thyself" taken to an extreme. Alexandra Carmichael, for example, tracks 40 things about herself, every day. These include mood, chronic pain levels, sexual activity, food intake and so on. She's a star in the Quantified Self community (**www.kk.org/quantifiedself**), which is led by Gary Wolf and Kevin Kelly. Among topics at QS meetups are chemical body load, personal genome sequencing, lifelogging, self-experimentation, behavior monitoring, location tracking, non-invasive probes, digitizing body info, sharing health records, psychological self-assessments and medical self-diagnostics, to name a few.

Now, would any of these be extreme if they were easy and routine? Well, that's the idea. ListenLog (**cyber.law.harvard.edu/projectvrm/ListenLog**), one of the projects I'm involved with, doesn't make sense unless it's easy, and unless the data it yields is plainly valuable.

This brings us to personal informatics, which is a general category that includes self-tracking and extends to actions. All this data needs to live somewhere, and stuff needs to be done with it.

In the commercial realm, I see two broad but different approaches. One is based on a personal data store that might be self-hosted by the customer or in a cloud operated by what we call a fourth-party service (serving the buyer rather than the seller—to differentiate it from third parties, which primarily serve sellers). As Iain Henderson (who leads this approach) puts it, what matters here "is what the individual brings to the party via their personal data store/user-driven and volunteered personal information. They bring the context for all subsequent components of the buying process (and high-grade fuel for the selling process if it can be trained to listen rather than shout)." The other approach is based on complete user autonomy, whereby self-tracking and personal relationships are entirely the responsibility of the individual. This is exemplified by The Mine! Project (**themineproject.org/about**), led by Adriana Lukas. As she puts it, the difference between the two approaches is providing vs. enabling (**www.mediainfluencer.net/2009/04/enabling-vs-providing**).

Either way, the individual is the primary actor. As distribution of the future evens out, the individual has the most to gain.■

Doc Searls is Senior Editor of *Linux Journal*. He is also a fellow with the Berkman Center for Internet and Society at Harvard University and the Center for Information Technology and Society at UC Santa Barbara.

Orion iX-N4224

The **Orion** features **unrivaled cooling efficiency, power efficiency** and excellent **storage density!**

Superior Savings On Energy Costs

Designed for storage-intensive applications and virtualization, the iX-N4224 4U storage server series delivers incredible performance, storage capacity, and energy savings with adaptability and superior hardware. Each iX-N4224 comes with a **Gold Level power supply**, boasting a high 93% energy efficiency. Powerful Intel® Xeon® 5500 series quad core processors **intelligently save power during low-use periods** and increase performance when systems require it. Intel® Xeon® 5500 series processors include virtualization technologies to lead the way in performance, scalability, and simplified server management and migration.

The iX-N4224 supports up to 144GB of DDR3 1333 energy efficient RAM and utilizes three 5000 RPM cooling PWM fans and two 5000 RPM rear exhaust PWM fans. iX-N4224 servers offer up to 48 terabytes of storage with 24 hot-swappable SAS/SATA drive bays in a 4U configuration. Storage sizes for the iX-N4224 are customizable, with 250GB, 500GB, 750GB, 1TB, and 2TB hard drives available.

The iX-N4224 provides the ideal solution for applications requiring maximum storage capacity and power savings. For particularly storage-hungry applications, Western Digital® offers 2TB WD™ RE4-GP hard drives, which offer lower power use during idle times, a 64 megabyte cache, up to 25% increased performance, and a savings of up to $10 per drive on yearly power costs. Each hard drive is equipped with improvements to rotary vibration tolerance and calculates optimum seek speeds to lower power consumption, noise, and vibration. These drives also require less power and time to start up, allowing more drives to start spinning simultaneously due to the decrease in the current each drive requires. Equipping the Orion iX-N4224 4U storage server with the WD™ RE4-GP drives provides unparalleled storage capacity and power efficiency.

For more information about the Orion Series visit *http://www.iXsystems.com/Orion*.

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

Intel, the Intel logo, and Xeon Inside are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries.

Features

- Dual 64-Bit Socket 1366 Quad-Core or Dual-Core Intel® Xeon® Processor 5500 Series
- 24 x 3.5" SAS/SATA Hot-swappable Drive Bays
- 1200W high-efficiency (1+1) redundant power supply (Gold Level 93%)
- 100% Cooling Redundancy
- Dual Intel® 5520 chipsets with QuickPath Interconnect (QPI)
- Up to 144GB DDR3 1333/1066/800 SDRAM ECC Registered Memory (18 DIMM Slots)
- 2 PCI-E 2.0 x16, 4 PCI-E x8 (1 in x16 slot), and 1 PCI-E x4 Expansion Slots
- Intel® 82576 Dual Port Gigabit Ethernet Controller
- Optional 2x Internal Fixed 3.5" HDD or 2x fixed 2.5" HDD + DVD
- Matrox G200eW Graphics
- Remote Management - IPMI 2.0 + IP-KVM with dedicated LAN

Call 1-800-820-BSDi Today!

iX systems™

intel® Xeon® inside™

Powerful. Intelligent.