LINUX JOURNAL™

Since 1994: The Original Magazine of the Linux Community

SEPTEMBER 2014 | ISSUE 245 | www.linuxjournal.com

AN IN-DEPTH LOOK AT **ZFS vs. BTRFS**

HOW-TOs

Ease the Pain of Provisioning X.509 Certificates

Synchronize Your Life with ownCloud

Check Your Exchange InBox from the Command Line

+ 12-Factor, Scalable, Maintainable **WEB APPS**

pi-web-agent
A DESKTOP ENVIRONMENT FOR THE RASPBERRY PI

DNSMasq
A HERO FOR SERVERS

OpenAxiom
A COMPUTER SYSTEM FOR ALGEBRA

presented by IT-oLogy™

ALL THINGS OPEN™

OCT 22-23, 2014 | RALEIGH CONVENTION CENTER

In Raleigh and the Research Triangle: A Global Epicenter of Technology and Open Source Innovation

A CONFERENCE EXPLORING OPEN SOURCE, OPEN TECH AND THE OPEN WEB IN THE ENTERPRISE.

LEADING TECHNOLOGY AND BUSINESS LEADERS WILL PARTICIPATE

SEAN MARTELL
Mozilla

DOUG CUTTING
Cloudera

JAMES TURNBULL
Docker

KAREN SANDLER
Software Freedom Conservancy

DWIGHT MERRIMAN
MongoDB

YEHUDA KATZ
Tilde

ELIZABETH JOSEPH
Hewlett Packard

STEVEN VUGHAN-NICHOLS
ComputerWorld
& CBS/ZDNet

Two days of keynotes, talks, tutorials, workshops and networking opportunities in Raleigh and the Research Triangle area

REGISTER NOW!

ALLTHINGSOPEN.ORG

SILICON MECHANICS
SERVERS • STORAGE • HPC

Are you tired of dealing with proprietary storage?

zStax®
ZFS Unified Storage
Powered by NexentaStor

zStax StorCore ZFS Unified Storage from Silicon Mechanics is truly software-defined storage.

From modest data storage needs to a multi-tiered production storage environment, **zStax StorCore** ZFS unified storage appliances have the right mix of performance, capacity, and reliability to fit your needs.

zStax StorCore 64

The **zStax StorCore 64** utilizes the latest in dual-processor Intel® Xeon® platforms and fast SAS SSDs for caching. The zStax StorCore 64 platform is perfect for:

- small-medium office file servers
- streaming video hosts
- small data archives

zStax StorCore 104

The **zStax StorCore 104** is the flagship of the zStax product line. With its highly available configurations and scalable architecture, the zStax StorCore 104 platform is ideal for:

- backend storage for virtualized environments
- mission critical database applications
- always available active archives

Talk with an expert today: 866-352-1173 - http://www.siliconmechanics.com/zstax

CONTENTS SEPTEMBER 2014 ISSUE 245

HOW-TOs

FEATURES

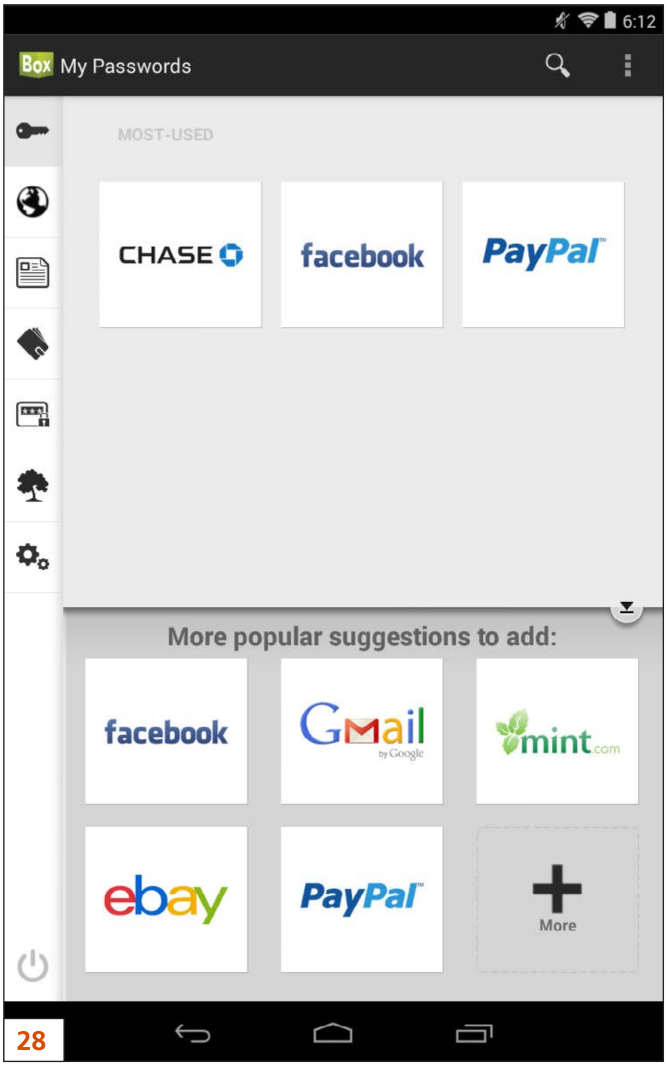
ON THE COVER

Cover Image: © Can Stock Photo Inc. / alexaldo

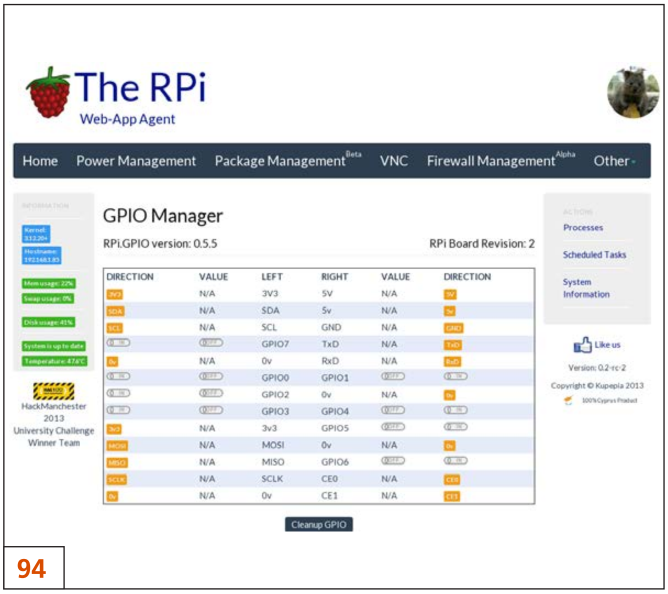
INDEPTH

COLUMNS

IN EVERY ISSUE



28



GPIO Manager

RPi.GPIO version: 0.5.5

RPi Board Revision: 2

DIRECTION	VALUE	LEFT	RIGHT	VALUE	DIRECTION
3V3	N/A	3V3	5V	N/A	5V
SDA	N/A	SDA	5v	N/A	5v
SCL	N/A	SCL	GND	N/A	GND
IN	OFF	GPIO7	TxD	N/A	TxD
0v	N/A	0v	RxD	N/A	RxD
IN	OFF	GPIO0	GPIO1	OFF	IN
IN	OFF	GPIO2	0v	N/A	0v
IN	OFF	GPIO3	GPIO4	OFF	IN
3v3	N/A	3v3	GPIO5	OFF	IN
MOSI	N/A	MOSI	0v	N/A	0v
MISO	N/A	MISO	GPIO6	OFF	IN
SCLK	N/A	SCLK	CE0	N/A	CE0
0v	N/A	0v	CE1	N/A	CE1

Cleanup GPIO

94

LINUX JOURNAL (ISSN 1075-3583) is published monthly by Belltown Media, Inc., 2121 Sage Road, Ste. 395, Houston, TX 77056 USA. Subscription rate is $29.50/year. Subscriptions start with the next issue.

LINUX JOURNAL™

Subscribe to *Linux Journal* Digital Edition for only **$2.45 an issue.**

ENJOY:

Timely delivery

Off-line reading

Easy navigation

Phrase search and highlighting

Ability to save, clip and share articles

Embedded videos

Android & iOS apps, desktop and e-Reader versions

SUBSCRIBE TODAY!

LINUX JOURNAL

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

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

Publisher	Carlie Fairchild publisher@linuxjournal.com
Director of Sales	John Grogan john@linuxjournal.com
Associate Publisher	Mark Irgang mark@linuxjournal.com
Webmistress	Katherine Druckman webmistress@linuxjournal.com
Accountant	Candy Beauchamp acct@linuxjournal.com

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

Editorial Advisory Panel
Brad Abram Baillio • Nick Baronian • Hari Boukis • Steve Case
Kalyana Krishna Chadalavada • Brian Conner • Caleb S. Cullen
Keir Davis • Michael Eager • Nick Faltys • Dennis Franklin Frey
Victor Gregorio • Philip Jacob • Jay Kruizenga • David A. Lane
Steve Marquez • Dave McAllister • Carson McDonald • Craig Oda
Jeffrey D. Parent • Charnell Pugsley • Thomas Quinlan • Mike Roberts
Kristin Shoemaker • Chris D. Stark • Patrick Swartz • James Walker

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

Subscriptions
E-MAIL: subs@linuxjournal.com
URL: www.linuxjournal.com/subscribe
MAIL: PO Box 980985, Houston, TX 77098 USA

LINUX is a registered trademark of Linus Torvalds.

11th Annual

High Performance Computing for Wall Street Show and Conference

September 22, 2014 (Monday) **Roosevelt Hotel, NYC**
Madison Ave and 45th St, next to Grand Central Station

Register and Save $100 for Low Cost Conference. Register for Free Show.

Big Data, Cloud, Linux, Low Latency, Networks, Data Centers, Cost Savings.

Wall Street IT professionals, code writers and programmers will assemble at this 2014 HPC Show and Conference, Sept. 22.

New for 2014 – Database Month code writers and programmers to speak on the program.

This 11th Annual HPC networking opportunity will assemble 600 Wall Street IT professionals, code writers and programmers at one time and one place in New York.

This HPC for Wall Street conference is focused on High Put-through, Low Latency, Networks, Data Centers, lowering costs of operation.

Our Show is an efficient one-day showcase and networking opportunity.

Leading companies will be showing their newest systems live on-the-show floor.

Register in advance for the full conference program which includes general sessions, drill-down sessions, a new code writers track, an industry luncheon, coffee breaks, exclusive viewing times in the exhibits, and more. Save $100. $295 in advance. $395 on site.

Don't have time for the full Conference?
Attend the free Show. Register in advance at: www.flaggmgmt.com/hpc

Show Hours: Mon, Sept 22	8:00 - 4:00
Conference Hours:	8:30 - 4:50

2014 Gold Sponsors

CISCO HP redhat. SI SCALABLE INFORMATICS NOVASPARKS NYC NoSQL NEW YORK NewSQL

Media Sponsors

LINUX JOURNAL LINUX NEW MEDIA The Pulse of Open Source SECURITIES TECHNOLOGY MONITOR datanami Integration developer news TRADERS MAGAZINE HPC wire ENTERPRISETECH MONEY management executive WSTA Wall Street Technology Association

Show & Conference: Flagg Management Inc
353 Lexington Avenue, New York 10016
(212) 286 0333 fax: (212) 286 0086 flaggmgmt@msn.com

Visit: www.flaggmgmt.com/hpc

Among the Featured Speakers:

John Ramsay
Chief Market Policy & Regulatory Officer, IEX Group, formerly with the SEC Regulatory Division of Trading & Markets

Brian Bulkowski
CTO & Founder, Aerospike, Speaker on the Code Writing Panel

Phil Albinus
Editor
Traders Magazine, SourceMedia, Moderator on the Low Latency Session

Jeffrey M Birnbaum
Founder & CEO, 60East Technologies, Inc., Moderator on the Code Writing Panel

Jeffrey Kutler
Editor-in-Chief, GARP Risk Professional, Moderator of HPC Session

Dino Vitale (invited)
Director, Morgan Stanley Quality Assurance and Production Management

Harvey Stein
Head of Credit Risk Modeling, Bloomberg

Cory Isaacson
CEO / Chief Technology Officer, CodeFutures, Speaker on the Code Writing Panel

Current_Issue.tar.gz

SHAWN POWERS

How'd You *Do* That?

Open-source advocates tend to make for rotten magicians. Whereas most illusionists insist on taking their secrets to the grave, we tend to give away the secret sauce to anyone who'll listen. Heck, sometimes we create things just so we can explain to others how they work! And, that is how this issue was born. We love the How-To concept. Heck, our entire magazine is based on the idea of spreading knowledge, and this month, we specifically go out of our way to show not only the result, but the "How" as well.

Reuven M. Lerner starts us off with a discussion of 12-Factor Apps. He basically describes a process for developing scalable, maintainable Web applications. As someone who recently started creating Web apps, I can attest that they get unmanageable quickly! Dave Taylor follows with some smarter code for solving the "how many days have passed" script he's been working with for the past few months.

VIDEO:
Shawn Powers runs through the latest issue.

All too often, our perfectly crafted solutions get ruined by someone changing something outside our control. Kyle Rankin recently had an issue with his fetchmail setup, and he walks through the process of troubleshooting and problem solving when someone changes the rules. If, like me, you're a fan of Kyle's passion for the command line, you'll appreciate his efforts to maintain his ASCII lifestyle.

I made true on my promise to get a little more serious this month and wrote about DNSMasq. It's not a new program, but it's so powerful and so simple, it's often overlooked as a viable option for serving DNS and DHCP. Although most people are fine with running DNS and DHCP from their off-the-shelf routers, there are times when you need to run one or both of the services on a server. That's just what I needed to do, and I was pleasantly surprised at how powerful the little dæmon can be!

Much like car alarms, self-signed SSL certificates are all too often just accepted, especially on systems we're familiar with using. The problem is that if there is a compromise on one of our trusted systems, an invalid certificate might be the only warning we get. John Foley walks through the entire process for using PKI certificates. Whether you are an old hand at creating certs for VPNs or just copy/paste something from Google whenever you need to create a Web cert, his article is interesting and educational.

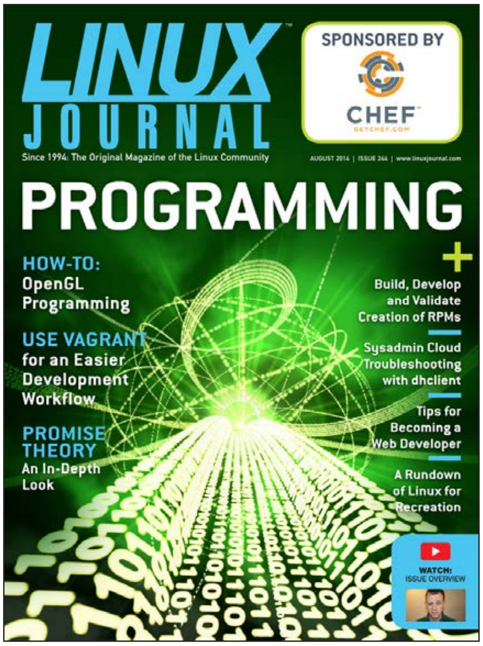
Last year you may recall I mentioned ownCloud as an alternative to Dropbox for those willing and able to host such a service on their own. Mike Diehl takes it to the next level with an incredible how-to on setting up, configuring and using ownCloud for all your cloud-based needs. At its core, ownCloud does indeed sync files, but it does so much more, it's worth taking a look at. And when it comes to file storage on your server, Russell Coker addresses another extremely important topic: data corruption. Using ZFS or BTRFS filesystems can protect your data, but which is better? Which should you choose? Russell answers all your questions and more.

If there's one product that fits into our How-To issue, it's the Raspberry Pi. It's the heart of just about every Linux-based DIY project on the Internet, and those little beauties run half the projects around my house as well. A quartet of authors (Vasilis Nicolaou, Angelos Georgiadis, Georgios Chairepetis and Andreas Galazis) give us a great description of pi-web-agent. Although the little RPi devices are incredible, they also are a bit intimidating for new Linux users. pi-web-agent changes that by providing a complete Web front end for managing and controlling the Raspberry Pi, making the RPi accessible to everyone!

If you've ever wanted to work with Linux, but weren't sure "how to" get started, this issue is for you. And if you're an old hat who wants to add more skills to your tech quiver? Again, for you too. To be honest, the entire Linux community is based on sharing information and collaborating ideas. There's no illusion at play, but there's plenty of magic! ■

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

letters


LINUX JOURNAL
SPONSORED BY
CHEF
Since 1994: The Original Magazine of the Linux Community
AUGUST 2014 | ISSUE 244 | www.linuxjournal.com
PROGRAMMING
HOW-TO: OpenGL Programming
USE VAGRANT for an Easier Development Workflow
PROMISE THEORY An In-Depth Look
+
Build, Develop and Validate Creation of RPMs
Sysadmin Cloud Troubleshooting with dhclient
Tips for Becoming a Web Developer
A Rundown of Linux for Recreation
WATCH: ISSUE OVERVIEW

Linux Journal—NSA's "Extremist Forum"

Just came across: "If you visit the forum page for the popular *Linux Journal*, dedicated to the open-source operating system Linux, you could be fingerprinted regardless of where you live because the XKEYSCORE source code designates the *Linux Journal* as an 'extremist forum'" (**http://www.defenseone.com/technology/2014/07/if-you-do-nsa-will-spy-you/88054/?oref=govexec_today_nl**).

—Andy Bach

Crazy, isn't it? Although it's frustrating to draw the attention of the NSA, we don't plan on changing how we do things. We're advocates of freedom, and if anything, this makes us more so!—Shawn Powers

Vim Macros

Regarding Kyle Rankin's "The Only Mac I Use" article in the July 2014 issue: I knew all about macros. What I didn't know was what <ctrl>-a did. That is going to save me so much time in the future.

—Rick

Extremist Bonus Points?

After the latest news regarding the XKEYSCORE program, I felt it was a great time to subscribe. Now that I subscribed, do I get a free upgrade to extremist financier?

Chris at **http://jupiterbroadcasting.com** brought me—LAS ep320.

—Sean Rodriguez

Although we haven't seen any "upgrades" yet ourselves, I must admit I'm a little more nervous than usual when I go through airport security now!—Shawn Powers

Leap Year Problems

Regarding Dave Taylor's "Days Between Dates" article in the July

2014 issue, I think he has a big problem with the valid_date.sh way of checking for leap year:

```
harrie@v1:~> ./lj243_valid_date.sh 2 29 1776
checking for feb 29 : was 1776 a leap year?
Yes, 1776 was a leapyear, so February 29, 1776 is a valid date.
harrie@v1:~> ./lj243_valid_date.sh 2 29 1929
checking for feb 29 : was 1929 a leap year?
Yes, 1929 was a leapyear, so February 29, 1929 is a valid date.
```

Well, 1929 was not a leap year. Using `grep -w` solves this:

```
harrie@v1:~> ./lj243_valid_date-w.sh 2 29 1929
checking for feb 29 : was 1929 a leap year?
Oops, 1929 wasn't a leapyear, so February only had 28 days.
harrie@v1:~> cat lj243_valid_date-w.sh
mon=$1; day=$2; year=$3
if [ $mon -eq 2 -a $day -eq 29 ] ; then
echo checking for feb 29 : was $3 a leap year?
leapyear=$(cal 2 $year | grep -w 29)
if [ ! -z "$leapyear" ] ; then
echo "Yes, $year was a leapyear, so February 29, $year \
is a valid date."
else
echo "Oops, $year wasn't a leapyear, so February only \
had 28 days."
fi
fi
```

—Harrie Wijnans

***Dave Taylor replies:** Yup, someone else also pointed out the flaw in my leap year test. Got a much better one in the next column. Thanks for writing in, Harrie!*

Harrie Wijnans replies: Yeah, my solution, of course, fails for the year 29. Adding `tail` solves that (and the `-w` flag for `grep` has no real meaning anymore):

```
leapyear=$(cal 2 $year | tail +3 | grep -w 29)
```

***Dave Taylor replies:** Nah, the solution is to take a sharp left turn and think about it differently. The question isn't "is there a Feb 29", but "how many days are in the year", So, with help from GNU date:*

```
$ date -d 12/31/1929 +%j
365
$ date -d 12/31/1932 +%j
366
```

With this test, you can see that 1932 was a leap year, but 1929 wasn't.

Harrie Wijnans replies: That one fails for years < 1901:

```
harrie@v1:~> date -d 1901/12/31
Tue Dec 31 00:00:00 AMT 1901
harrie@v1:~> date -d 1900/12/31
date: invalid date `1900/12/31'
```

That is, for the date on my Linux (openSUSE 12.1), and it also fails for 12/31/1900:

```
harrie@v1:~> date --version
date (GNU coreutils) 8.14
Copyright (C) 2011 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later
 ➥<http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.

Written by David MacKenzie.
```

***Dave Taylor replies:** Actually, here's the cleanest version of all of this:*

```
function isleap
{
  # If you have GNU date on your Linux system, this is superior:
  #   leapyear=$(date -d 12/31/$1 +%j | grep 366)

  # If you don't have GNU date (Mac OS X doesn't, for example),
  # use this:

  leapyear=$(cal -j 12 $1 | grep -E '[^12]366')
}
```

But, there's something else going on then, because I also have date 8.4, and it works fine:

```
$ date -d 12/31/1844 +%j
366
$ date -d 12/31/1810 +%j
365
```

Hmm...

Harrie Wijnans replies: The date.c from the 18.4 that I got from **http://rpmfind.net/linux/sourceforge/m/ma/magiclinux-plus/Source26/coreutils-8.14-1mgc25.src.rpm** uses `parse_datetime` to fill a `struct tm`, which has:

```
/* Used by other time functions.  */
struct tm
{
  int tm_sec;           /* Seconds.     [0-60] (1 leap second) */
  int tm_min;           /* Minutes.     [0-59] */
  int tm_hour;          /* Hours.       [0-23] */
  int tm_mday;          /* Day.         [1-31] */
  int tm_mon;           /* Month.       [0-11] */
  int tm_year;          /* Year - 1900.  */
  int tm_wday;          /* Day of week. [0-6] */
  int tm_yday;          /* Days in year.[0-365] */
  int tm_isdst;         /* DST.         [-1/0/1]*/

#ifdef  __USE_BSD
  long int tm_gmtoff;       /* Seconds east of UTC.  */
  __const char *tm_zone;    /* Timezone abbreviation.  */
#else
  long int __tm_gmtoff;     /* Seconds east of UTC.  */
  __const char *__tm_zone;  /* Timezone abbreviation.  */
#endif
};
```

Apparently, openSUSE library does not accept the year field to be <= 0. Sorry, I'm lost here. I'm not that experienced in Linux C programming, and lib/parse-datetime.y even uses bison, which I hardly ever used.

Beware of the `cal -j` you gave as an example. Try it for 2000, which is missed as a leap year. (Are we lucky it was a leap year? Otherwise, the "millennium-bug" would have been much more severe.)

This is because there, 366 is at the start of the line, so there are no characters in front of it, hence it does not match `[^12]366`. An extra `-e ^366` would solve that and still not see 1366 or 2366 as leap years; 366 still would be considered a leap year.

I thought about the `cal 2 $year`, and `grep -v February`, but that is language-dependent. (Try `LANG=nl_NL.UTF8 cal 2 2012` or `fr_FR.UTF8`—no February in there.)

I'd say, the person who claims creating scripts for everyone and every environment is simple, never tried it.

Looking forward to reading your next column.

Letters?

It is 19.48 GMT on 9 July 2014. I only say this in case at this point in time there is a problem with your system. Reading my *LJ*, I followed the link to the new Letters "page" to find just two letters there. There always were more when they were included in the magazine (print and of course digital). Only two! I cannot believe that *LJ* readers have been that quiet.
—Roy Read

The past couple months, we've been experimenting with moving Letters to the Editor to the Web, or at least partially to the Web. We received a lot of feedback, and so we have put things back the way they used to

The White Paper Library on LinuxJournal.com

www.linuxjournal.com/whitepapers

be. We appreciate the feedback, and really do listen!—Shawn Powers

Dave Taylor's June 2014 Work the Shell Column

The first attempt at running Dave's script on Solaris 8 failed because the default Solaris shell in Solaris 10 and before does not support the `$()` syntax.

The issue after switching to /bin/bash appears to be likely a cut-and-paste error. The script works as is for me, but if you modify the first sed replacement so that the space is removed:

```
myPATH="$(echo $PATH | sed -e 's//~~/g' -e 's/:/ /g')"
```

Then the output is a sed error, which matches your published results:

```
First RE may not be null

0 commands, and 0 entries that weren't marked executable
```

—Brian

Dave Taylor replies: *Thanks for the update. It's been a while since I had access to a Solaris system!*

Bug in Dave Taylor's Days Between Dates Script

Dave's leap year test in the July 2014 issue contains a little bug. He uses:

```
leapyear=$(cal 2 $year | grep '29')
```

which looks for 29 in a calendar output. If 29 is present, it is supposed to be a leap year. However, if the user tries to find out if 2029 is a leap year, the script will fail, because it does contain 29 in the output, even though it can never be a leap year. A better version of this test would be:

```
leapyear=$(cal -h 2 $year | tail -n 2 | grep '29')
```

This will look for 29 only in the last two lines of the calendar. And, it also will omit the highlighting of today's date, just in case it is February 29 today. A highlighted 29 would not be found by grep.
—San Bergmans

Dave Taylor replies: *Another good solution to a problem that a number of people have pointed out, San!*

Here's my latest solution to that problem, one requiring GNU date:

```
leapyear=$(date -d 12/31/$1 +%j | grep 366)
```

A different approach, for sure! Thanks for writing.

Digital Encryption and a Random Bit Generator

A printable poster of an unbreakable

encryption algorithm declassified by the NSA can be downloaded from **http://www.tag.md/public**.

In addition, there is a POSIX C implementation of a digital non-deterministic random bit generator.
—Mark Knight

Cool stuff, thanks Mark!
—Shawn Powers

Comments on Dave Taylor's June 2014 Work the Shell Column

Solaris 8 was actually released in 2000, not 2004.
—Peter Schow

Dave Taylor replies: *Bah, and I looked it up on-line too. Thanks for the update. Now for the real question: are you still running it, Peter?*

Peter Schow replies: No, I run the latest-and-greatest Solaris, but there are stories about Solaris 7 (~1998) still in production out there.

Enjoy your column!

WRITE *LJ* A LETTER

We love hearing from our readers. Please send us your comments and feedback via http://www.linuxjournal.com/contact.

PHOTO OF THE MONTH
Remember, send your Linux-related photos to ljeditor@linuxjournal.com!

LINUX JOURNAL

At Your Service

SUBSCRIPTIONS: *Linux Journal* is available in a variety of digital formats, including PDF, .epub, .mobi and an on-line digital edition, as well as apps for iOS and Android devices. Renewing your subscription, changing your e-mail address for issue delivery, paying your invoice, viewing your account details or other subscription inquiries can be done instantly on-line: **http://www.linuxjournal.com/subs.** E-mail us at subs@linuxjournal.com or reach us via postal mail at *Linux Journal*, PO Box 980985, Houston, TX 77098 USA. Please remember to include your complete name and address when contacting us.

ACCESSING THE DIGITAL ARCHIVE: Your monthly download notifications will have links to the various formats and to the digital archive. To access the digital archive at any time, log in at **http://www.linuxjournal.com/digital**.

LETTERS TO THE EDITOR: We welcome your letters and encourage you to submit them at **http://www.linuxjournal.com/contact** or mail them to *Linux Journal, PO Box 980985,* Houston, TX 77098 USA. Letters may be edited for space and clarity.

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

FREE e-NEWSLETTERS: *Linux Journal* editors publish newsletters on both a weekly and monthly basis. Receive late-breaking news, technical tips and tricks, an inside look at upcoming issues and links to in-depth stories featured on **http://www.linuxjournal.com**. Subscribe for free today: **http://www.linuxjournal.com/enewsletters**.

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: **http://ww.linuxjournal.com/advertising**. Contact us directly for further information: ads@linuxjournal.com or +1 713-344-1956 ext. 2.

diff -u
WHAT'S NEW IN KERNEL DEVELOPMENT

Sometimes a new piece of code turns out to be more useful than its author suspected. **Alejandra Morales** recently came out with the **Cryogenic Project** as part of his Master's thesis, supervised by **Christian Grothoff**. The idea was to reduce energy consumption by scheduling input/output operations in batches.

This idea turned out to be so good that **H. Peter Anvin** didn't want Cryogenic to be a regular driver, he wanted it to be part of the core Linux input/output system. On the other hand, he also felt that the programmer interface needed to be cleaned up and made a bit sleeker.

Pavel Machek also was highly impressed and remarked that this could save power on devices like phones and tablets that were always running low. And, Christian confirmed that this was one of the main goals of the code.

Christian added that power savings seemed to be on the order of 10%, though that number could be tweaked up or down by increasing or decreasing the amount of delay that was tolerable for each data packet.

David Lang also liked Cryogenic and agreed that it should go into the core input/output system. He added that a lot of other folks had attempted to accomplish similar things. It was a highly desired feature in the kernel. David also pointed out that in order to get into the core input/output system, the Cryogenic code would have to demonstrate that it had no performance impact on code that did not use its features, or that the impact would be minimal.

Luis R. Rodriguez recently pointed out that a lot of drivers were routinely backported to a large array of older kernels, all the way down to version 2.6.24. And although he acknowledged that this was currently manageable, he expected the number of drivers and other backportable features to continue to increase, making the situation progressively more difficult to sustain.

Luis said the kernel folks should do more to educate users about the need to upgrade. But, he also wrote up a recommendation that the kernel folks use **Coccinelle** to automate the backporting process (**http://www.do-not-panic.com/2014/04/automatic-linux-kernel-backporting-with-coccinelle.html**).

Coccinelle is a tool used to transform source code programmatically. It can be used to generate changes to earlier kernel code to match the functionality provided by newer patches. That's so crazy, it just might work!

But to get started, Luis wanted to draw a line between kernels that would receive backports and kernels that would not. Hopefully, that line would only move forward. So he asked the linux-kernel mailing list members in general to tell him which were the earliest kernels they really needed to keep using.

As it turned out, **Arend van Spriel** knew of **Broadcom WLAN testers** that still relied on **Fedora 15**, running the 2.6.38 kernel. He said he was working with them to upgrade to **Fedora 19** and the 3.13 kernel, but that this hadn't happened yet.

So it appears that a certain amount of backporting will become automated, but of course, the Coccinelle transformations still would need to be written and maintained by someone, which is why Luis wanted to limit the number of target kernels.

It turns out **Windows** does certain things better than Linux—for example, in the area of rebooting. Apparently, there are several techniques that can be done in software to cause a system to reboot. But in some cases, the Linux system will go down successfully, and then not come up again. This is a problem, for example, in server farms with minimal human staff. If 20 systems are acting up and you want to reboot them all, it's easier to give a single command from a remote terminal than to send a human out into the noise and the cold to press each reset button by hand.

One rebooting technique involves sending certain values to the **0xCF9 port** on the system. Another is to use the **EFI** (Extensible Firmware Interface) BIOS replacement from **Intel**. Depending on the circumstances, one or the other rebooting technique is preferred, but the logic behind that selection can be tricky. In particular, changing the state of various pieces of hardware can change the appropriate reboot technique. So, if you run through a series of reboot attempts, and somehow change hardware state along the way, you can find that none of the attempts can succeed.

The cool thing about this particular bug is the straightforward way **Linus Torvalds** said that Windows must be doing something right, and that the Linux people needed to figure out what that was so Linux could do it right too.

Steven Rostedt pointed out the boot failure in one of his systems, and this triggered the bug hunt. Part of the problem is that it's very difficult to understand exactly what's going on with a system when it boots up. Strange magical forces are apparently invoked.

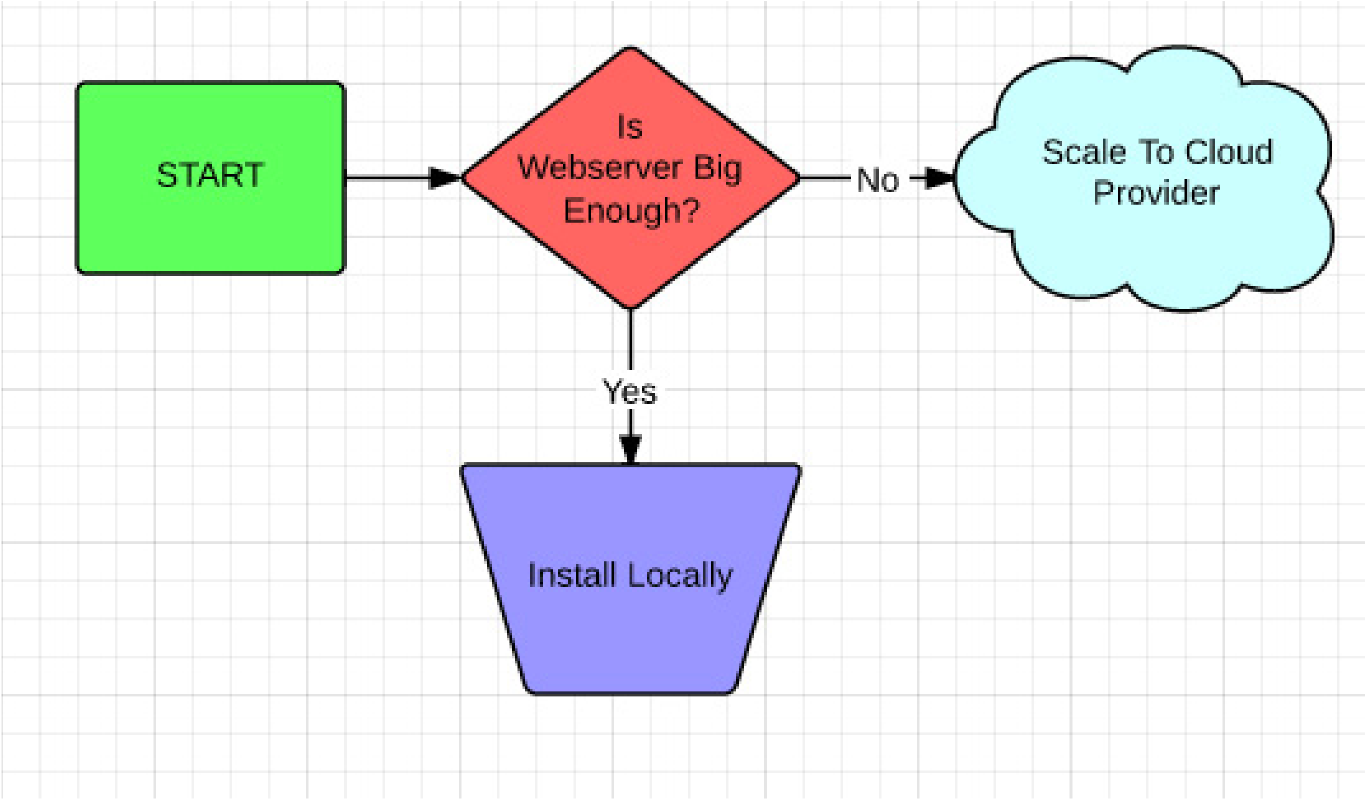
During the course of a somewhat heated debate, **Matthew Garrett** summed up what he felt was the underlying issue, and why the problem was so difficult to solve. In response to any of the various bootups attempted, he said, "for all we know the firmware is running huge quantities of code in response to any of those register accesses. We don't know what other hardware that code touches. We don't know what expectations it has. We don't know whether it was written by humans or written by some sort of simulated annealing mechanism that finally collapsed into a state where Windows rebooted."

Matthew was in favor of ditching the 0xCF9 bootup technique entirely. He argued, "We know that CF9 fixes some machines. We know that it breaks some machines. We don't know how many machines it fixes or how many machines it breaks. We don't know how many machines are flipped from a working state to a broken state whenever we fiddle with the order or introduce new heuristics. We don't know how many go from broken to working. The only way we can be reasonably certain that hardware will work is to duplicate precisely what Windows does, because that's all that most vendors will ever have tested."

But, Linus Torvalds felt that ditching CF9 was equivalent to flailing at the problem. In the course of discussion he said, "It would be interesting if somebody can figure out *exactly* what Windows does, because the fact that a lot of **Dell** machines need quirks almost certainly means that it's *us* doing something wrong. Dell doesn't generally do lots of fancy odd things. I pretty much guarantee it's because we've done something odd that Windows doesn't do."

The discussion had no resolution—probably because it's a really tough problem that hits only a relatively small number of systems. Apparently the bug hunt—and the debate—will continue.—**ZACK BROWN**

Lucidchart



I am a visual learner. When I try to teach something, I naturally like to use visual examples. That usually involves me working for hours to create flowcharts in Google Docs using the drawing program. Yes, it works, but it's a very cumbersome way to create a flowchart. Thankfully, I recently discovered Lucidchart (**http://www.lucidchart.com**).

Lucidchart is an on-line service that provides a free way to create flowcharts quickly and easily. Once the flowchart is created, items can be dragged around, resized and modified while still staying connected. I used Lucidchart for this month's Open-Source Classroom column, and if you need to create a quick flowchart, I recommend you give it a try as well.

Lucidchart provides its service for free, but there also are paid tiers that give you more options, such as importing and exporting Visio documents and so on. There certainly are other more powerful charting tools available, but few are as simple and quick to use.

—SHAWN POWERS

One Charger to Rule Them All

If you're anything like me, your nightstand is full of electronic devices that need to be charged regularly. Every night I have:

- Nexus 7 tablet.
- Cell phone.
- Kindle Paperwhite.
- iPad Air.
- Fitbit.

Granted they don't all need a daily charge, but the two tablets and cell phone certainly do. Although many of you are probably tsk'ing me for buying an iPad, for this purpose, it's a fine example of a device that is finicky about being charged. Many tablets, the iPad especially, require a lot of amperage to charge properly. Enter the Anker 40W, five-port USB charger.

Before buying the Anker, I had to get a power strip in order to plug in all the wall-warts required to charge my devices. Two of those devices (the Fitbit and Kindle) didn't even come with power adapters, just USB cables to plug in to a computer for charging. With the Anker USB charger, I'm able to use a single, regular-sized power cord to charge all my devices. Because it's designed specifically to charge, it has some great features as well:

- Dynamic, intelligently assigned amperage, up to 2.4 amps per port (8 amps max for all ports combined).
- Compact size (about the size of a deck of playing cards).
- Supports Apple, Android and other USB-based charging.

I've been using the Anker charger for several weeks and absolutely love it. There also is a 25 watt version if you don't need the full 40 watts, but I highly recommend getting the larger version, just in case you need more power in the future.

I purchased the charger on Amazon for $26, and although that's more than I'd normally pay for a USB charger, it's more like getting five chargers in one. Check it out at http://www.ianker.com/support-c7-g345.html.

—SHAWN POWERS

They Said It

It is necessary to try to surpass oneself always; this occupation ought to last as long as life.
—Queen Christina

We go where our vision is.
—Joseph Murphy

I didn't mind getting old when I was young. It's the being old now that's getting to me.
—John Scalzi

There's no such thing as quitting. Just sometimes there's a longer pause between relapses.
—Alan Moore

It is better to offer no excuse than a bad one.
—George Washington

OpenAxiom

Several computer algebra systems are available to Linux users. I even have looked at a few of them in this column, but for this issue, I discuss OpenAxiom (**http://www.open-axiom.org**). OpenAxiom actually is a fork of Axiom. Axiom originally was developed at IBM under the name ScratchPad. Development started in 1971, so Axiom is as old as I am, and almost as smart. In the 1990s, it was sold off to the Numerical Algorithms Group (NAG). In 2001, NAG removed it from commercial sale and released it as free software. Since then, it has forked into OpenAxiom and FriCAS. Axiom still is available. The system is specified in the book *AXIOM: the Scientific Computation System* by Richard Jenks and Robert Sutor. This book is available on-line at **http://wiki.axiom-developer.org/axiom-website/hyperdoc/axbook/book-contents.xhtml**, and it makes up the core documentation for OpenAxiom.

Most Linux distributions should have a package for OpenAxiom. For example, with Debian-based distributions, you can install OpenAxiom with:

```
sudo apt-get install openaxiom
```

If you want to build OpenAxiom from source, you need to have a Lisp engine installed. There are several to choose from on Linux, such as CLisp or GNU Common Lisp. Building is a straightforward:

```
./configure; make; make install
```

To use OpenAxiom, simply execute `open-axiom` on the command line. This will give you an interactive OpenAxiom session. If you have a script of commands you want to run as a complete unit, you can do so with:

```
open-axiom --script myfile.input
```

where the file "myfile.input" contains the OpenAxiom commands to be executed.

So, what can you actually do with OpenAxiom? OpenAxiom has many different data types. There are algebraic ones (like polynomials, matrices and power series) and data structures (like lists and dictionaries).

You can combine them into any reasonable combinations, like polynomials of matrices or matrices of polynomials. These data types are defined by programs in OpenAxiom. These data type programs also include the operations that can be applied to the particular data type. The entire system is polymorphic by design. You also can extend the entire data type system by writing your own data type programs. There are a large number of different numeric types to handle almost any type of operation as well.

The simplest use of OpenAxiom is as a calculator. For example, you can find the cosine of 1.2 with:

```
cos(1.2)
```

This will give you the result with 20 digits, by default. You can change the number of digits being used with the `digits()` function. OpenAxiom also will give you the type of this answer. This is useful when you are doing more experimental calculations in order to check your work. In the above example, the type would be `Float`. If you try this:

```
4/6
```

the result is `2/3`, and you will see a new type, `Fraction Integer`. If you have used a commercial system like Maple before, this should be familiar.

OpenAxiom has data types to try to keep results as exact values. If you have a reason to use a particular type, you can do a conversion with the `::` operator. So, you could redo the above division and get the answer as a float with:

```
(4/6)::Float
```

It even can go backward and calculate the closest fraction that matches a given float with the command:

```
%::Fraction Integer
```

The `%` character refers to the most recent result that you calculated. The answer you get from this command may not match the original fraction, due to various rounding errors.

There are functions that allow you to work with various parts of numbers. You can `round()` or `truncate()` floating-point numbers. You even can get just the fractional part with `fractionPart()`.

One slightly unique thing in OpenAxiom is a set of test functions. You can check for oddness and evenness with the functions `odd?()`

and even?(). You even can check whether a number is prime with prime?(). And, of course, you still have all of the standard functions, like the trigonometric ones, and the standard operators, like addition and multiplication.

OpenAxiom handles general expressions too. In order to use them, you need to assign them to a variable name. The assignment operator is :=. One thing to keep in mind is that this operator will execute whatever is on the right-hand side and assign the result to the name on the left-hand side. This may not be what you want to have happen. If so, you can use the delayed assignment operator ==. Let's say you want to calculate the square of some numbers. You can create an expression with:

```
xSquared := x**2
```

In order to use this expression, you need to use the eval function:

```
eval(xSquared, x=4)
```

You also can have multiple parameters in your expression. Say you wanted to calculate area. You could use something like this:

```
xyArea := x * y eval(xyArea, [x=2, y=10])
```

The last feature I want to look at in this article is how OpenAxiom handles data structures. The most basic data structure is a list. Lists in OpenAxiom are homogeneous, so all of the elements need to be the same data type. You define a list directly by putting a comma-separated group in square brackets—for example:

```
[1,2,3,4]
```

This can be done equivalently with the listfunction:

```
list(1,2,3,4)
```

You can put two lists together with the append function:

```
append([1,2],[3,4])
```

If you want to add a single element to the front of a list, you can use the cons function:

```
cons(1, [2,3,4])
```

List addressing is borrowed from the concepts in Lisp. So the most basic addressing functions to get elements are the functions first and rest. Using the basic list from above, the function:

```
first([1,2,3,4])
```

will return the number 1, and the function:

```
rest([1,2,3,4])
```

will return the list [2,3,4]. Using these functions and creative use of loops, you can get any element in a given list. But, this is very inconvenient, so OpenAxiom provides a simpler interface. If you had assigned the above list to the variable `mylist`, you could get the third element with:

```
mylist.3
```

or, equivalently:

```
mylist(3)
```

These index values are 1-based, as opposed to 0-based indexing in languages like C.

A really unique type of list available is the infinite list. Say you want to have a list of all integers. You can do that with:

```
myints := [i for i in 1..]
```

This list will contain all possible integers, and they are calculated only when you need the value in question. You can have more complicated examples, like a list of prime numbers:

```
myprimes := [i for i in 1.. | prime?(i)]
```

One issue with lists is that access times depend on how big the list is. Accessing the last element of a list varies, depending on how big said list is. This is because lists can vary in length. If you have a piece of code that deals with lists that won't change in length, you can improve performance by using an array. You can create a one-dimensional array with the function:

```
oneDimensionalArray([2,3,4,5])
```

This assigns a fixed area of memory to store the data, and access time now becomes uniform, regardless of the size of the list. Arrays also are used as the base data structure for strings and vectors. You even can create a bits data structure. You could create a group of eight 1-bits with:

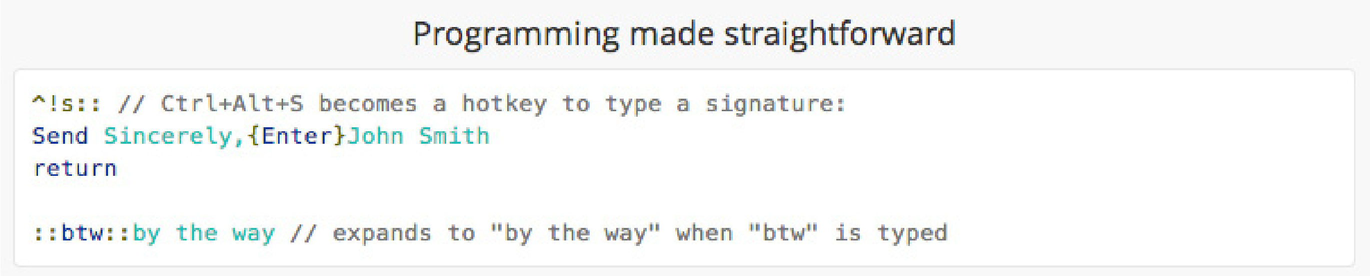
```
bits(8,true)
```

In this way, you can begin to do some rather abstract computational work.

As you have seen, OpenAxiom and its variants are very powerful systems for doing scientific computations. I covered only the very basic functionality available here, but it should give you a feeling for what you can do. In the near future, I plan to take another look at OpenAxiom and see what more advanced techniques are possible, including writing your own functions and programs.

—JOEY BERNARD

Non-Linux FOSS: AutoHotkey



Programming made straightforward

```
^!s:: // Ctrl+Alt+S becomes a hotkey to type a signature:
Send Sincerely,{Enter}John Smith
return

::btw::by the way // expands to "by the way" when "btw" is typed
```

Text expansion and hotkey automation are the sort of things you don't realize you need until you try them. Those of you who ever have played with system settings in order to change the function of a keystroke on your system will understand the value of custom hotkeys.

For Windows users, the customization of keystrokes is pretty limited with the system tools. Thankfully, the folks at **http://www.autohotkey.com** have created not only an incredible tool for creating scripted hotkeys, but they've also included automatic text expansion/replacement for speed boosts on the fly.

Programming the hotkeys and text replacements is pretty straightforward, and the Web site offers plenty of tutorials for making complex scripts for elaborate automation. Even if you just want to do a few simple hotkeys, however, AutoHotkey is a tool you really will want to check out. Best of all, it's completely open source, so there's no reason not to go download it today!

—SHAWN POWERS

The Best SharePoint Training in the World returns to Boston!

Choose from more than 80 classes and tutorials!

> "I really enjoyed it. I can hardly wait to get back to work and start using what I learned. I will encourage employees and co-workers to attend future SPTechCons. The conference had great speakers with relevant subjects, and the whole thing was well organized."
>
> —Greg Long, Infrastructure Development Manager, ITG, Inc.

> "I prefer SPTechCon over Microsoft's SharePoint Conference in Vegas. I'm definitely going to tell others to go."
>
> —Ray Ranson, Senior Architect, RSUI

SPTechCon

The SharePoint Technology Conference

September 16-19, 2014

The Boston Park Plaza Hotel & Towers

Bolster your career by becoming a SharePoint Master!

- Learn from SharePoint experts, including dozens of SharePoint MVPs and Certified SharePoint Professionals
- Master document management
- Study SharePoint governance
- Find out about SharePoint 2013
- Learn how to create applications for SharePoint that solve real business problems
- Exchange SharePoint tips and tricks with colleagues
- Test-drive SharePoint solutions in the Exhibit Hall

If you or your team needs Microsoft SharePoint training, come to SPTechCon Boston!

Register Early and SAVE! → www.sptechcon.com

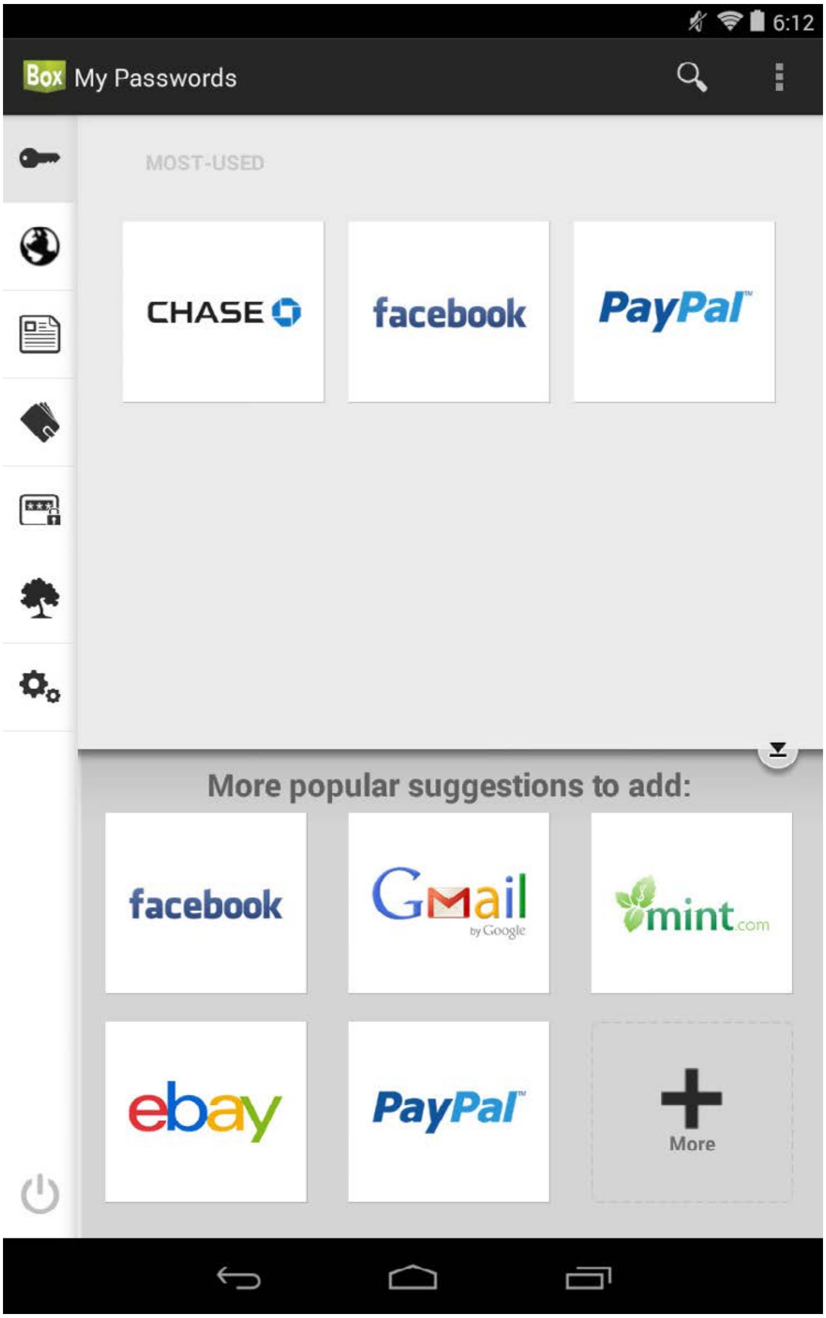
A BZ Media Event @SPTechCon Central Contractor Registration

SPTechCon™ is a trademark of BZ Media LLC. SharePoint® is a registered trademark of Microsoft.

Android Candy: Quit Thumbing Your Passwords!

I use my phone more often to log in to on-line accounts than I use a computer. I can assure you it's not because typing passwords on a tiny keyboard is fun. For most of us, we just have instant access to our phones at any given time during the day. The big problem with always using a tiny phone is that it means logging in to tiny Web sites (especially if there is no mobile version of the site) with tiny virtual keys and a long, complex password. It makes for real frustration.

With PasswordBox, you not only can store your user names and passwords, but also log in to those Web



sites with a single click. Once you authenticate with your master password to the PasswordBox app, it will allow you to create login profiles for dozens of sites and give you the ability to add entries for your own personal sites. If you want to log in to your bank with your phone, but don't want anyone to see you type in your banking credentials, PasswordBox is the perfect tool for you.

With great power comes great responsibility, and it's important to understand what PasswordBox allows you to do. When you initially launch it, you'll be prompted for how you want the application to handle when it locks your data and requires you to retype the master password. Ideally, this would be "immediately after you quit the app", but PasswordBox allows you to sacrifice security for convenience and will stay unlocked anywhere from 30 seconds to several hours. It even will let you rely on your Android lock screen for security and never prompt you for your master password!

Even with its potential for insecurity, PasswordBox is a powerful and convenient tool that makes using your phone much less burdensome when logging in to on-line services. In fact, it greatly can improve security as you won't need to type in your banking information in plain sight of the guy next to you at McDonald's. For those reasons, PasswordBox gets this month's Editors' Choice Award. Check it out today at http://www.passwordbox.com.

—SHAWN POWERS

LINUX JOURNAL
on your
Android device

Download the app now in the **Android Marketplace**

www.linuxjournal.com/android

Sinatra | HTML5 | Catalyst | Drupal 7 | Moodle
LINUX JOURNAL
INTRO TO THE SINATRA MICRO-FRAMEWORK
THE BEST WAY TO CLIENT CONCURRENCY
WEB DEVELOPMENT
CREATE Web Apps with Catalyst and Perl
NEW COLUMN: THE OPEN-SOURCE CLASSROOM

REUVEN M. LERNER

12-Factor Apps

Reuven describes an interesting perspective on scalable, maintainable Web apps.

I often tell the students in my programming classes that back in the 1960s and 1970s, it was enough for a program to run. In the 1980s and 1990s, we needed programs not only to run, but also to run quickly. In the modern era, programs need to run, and run quickly, but even more crucial is that they be maintainable.

Of course, we don't talk much about "programs" any more. Now we have "applications", or as Steve Jobs taught us to say, "apps". This is especially true for Web applications, which are a combination of many different programs, and often different languages as well—a server-side language, plus SQL, HTML, CSS and JavaScript. And, don't forget the configuration files, which can be in XML, YAML or other formats entirely.

Modern Web frameworks have tried to reduce the potential for clutter and chaos in Web applications. Ruby on Rails was the most prominent framework to suggest that we prefer "convention over configuration", meaning that developers should sacrifice some freedom in naming conventions and directory locations, if it means easier maintenance. And indeed, when I take over someone else's Rails codebase, the fact that the framework dictates the names and locations of many parts of the program reduces the time it takes for me to understand and begin improving the program.

Even in a Rails application though, we can expect to see many different files and programs. Heroku, a well-known hosting company for Web apps, looked at thousands of apps and tried to extract from them the factors that made it more likely that they would succeed. Their recommendations, written up by then-CTO Adam Wiggins, are known as the "12-factor app", and they describe practices that Heroku believes

The growth of Vagrant and Docker, two open-source systems that allow for easy virtualization and containers, means that we might see this aspect of the 12-factor app change, looking at "containers" rather than "repositories".

will make your app more maintainable and more likely to succeed.

In this article, I take a look at each of the factors of a 12-factor app, describing what they mean and how you can learn from them. I should note that not every aspect of a 12-factor app is unique to this set of recommendations; some of these practices have been advisable for some time. Moreover, it's important to see these as recommendations and food for thought, not religious doctrine. After all, these recommendations come from examining a large number of applications, but that doesn't mean they're a perfect match for your specific needs.

1. Codebase

A 12-factor app has "one codebase tracked in revision control, with many deploys". I remember the days before version control, in which we would modify Web applications on the production server. Obviously, things have improved a great deal since then, and many (most?) developers now understand the importance of keeping their code inside a Git repository.

So, why would it be important to state this as part of a 12-factor app?

It would seem that the reason is two-fold: keep everything you need for the application inside a single repository, and don't use the same repository for more than one app. In other words, there should be a one-to-one correspondence between your app and the repository in which it sits.

Following this advice means, among other things, that you can distribute your app multiple times. I recently finished a project for a client that originally had developed the software in Lotus Notes. Now, I don't know much about Notes, but the fact is that you cannot easily distribute an application written in Notes to new servers, let alone to your laptop. A 12-factor app puts everything inside the repository, dramatically reducing the work needed to deploy to a new server or new environment.

The growth of Vagrant and Docker, two open-source systems that allow for easy virtualization and containers, means that we might see this aspect of the 12-factor app change, looking at "containers" rather than "repositories". Indeed, Discourse already has moved in this direction, encouraging users to deploy within a Docker container, rather than installing the software themselves. However, the idea would be the same—have one configured version of the application and then deploy it many times.

2. Dependencies

Every program has dependencies; if nothing else, software depends on the language in which it was written, as well as the core libraries of that language. But if you are using an open-source language and framework, you likely are using numerous packages. A 12-factor app, according to the definition, explicitly declares and isolates dependencies. That is, the application should indicate what external libraries it uses and make it possible to change or remove those dependencies.

This factor does raise the question, at least for me, of "As opposed to what?" In Rails, for example, I cannot easily use a package (Ruby gem) without explicitly mentioning it my application's Gemfile. In Python, I need to import packages explicitly in files that use them. To what is the 12-factor author referring when he says that we shouldn't implicitly use external resources?

The author writes that apps should not "rely on the implicit existence of any system tools", going so far as to say that they should not "shell out" to run external programs. As a general rule, that's certainly a good idea; the days in which it was acceptable to open a subshell to run external UNIX utilities are long gone. And yet, there are times when it is necessary to do so.

So I have to wonder about the advice given in the 12-factor app, saying that all external programs should be bundled with the application, in its repository. It's a good idea to have everything in one place, but I recently worked on a project that needed to use the open-source PdfTk program. Did I really want to include PdfTk in my app's repository? I expect it wouldn't even work to do that, given the mix of Windows, Macintosh and Linux boxes among the developers, servers and project managers working on the site.

Another aspect of this factor has to do with making not only the library dependencies explicit, but also their versions. It's very easy and tempting simply to use whatever version of a Ruby gem or Python package is installed on a system. But without version numbers, this can cause huge problems—the application might behave differently on various computers, depending on what versions they have installed. Explicitly marking versions that are known to work reduces the chance of such trouble.

3. Configuration

It used to be that we would, in our code, explicitly connect to a database or to other servers from within our application. One of the many risks associated with that practice was that if others got a hold of our codebase, they then would have our credentials to log in to databases and third-party services.

A solution has been to put such information in configuration files that aren't committed into the code repository. How do such files then get to the system? That's actually a bit tricky, and there are a number of solutions to the problem.

One solution that has become popular, and is encouraged at **http://12factor.net**, is the use of environment variables. That is, all of the configuration is set in environment variables, in the deployment user's shell or elsewhere in the system (for example, the HTTP server's configuration file). If you don't want to set environment variables yourself, you can use a system like "dotenv" to provide them for you.

But it gets even better. By putting configuration settings in environment variables, you also ensure that you can run the same code in different environments. Want to change the credentials for a third-party service? You don't have to touch or redeploy the code itself. Want to connect to a different database when running your tests? Just change an environment variable.

Of all of the suggestions at 12factor.net, this is the one that I believe offers the most bang for the buck. It increases security and increases the flexibility of your application. The trick is to reduce, as much as possible, the number of "if" statements in your code that test for such values. You want to be using whatever the environment provides, not making conditional statements that change the behavior based on them.

4. Backing Services

The next factor at 12factor.net says that we should use external services as resources. This is a shift that has been happening more and more. For example, if you ran a Web app on a standalone server ten years ago, you would have just sent e-mail to your customers directly from the app, using the built-in "sendmail" (or equivalent) program.

However, the rise of spam filters and the intricacies of e-mail delivery, as well as the move toward server-oriented architectures (SOA), has encouraged people to move away from using their own systems and toward using separate, third-party systems. Sending e-mail, for example, can be done via Google's Gmail servers or even through a specialized company, such as Sendgrid.

Such third-party services have become increasingly common. This is true because networks have become faster and servers have become more reliable, but also because it's less and less justifiable for a company to spend money on an entire IT staff, when those functions can be outsourced to a third party.

In my opinion, such third-party services are an excellent way to create an application. It allows you to focus on the parts of your application that are special to your organization and not spend time or effort on the configuration and tuning of external services. However, this doesn't mean I'd want to outsource everything; in particular, I'm not sold on the idea of third-party database servers. Perhaps this is something I'll just have to get used to, but for now, there are certain aspects of my apps that I'll continue to keep in-house or at least on servers I've bought or rented.

5. Build, Release, Run

This factor says you first should build your app and then run it—something that I would think we normally do anyway. Certainly, deployment tools, such as Capistrano, have changed forever the way that I think about deploying apps. As I've started to experiment with such technologies as Chef, Puppet, Vagrant and Docker, I believe it's only a matter of time before we see an app as a self-contained, almost-compiled binary image that we then distribute to one or more servers. I have heard of a growing number of companies that not only use this approach, but that also deploy an entirely new Amazon EC2 server with each new release. If there is a problem with the code on a server, you just shut down the server and replace it with another one.

I'd generally agree that it's a bad idea to modify production code. However, I've certainly been in situations when it was necessary to do so, with particularly hard-to-track-down bugs that even a seemingly identical staging system could not change. Yes, that might point to problems with the testing environment, but when there is a crisis, my clients generally don't want to hear clever suggestions about the development and build process. Rather, they first want to fix the problem and then worry about how they can avoid the same problem in the future.

Nevertheless, if you aren't using a deployment tool to put your sites out on their servers, you might want to consider that.

6. Processes

The next factor says that the application should be one or more stateless processes. The main implication is that the application should be stateless, something I truly would hope has been the case for most Web apps for some time. And yet, when I was speaking with folks at a Fortune 500 company several days ago, asking about the scalability of an application that I'm building for them, they seemed genuinely surprised to hear that we could add as many Web servers as we wanted, because of the "share nothing" architecture.

Now, you do sometimes want to have cached memory or state for users or other resources. In such cases, it's best to use something like Memcached or Redis—or even a full-fledged relational database—to keep the state. This has the advantage of not only keeping it separate from your application, but also of sharing the data across all the Web servers you deploy.

7. Port Binding

This factor suggests that the application should be a self-contained system and, thus, export itself and its services via HTTP on a particular port. The idea here seems to be that every application should include an HTTP server of some sort and then start that server on a port. In this way, every Web application becomes a resource on the Internet, available via a URL and a port.

I must admit that this strikes me as a bit odd. Do I really want to see my HTTP server as part of my Web app? Probably not, so I don't see a good reason to include them together.

At the same time, I do see a strong advantage of considering each Web application as a small SOA-style resource, which can be queried and used from other applications. The entire Web is becoming an operating system, and the API calls of that operating system are growing all the time, as new URLs expose new services. By exposing your application as a resource, you are contributing to this new operating system and increasing the richness of the apps that run on it. However, I'm not convinced that where the HTTP server lies, and how closely it is bound to the app, really affects that mindset.

8. Concurrency

Those of us in the Linux world are fully familiar with the idea of creating new processes to take care of tasks. UNIX has a long history and tradition of using processes for multitasking, and while threads certainly exist, they're not the default way of scaling up.

The "concurrency" section of 12factor.net says that we should indeed use processes and not be afraid to spin up processes that will handle different aspects of our application. Each process then can run a specialized program, which communicates with the other processes using an API—be it a named pipe, socket, a database or even just the filesystem.

True, we could use threads for some of these things. But as 12factor.net says, threads cannot be moved to a different server or VM, whereas processes (especially if they don't contain state and save things to a common storage facility) can.

9. Disposability

This aspect of 12factor.net says that we can start up or shut down an app at any time, on any number of servers. To be honest, I'm not sure how this is different from the existing world of Web applications. For as long as I can remember, I was able to start up a Web app without too much fuss and take it down with even less trouble.

"Fast startup" is a good goal to have, but it can be hard to achieve, particularly with the overhead of many modern frameworks and languages. If you're using a language that sits on top of the Java virtual machine (JVM), you're likely to experience slow startup time but very fast execution time.

That said, I'm not sure how important it is to be able to start up a new copy of your application quickly, relative to other issues and constraints. True, it's frustrating to have slow startup times, particularly if those affect your ability to run a test suite. But most of the time, your application will be running, not starting up—thus, I'd downplay the importance of this particular factor.

10. Dev/Prod Parity

The idea behind this factor is extremely simple but also important: keep your development, staging and production environments as similar as possible.

It's hard to exaggerate the number of times I have experienced problems because of this. The system worked one way on my own development machine, another way on another programmer's machine, a third way on the staging server and a fourth way on the production machine. Such a situation is asking for trouble. It means that even if you have excellent test coverage of your application, you're likely to experience hard-to-debug problems that have to do with the configuration or the inherent differences between operating-system versions.

Once again, an increasingly popular solution to this problem is to use a virtual machine and/or a container. With a Vagrant VM, for example, you can share the same machine, not

Powerful: Rhino

Rhino M4800/M6800
- Dell Precision M6800 w/ Core i7 Quad (8 core)
- 15.6"-17.3" QHD+ LED w/ X@3200x1800
- NVidia Quadro K5100M
- 750 GB - 1 TB hard drive
- Up to 32 GB RAM (1866 MHz)
- DVD±RW or Blu-ray
- 802.11a/b/g/n
- Starts at $1375
- E6230, E6330, E6440, E6540 also available

- High performance NVidia 3-D on an QHD+ RGB/LED
- High performance Core i7 Quad CPUs, 32 GB RAM
- Ultimate configurability — choose your laptop's features
- One year Linux tech support — phone and email
- Three year manufacturer's on-site warranty
- Choice of pre-installed Linux distribution:

Tablet: Raven

Raven X240
- ThinkPad X240 by Lenovo
- 12.5" FHD LED w/ X@1920x1080
- 2.6-2.9 GHz Core i7
- Up to 16 GB RAM
- 180-256 GB SSD
- Starts at $1910
- W540, T440, T540 also available

Rugged: Tarantula

Tarantula CF-31
- Panasonic Toughbook CF-31
- Fully rugged MIL-SPEC-810G tested: drops, dust, moisture & more
- 13.1" XGA TouchScreen
- 2.4-2.8 GHz Core i5
- Up to 16 GB RAM
- 320-750 GB hard drive / 512 GB SSD
- CF-19, CF-52, CF-H2, FZ-G1 available

EmperorLinux
...where Linux & laptops converge

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

Model specifications and availability may vary.

just the same environment, among all developers and servers. Working in this way saves time and reliability, although it does admittedly have some negative effects on the performance of the system.

11. Logs

I love logs and sending data to them. In a small application, it's enough to have a single logfile on the filesystem. But if you're using a read-only system (such as Heroku), or if you are going to create and remove servers on a regular basis with Chef or Puppet, or if you have multiple servers, you will likely want to have logs as an external service.

Now, old-time UNIX users might say that syslog is a good solution for this. And indeed, syslog is fairly flexible and allows you to use one system as the logging server, with the others acting as clients.

The 12-factor suggestion is to go one step further than this, treating a log as a writable event stream to which you send all of your data. Where does it go? It might be syslog, but it's more likely going to be a third-party service, which will allow you to search and filter through the logs more easily than would be possible in simple text files.

I must admit there's still some comfort in my being able to run a `tail -f` on a textual logfile or `grep` on a file that's of interest to me. But I have used some third-party logging solutions, such as Papertrail, and have come away impressed. There also are open-source solutions, such as Greylog2, which some of my clients have used to great satisfaction.

12. Admin Processes

The final factor in a 12-factor app is that of administrative processes. Now, I often compare a Web app to a hotel, in that the end user sees only the minority of the actual workings. Just as guests never see the kitchen, laundry or administrative offices of a hotel, users of a Web app never see the administrative pages, which often can be extensive, powerful and important.

However, the 12-factor app prescription for admin processes isn't about administrative parts of the site. Rather, it's about the administrative tasks you need to do, such as updating the database. This factor says that you really should have a REPL (that is, a read-eval-print loop, aka an interactive shell), and that you can put many administrative tasks into small programs that execute.

I agree that an REPL is one of the most powerful and important aspects of the languages I typically use. And I love database migrations, as opposed to manual tinkering with the database that always can lead to problems. However, I'm not sure if this warrants inclusion as a crucial component of a maintainable Web application.

Conclusion

I see the 12-factor app as a great way to think about Web applications and often to increase their stability and maintainability. In some ways, I see it as similar to design patterns, in that we have gained a language that allows us to communicate with others about our technical design in a way that all can agree on. However, as with design patterns, it's important to see this as a tool, not a religion. Consider your needs, take the 12-factor app prescriptions into account, and apply as necessary. If all goes well, your app will end up being more scalable, reliable and maintainable.■

Reuven M. Lerner is a Web developer, consultant and trainer. He recently completed his PhD in Learning Sciences from Northwestern University. You can read his blog, Twitter feed and newsletter at http://lerner.co.il. Reuven lives with his wife and three children in Modi'in, Israel.

OLPC | Arduino | Trim-Slice | vDSO | Elmer | MPD
LINUX JOURNAL
ONE LAPTOP PER CHILD STATUS REPORT
USE YOUR PHONE TO LOCK YOUR SCREEN AUTOMATICALLY
HACK THIS
HOW TO: ADD A CUSTOM vDSO TO A LINUX KERNEL
Get Started Programming the Arduino
Create a Unified Boot Environment on a USB Drive
NETWORKING

LINUX JOURNAL now available for the **iPad** and **iPhone** at the App Store.

Available on the App Store

linuxjournal.com/ios

For more information about advertising opportunities within *Linux Journal* iPhone, iPad and Android apps, contact John Grogan at +1-713-344-1956 x2 or ads@linuxjournal.com.

DAVE TAYLOR

Days Between Dates: a Smarter Way

How many days have elapsed? Our intrepid shell script programmer Dave Taylor looks at how to solve date-related calculations and demonstrates a script that calculates elapsed days between any specified day in the past and the current date.

In case you haven't been reading my past few columns or, perhaps, are working for the NSA and scanning this content to identify key phrases, like "back door" and "low-level vulnerability", we're working on a shell script that calculates the number of days between a specified date in the past and the current date.

When last we scripted, the basic functionality was coded so that the script would calculate days from the beginning date to the end of that particular year, then the number of years until the current year (accounting for leap years), followed by the current number of days into the year. The problem is, it's not working.

Here's the state of things:

```
$ date
Mon Jul  7 09:14:37 PDT 2014
$ sh daysago.sh 7 4 2012
The date you specified -- 7-4-2012 -- is valid. Continuing...
0 days transpired between end of 2012 and beginning of this year
calculated 153 days left in the specified year
Calculated that 7/4/2012 was 341 days ago.
```

The script correctly ascertains that the current date, July 7, 2014, is 153 days into the year, but the rest of it's a hopeless muddle. Let's dig in to the code and see what's going on!

Two Versions of date = Not Good

The code in my last article was fairly

Did you try that and find that `date` complained about the `-j` flag? Good.

convoluted in terms of calculating the number of days left in the starting year subsequent to the date specified (July 4, 2012, in the above example), but there's a far easier way, highlighted in this interaction:

```
$ date -j 0803120010
Tue Aug  3 12:00:00 PDT 2010
$ date -j 0803120010 +%j
215
```

In other words, modern `date` commands let you specify a date (in MON DAY HOUR MIN YEAR format) and then use the `%j` format notation to get the day-of-the-year for that specific date. August 3, 2010, was the 215th day of the year.

Did you try that and find that `date` complained about the `-j` flag? Good. That means you're likely using GNU `date`, which is far superior and is actually something we'll need for the full script to work. Test which version you have by using the `--version` flag:

```
$ date --version
date (GNU coreutils) 8.4
Copyright (C) 2010 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later
<http://gnu.org/licenses/gpl.html>.
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.
Written by David MacKenzie.
```

How many days were in a given year? That's also easily done with a shortcut, checking the day of the year of December 31st. For example:

```
$ date -d 12/31/2010 +%j
365
```

But, 2012 was a leap year. So:

```
$ date -d 12/31/2012 +%j
366
```

Therefore, the days-left-in-year calculation is simply days-in-year – day-of-the-year.

The next calculation is days/year * years between the specified date and the current year.

Days Left in Year

The first step of calculating the days left in the starting year is to create the correct date format string for the `date` command. Fortunately, with

GNU date, it's easily done:

```
# build date string format for the specified starting date
startdatefmt="$startmon/$startday/$startyear"
calculate="$(date -d "12/31/$startyear" +%j) - $(date -d
 ➥$startdatefmt +%j)"
echo "Calculating $calculate"
daysleftinyear=$(( $calculate ))
```

When run as part of the script, the debugging echo statement offers useful information to help debug things:

```
$ sh daysago.sh 2 4 2012
The date you specified -- 2-4-2012 -- is valid. Continuing...
Calculating 366 - 035
There were 337 days left in the starting year
$ sh daysago.sh 2 4 2013
The date you specified -- 2-4-2013 -- is valid. Continuing...
Calculating 365 - 035
There were 336 days left in the starting year
```

Notice that when we specified Feb 4, 2012, a leap year, there are 337 days left in the year, but when we specify the same date on the following non-leapyear, there are 336 days. Sounds right!

Days in Intervening Years

The next portion of the calculation is to calculate the number of days in each year between the start year and the current year, not counting either of those. Sounds like a `while` loop to me, so let's do this:

```
daysbetweenyears=0
tempyear=$(( $startyear + 1 ))
while [ $tempyear -lt $thisyear ] ; do
  echo "intervening year: $tempyear"
  daysbetweenyears=$(( $daysbetweenyears + $(date -d
   ➥"12/31/$tempyear" +%j) ))
  tempyear=$(( $tempyear + 1 ))
done
echo "Intervening days in years between $startyear and
 ➥$thisyear = $daysbetweenyears"
```

Again, I'm adding a debugging `echo` statement to clarify what's going on, but we're getting pretty close:

```
$ sh daysago.sh 2 4 2010
The date you specified -- 2-4-2010 -- is valid. Continuing...
Calculating 365 - 035
Calculated that there were 336 days left in the starting year
intervening year: 2011
intervening year: 2012
intervening year: 2013
intervening days in years between 2010 and 2014 = 1096
```

That seems to pass the reasonable test, as there are three years between 2010 and 2014 (namely 2011, 2012 and 2013) and the back-of-envelope calculation of 3*365 = 1095.

If you're using non-GNU `date`, you already have realized that the string format is different and that you need to use the `-j` flag instead of the `-d` flag.

The problem is, the older date command also works differently, because 1969 is the beginning of Epoch time.

The problem is, the older `date` command also works differently, because 1969 is the beginning of Epoch time. Look:

```
$ date -j 0204120060  # last two digits are year, so '60
Wed Feb  4 12:00:00 PST 2060
```

It interprets the two-digit "60" as 2060, not 1960. Boo!

If you're not running GNU `date`, you're going to have a really big problem for dates prior to 1969, and I'm just going to say "get GNU date, dude" to solve it.

Total Days That Have Passed

We have everything we need to do the math now. We can calculate the number of days left in the start year, the number of days in intervening years, and the day-of-year number of the current date. Let's put it all together:

```
###   DAYS IN CURRENT YEAR
dayofyear=$(date +%j)            # that's easy!
###   NOW LET'S ADD IT ALL UP
totaldays=$(( $daysleftinyear + $daysbetweenyears + $dayofyear ))
echo "$totaldays days have elapsed between
 ➥$startmon/$startday/$startyear and today,
 ➥day $dayofyear of $thisyear."
```

That's it. Now, stripping out the debug echo statements, here's what we can ascertain:

```
$ sh daysago.sh 2 4 1949
23900 days have elapsed between 2/4/1949 and today,
 ➥day 188 of 2014.
$ sh daysago.sh 2 4 1998
6003 days have elapsed between 2/4/1998 and today,
 ➥day 188 of 2014.
$ sh daysago.sh 2 4 2013
524 days have elapsed between 2/4/2013 and today,
 ➥day 188 of 2014.
```

But look, there's still a lurking problem when it's the same year that we're calculating:

```
$ sh daysago.sh 2 4 2014
524 days have elapsed between 2/4/2014 and today,
 ➥day 188 of 2014.
```

That's a pretty easy edge case to debug, however, so I'm going to wrap things up here and let you enjoy trying to figure out what's not quite right in the resultant script.

Stumped? Send me an e-mail via http://www.linuxjournal.com/contact.■

Listing 1. Full daysago.sh Script

```
#!/bin/sh

# valid-date lite

function daysInMonth
{
  case $1 in
    1|3|5|7|8|10|12 ) dim=31 ;; # most common value
    4|6|9|11        ) dim=30 ;;
    2               ) dim=29 ;;  # depending on if it's a leap year
    *               ) dim=-1 ;;  # unknown month
  esac
}

function isleap
{
  leapyear=$(cal 2 $1 | grep '29')
}


if [ $# -ne 3 ] ; then
  echo "Usage: $(basename $0) mon day year"
  echo "  with just numerical values (ex: 7 7 1776)"
  exit 1
fi

eval $(date "+thismon=%m;thisday=%d;thisyear=%Y;dayofyear=%j")

startmon=$1; startday=$2; startyear=$3

daysInMonth $startmon                    # sets global var dim

if [ $startday -lt 0 -o $startday -gt $dim ] ; then
  echo "Invalid date: Month #$startmon has $dim days,
   ➥so day $startday is impossible."
  exit 1
fi

if [ $startmon -eq 2 -a $startday -eq 29 ] ; then
  isleap $startyear
  if [ -z "$leapyear" ] ; then
    echo "$startyear wasn't a leapyear, so February
     ➥only had 28 days."
    exit 1
  fi
fi

###################################
## Now part two: the number of days since the day you specify.
###################################

###  FIRST, DAYS LEFT IN START YEAR

# calculate the date string format for the specified starting date

startdatefmt="$startmon/$startday/$startyear"

calculate="$(date -d "12/31/$startyear" +%j) - $(date -d
 ➥$startdatefmt +%j)"

daysleftinyear=$(( $calculate ))

###  DAYS IN INTERVENING YEARS

daysbetweenyears=0
tempyear=$(( $startyear + 1 ))

while [ $tempyear -lt $thisyear ] ; do
  daysbetweenyears=$(( $daysbetweenyears + $(date
   ➥-d "12/31/$tempyear" +%j) ))
  tempyear=$(( $tempyear + 1 ))
done

###   DAYS IN CURRENT YEAR

dayofyear=$(date +%j)     # that's easy!

###   NOW LET'S ADD IT ALL UP

totaldays=$(( $daysleftinyear + $daysbetweenyears + $dayofyear ))

echo "$totaldays days have elapsed between
 ➥$startmon/$startday/$startyear and today,
 ➥day $dayofyear of $thisyear."

exit 0
```

Dave Taylor has been hacking shell scripts for more than 30 years. Really. He's the author of the popular *Wicked Cool Shell Scripts* and can be found on Twitter as @DaveTaylor and more generally at his tech site http://www.AskDaveTaylor.com.

Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.

Android Everywhere!

Android is everywhere! Come to AnDevCon and learn how to develop apps for the next generation of connected devices and how to create a seamless user experience. Learn about Android Wear, Android L, Android TV and more!

Get the best Android developer training anywhere!

- Choose from more than 80 classes and tutorials
- Network with expert speakers and other Android developers
- Check out more than 50 exhibiting companies

"AnDevCon is a great opportunity to take your Android skills to the next level, get exposed to technologies you haven't touched yet, and to network with some of the best Android developers in the world."
—Joe Mitchell, Software Engineer, Quicken Loans

Learn more at www.AnDevCon.com

AnDevCon
The Android Developer Conference
San Francisco Bay Area
November 18-21, 2014

A BZ Media Event

AnDevCon™ is a trademark of BZ Media LLC. Android™ is a trademark of Google Inc. Google's Android Robot is used under terms of the Creative Commons 3.0 Attribution License.

KYLE RANKIN

Check Exchange from the Command Line

When fetchmail can't fetch mail, it's time to fall back to raw command-line commands.

Through the years, you tend to accumulate a suite of tools, practices and settings as you use Linux. In my case, this has meant a Mutt configuration that fits me like a tailored suit and a screen session that at home reconnects me to my IRC session and at work provides me with quick access to e-mail with notifications along the bottom of the terminal for Nagios alerts and incoming e-mail. I've written about all of these different tools in this column through years, but in this article, I talk about how I adapted when one of my scripts no longer worked.

My e-mail notification script is relatively straightforward. I configure fetchmail on my local machine, but instead of actually grabbing e-mail, I just run `fetchmail -c`, which returns each mailbox along with how many messages are unseen. I parse that, and if I have any unread mail, I display it in the notification area in screen. I've written about that before in my February 2011 Hack and / column "Status Messages in Screen" (**http://www.linuxjournal.com/article/10950**), and up until now, it has worked well for me. Whenever I set up my computer for a new job, I just configure fetchmail and reuse the same script.

Recently, however, we switched our mail servers at work to a central Exchange setup, which by itself wouldn't be too much of an issue—in the past I just configured Mutt and fetchmail to treat it like any other IMAP host—but in this case, the Exchange server was configured with security in mind. So in addition to using IMAPS, each client was given a client certificate to present to the server during authentication. Mutt was able to handle this just fine with a few configuration tweaks, but fetchmail didn't fare so well. It turns out that fetchmail has what some would call a configuration quirk and others would call a bug. When you configure fetchmail to use a client certificate, it overrides whatever user name you have configured in favor of the user specified inside the client certificate. In my case, the two didn't match, so fetchmail wasn't able to log in to the Exchange server, and I no longer got new mail notifications inside my screen session.

I put up with this for a week or so, until I realized I really missed knowing when I had new e-mail while I was working. I decided there must be some other way to get a count of unread messages from the command line, so I started doing research. In the end, what worked for me was to use OpenSSL's `s_client` mode to handle the SSL session between me and the Exchange server (including the client certificate), and then once that session was established, I was able to send raw IMAP commands to authenticate and then check for unread messages.

OpenSSL s_client

The first step was to set up an OpenSSL `s_client` connection. Most people probably interact with OpenSSL on the command line only when they need to generate new self-signed certificates or read data from inside a certificate, but the tool also provides an `s_client` mode that you can use to troubleshoot SSL-enabled services like HTTPS. With `s_client`, you initiate an SSL connection and after it outputs relevant information about that SSL connection, you are presented with a prompt just as though you used Telnet or Netcat to connect to a remote port. From there, you can type in raw HTTP, SMTP or IMAP commands depending on your service.

The syntax for `s_client` is relatively straightforward, and here is how I connected to my Exchange server over IMAPS:

```
$ openssl s_client -cert /home/kyle/.mutt/imaps_cert.pem
➥-crlf -connect imaps.example.com:993
```

The `-cert` argument takes a full path to my client certificate file, which I store with the rest of my Mutt configuration. The `-crlf` option makes sure that I send the right line feed characters each time I press enter—important for

`expect` allows you to construct incredibly complicated programs that look for certain output and then send your input.

some touchy IMAPS servers. Finally the `-connect` argument lets me specify the hostname and port to which to connect.

Once you connect, you will see a lot of SSL output, including the certificate the server presents, and finally, you will see a prompt like the following:

```
* OK The Microsoft Exchange IMAP4 service is ready.
```

From here, you use the `tag login` IMAP command followed by your user name and password to log in, and you should get back some sort of confirmation if login succeeded:

```
tag login kyle.rankin supersecretpassword
tag OK LOGIN completed.
```

Now that you're logged in, you can send whatever other IMAP commands you want, including some that would show you a list of mailboxes, e-mail headers or even the full contents of messages. In my case though, I just want to see the number of unseen messages in my INBOX, so I use the `tag STATUS` command followed by the mailbox and then `(UNSEEN)` to tell it to return the number of unseen messages:

```
tag STATUS INBOX (UNSEEN)
* STATUS INBOX (UNSEEN 1)
tag OK STATUS completed.
```

In this example, I have one unread message in my INBOX. Now that I have that information, I can type `tag LOGOUT` to log out.

expect

Now this is great, except I'm not going to go through all of those steps every time I want to check for new mail. What I need to do is automate this. Unfortunately, my attempts just to pass the commands I wanted as input didn't work so well, because I needed to pause between commands for the remote server to accept the previous command. When you are in a situation like this, a tool like `expect` is one of the common ways to handle it. `expect` allows you to construct incredibly complicated programs that look for certain output and then send your input. In my case, I just needed a few simple commands: 1) confirm Exchange was ready; 2) send my login; 3) once I was authenticated, send the `tag STATUS` command; 4) then finally log out. The `expect` script turned into the following:

```
set timeout 10
spawn openssl s_client -cert /home/kyle/.mutt/imaps_cert.pem
 ➥-crlf -connect imaps.example.com:993
expect "* OK"
send "tag login kyle.rankin supersecretpassword\n"
expect "tag OK LOGIN completed."
sleep 1
send "tag STATUS INBOX (UNSEEN)\n"
expect "tag OK"
send "tag LOGOUT\n"
```

I saved that to a local file (and made sure only my user could read it) and then called it as the sole argument to `expect`:

```
$ expect .imapsexpectscript
```

Of course, since this script runs through the whole IMAPS session, it also outputs my authentication information to the screen. I need only the INBOX status output anyway, so I just `grep` for that:

```
$ expect ~/.imapsexpectscript | egrep '\(UNSEEN [0-9]'
* STATUS INBOX (UNSEEN 1)
```

For my screen session, I just want the name of the mailbox and the number of read messages (and no output if there are no unread messages), so I modify my `egrep` slightly and pipe the whole thing to a quick Perl one-liner to strip output I don't want. The final script looks like this:

```
#!/bin/bash

MAILCOUNT=`expect ~/.imapsexpectscript | egrep '\(UNSEEN [1-9]'
 ➥| perl -pe 's/.*STATUS \w+.*?(\d+)\).*?$/$1/'`
if [ "$MAILCOUNT" != "" ]; then
 echo INBOX:${MAILCOUNT}
fi
```

Now I can just update my .screenrc to load the output of that script into one of my backtick fields instead of fetchmail (more on that in my previous column about screen), and I'm back in business.■

Kyle Rankin is a Sr. Systems Administrator 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.

Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.

SHAWN POWERS

DNSMasq, the Pint-Sized Super Dæmon!

What's better than a DNS server, a DHCP server and a TFTP server? A single dæmon that does it all!

I've always been a fan of putting aftermarket firmware on consumer-grade routers. Whether it's DD-WRT, Tomato, OpenWRT or whatever your favorite flavor of "better than stock" firmware might be, it just makes economic sense. Unfortunately, my routing needs have surpassed my trusty Linksys router. Although I could certainly buy a several-hundred-dollar, business-class router, I really don't like spending money like that. Thankfully, I found an incredible little router (the EdgeRouter Lite) that can route a million packets per second and has three gigabit Ethernet ports. So far, it's an incredible router, but that's all it does—route. Which brings me to the point of this article.

I've always used the DHCP and DNS server built in to DD-WRT to serve my network. I like having those two services tied to the router, because if every other server on my network fails, I still can get on-line. I figure the next best thing is to have a Raspberry Pi dedicated to those services. Because all my RPi devices

Figure 1. The Cubox is more powerful than a Raspberry Pi, but even an RPi is more power than DNSMasq requires!

currently are attached to televisions around the house (running XBMC), I decided to enlist the Cubox computer I reviewed in November 2013 (Figure 1). It's been sitting on my shelf collecting dust, and I'd rather have it do something useful.

Although the Cubox certainly is powerful enough to run BIND and the ISC DHCP server, that's really overkill for my network. Plus, BIND really annoys me with its serial-number incrementation and such whenever an update is made. It wasn't until I started to research alternate DNS servers that I realized just how powerful DNSMasq can be. Plus, the way it works is simplicity at its finest. First, let's look at its features:

- Extremely small memory and CPU footprint: I knew this was the case, because it's the program that runs on Linux-based consumer routers where memory and CPU are at a premium.

- DNS server: DNSMasq approaches DNS in a different way from the traditional BIND dæmon. It doesn't offer the complexity of domain transfers, master/slave relationships and so on. It does offer extremely simple and highly configurable options that are, in my opinion, far more useful in a small- to medium-size network. It even does reverse DNS (PTR records) automatically! (More on those details later.)

- DHCP server: where the DNS portion of DNSMasq lacks in certain advanced features, the DHCP services offered actually are extremely robust. Most routers running firmware like DD-WRT don't offer a Web interface to the advanced features DNSMasq provides, but it rivals and even surpasses some of the standalone DHCP servers.

- TFTP server: working in perfect tandem with the advanced features of DHCP, DNSMasq even offers a built-in TFTP server for things like booting thin clients or sending configuration files.

- A single configuration file: it's possible to use multiple configuration files, and I even recommend it for clarity's sake. In the end, however, DNSMasq requires you to edit only a single configuration file to manage all of its powerful services. That configuration file also is very well commented, which makes using it much nicer.

Installation

DNSMasq has been around for a very long time. Installing it on any Linux operating system should be as simple as searching for it in your distribution's package management system. On Debian-based systems, that would mean something like:

```
sudo apt-get install dnsmasq
```

Or, on a Red Hat/CentOS system:

```
yum install dnsmasq (as root)
```

The configuration file (there's just one!) is usually stored at /etc/dnsmasq.conf, and as I mentioned earlier, it is very well commented. Figuring out even the most advanced features is usually as easy as reading the configuration file and un-commenting those directives you want to enable. There are even examples for those directives that require you to enter information specific to your environment.

After the dnsmasq package is installed, it most likely will get started automatically. From that point on, any time you make changes to the configuration (or make changes to the /etc/hosts file), you'll need to restart the service or send an HUP signal to the dæmon. I recommend using the init script to do that:

```
sudo service dnsmasq restart
```

But, if your system doesn't have the init service set up for DNSMasq, you can issue an HUP signal by typing something like this:

```
sudo kill -HUP $(pidof dnsmasq)
```

This will find the PID (process ID) and send the signal to reload its configuration files. Either way should work, but the init script will give you more feedback if there are errors.

First Up: DNS

Of all the features DNSMasq offers, I find its DNS services to be the most useful and awesome. You get the full functionality of your upstream DNS server (usually provided by your ISP), while seamlessly integrating DNS records for you own network.
To accomplish that "split DNS"-type setup with BIND, you need to create a fake DNS master file, and even then you run into problems if you are missing a DNS name in your local master file, because BIND won't query another server by default for records it thinks it's in charge of serving. DNSMasq, on the other hand, follows a very simple procedure

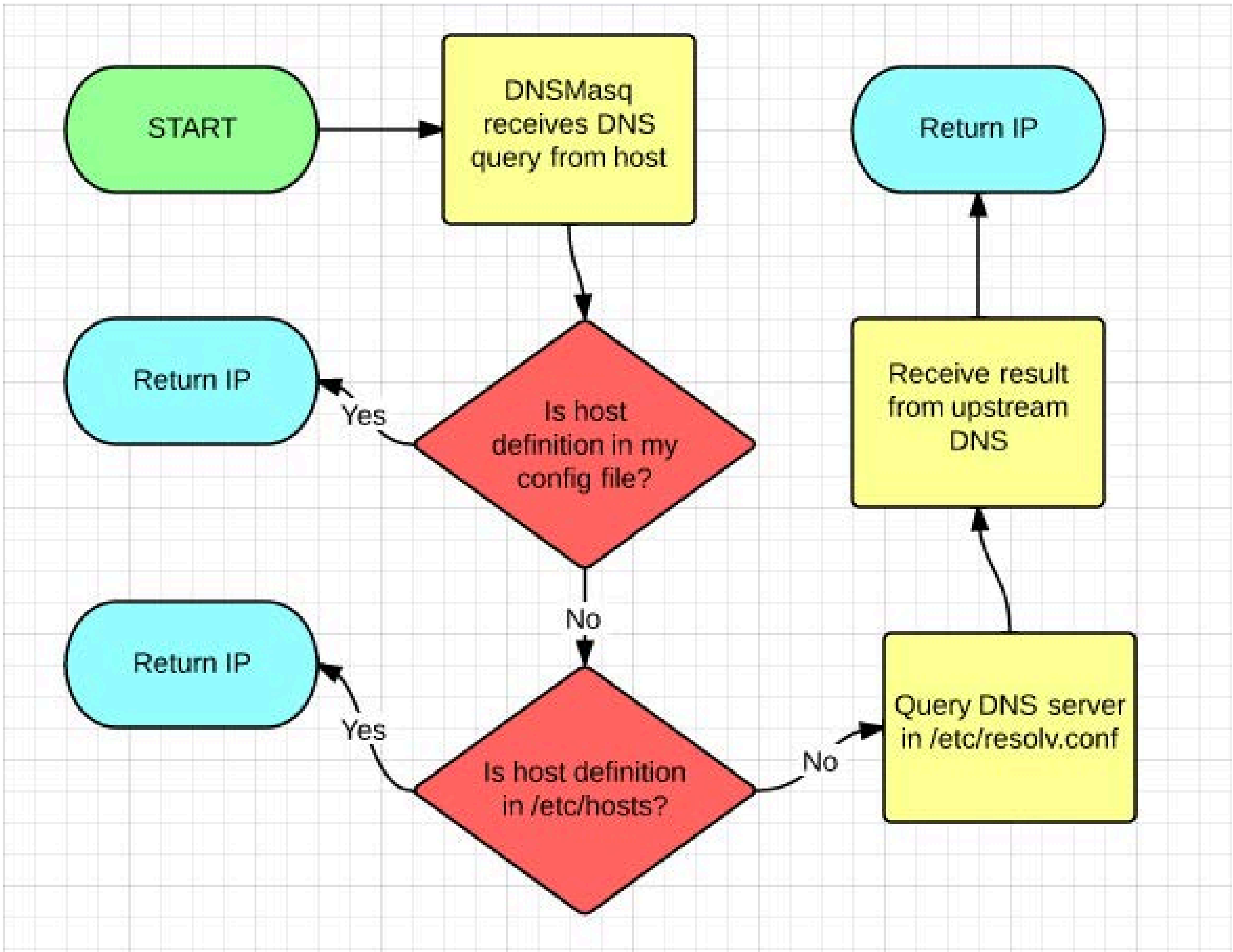

START
DNSMasq receives DNS query from host
Return IP
Is host definition in my config file?
Yes
Return IP
No
Receive result from upstream DNS
Return IP
Yes
Is host definition in /etc/hosts?
No
Query DNS server in /etc/resolv.conf

Figure 2. DNSMasq makes DNS queries simple, flexible and highly configurable.

when it receives a request. Figure 2 shows that process.

For my purposes, this means I can put a single entry into my server's /etc/hosts file for something like "server.brainofshawn.com", and DNSMasq will return the IP address in the /etc/hosts file. If a host queries DNSMasq for an entry not in the server's /etc/hosts file, www.brainofshawn.com for instance, it will query the upstream DNS server and return the live IP for my Web host. DNSMasq makes a split-DNS scenario extremely easy to maintain, and because it uses the server's /etc/hosts file, it's simple to modify entries.

My personal favorite feature of DNSMasq's DNS service, however, is that it supports round-robin load balancing. This isn't something that normally works with an /etc/hosts file entry, but with DNSMasq, it does. Say you have two entries in your /etc/hosts file like this:

```
192.168.1.10      webserver.example.com
192.168.1.11      webserver.example.com
```

On a regular system (that is, if you put it in your client's /etc/hosts file), the DNS query always will return 192.168.1.10 first. DNSMasq, on the other hand, will see those two entries and mix up their order every time it's queried. So instead of 192.168.1.10 being the first IP, half of the time, it will return 192.168.1.11 as the first IP. It's a very rudimentary form of load balancing, but it's a feature most people don't know exists!

Finally, DNSMasq automatically will create and serve reverse DNS responses based on entries found in the server's /etc/hosts file. In the previous example, running the command:

```
dig -x 192.168.1.10
```

would get the response "webserver.example.com" as the reverse DNS lookup. If you have multiple DNS entries for a single IP address, DNSMasq uses the first entry as the reverse DNS response. So if you have a line like this in your server's /etc/hosts file:

```
192.168.1.15 www.example.com mail.example.com ftp.example.com
```

Any regular DNS queries for www.example.com, mail.example.com or ftp.example.com will get answered with "192.168.1.15", but a reverse DNS lookup on 192.168.1.15 will get the single response "www.example.com".

DNSMasq is so flexible and feature-rich, it's hard to find a reason not to use it. Sure, there are valid reasons for using a more robust DNS server like BIND, but for most small to medium networks, DNSMasq is far more appropriate and much, much simpler.

Serving DHCP

It's possible to use DNSMasq for DNS only and disable the DHCP services it offers. Much like DNS, however, the simplicity and power offered by DNSMasq makes it a perfect candidate for small- to medium-sized networks. It supports both dynamic ranges for automatic IP assignment and static reservations based on the MAC address of computers on your network. Plus, since it also acts as the DNS server for your network, it has a really great hostname-DNS integration for computers connected to your network that may not have a DNS entry. How does that work?

Figure 3 shows the modified method used when the DNS server receives a query if it's also serving as a DHCP server. (The extra step is shown as the orange-colored diamond in the flowchart.)

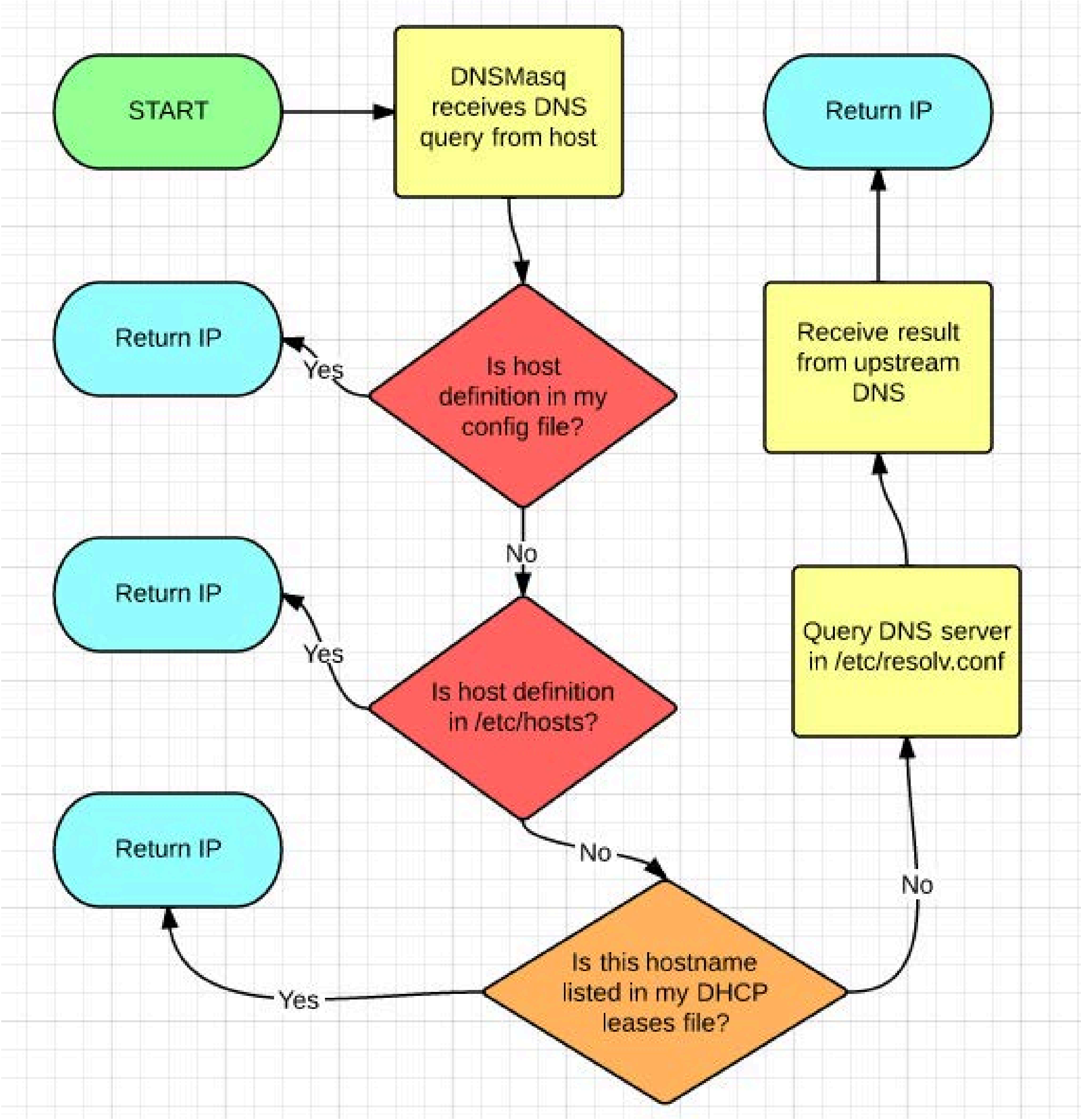
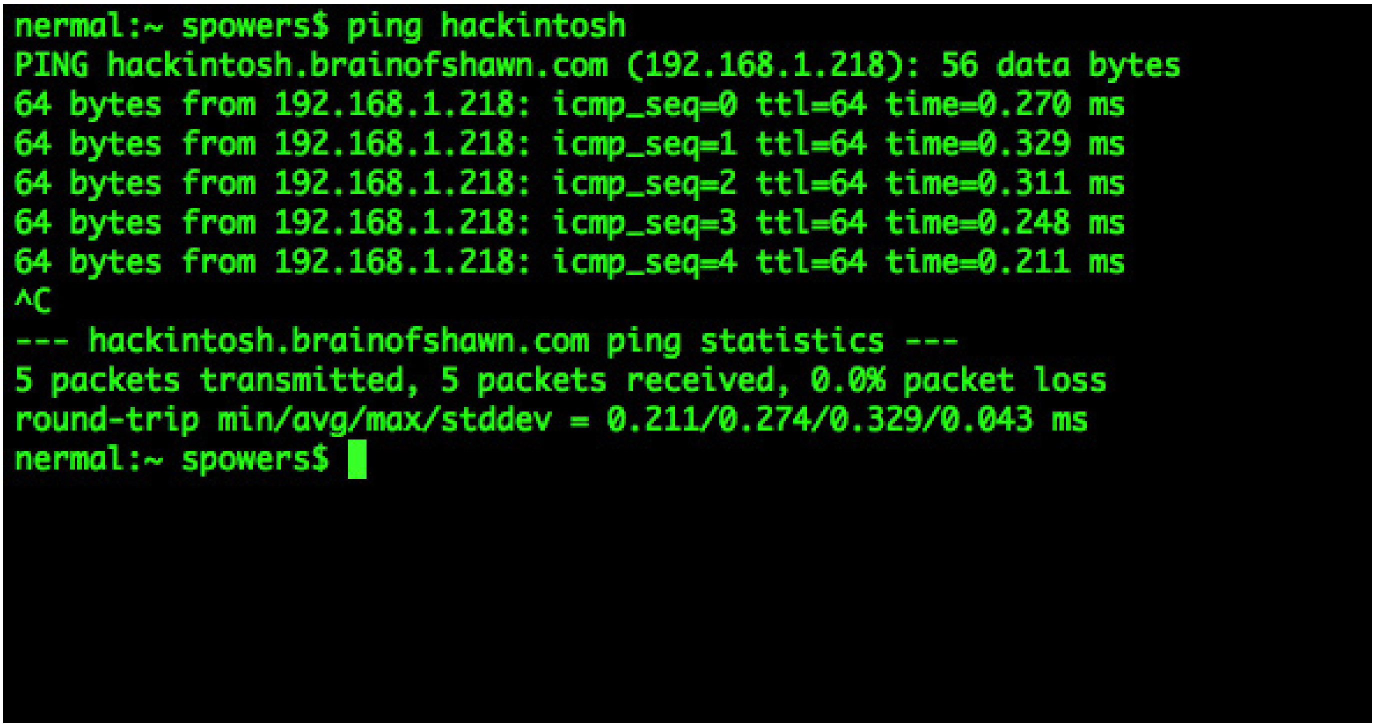

START
DNSMasq receives DNS query from host
Return IP
Is host definition in my config file?
Yes
Return IP
No
Receive result from upstream DNS
Is host definition in /etc/hosts?
Yes
Return IP
Query DNS server in /etc/resolv.conf
No
No
Return IP
Is this hostname listed in my DHCP leases file?
Yes

Figure 3. If you use DHCP, it automatically integrates into your DNS system—awesome for finding dynamically assigned IPs!



```
nermal:~ spowers$ ping hackintosh
PING hackintosh.brainofshawn.com (192.168.1.218): 56 data bytes
64 bytes from 192.168.1.218: icmp_seq=0 ttl=64 time=0.270 ms
64 bytes from 192.168.1.218: icmp_seq=1 ttl=64 time=0.329 ms
64 bytes from 192.168.1.218: icmp_seq=2 ttl=64 time=0.311 ms
64 bytes from 192.168.1.218: icmp_seq=3 ttl=64 time=0.248 ms
64 bytes from 192.168.1.218: icmp_seq=4 ttl=64 time=0.211 ms
^C
--- hackintosh.brainofshawn.com ping statistics ---
5 packets transmitted, 5 packets received, 0.0% packet loss
round-trip min/avg/max/stddev = 0.211/0.274/0.329/0.043 ms
nermal:~ spowers$ █
```

Figure 4. There are no DNS entries anywhere for my Hackintosh, but thanks to DNSMasq, it's pingable via its hostname.

Basically, if your friend brings a laptop to your house and connects to your network, when it requests a DHCP address, it tells the DNSMasq server its hostname. From that point on, until the lease expires, any DNS queries the server receives for that hostname will be returned as the IP it assigned via DHCP. This is very convenient if you have a computer connected to your network whose hostname you know, but it gets a dynamically assigned IP address. In my house, I have a Hackintosh computer that just gets a random IP assigned via DNSMasq's DHCP server. Figure 4 shows what happens when I ping the name "hackintosh" on my network. Even though it isn't listed in any of the server's configuration files, since it handles DHCP, it creates a DNS entry on the fly.

Static DHCP entries can be entered in the single configuration file using this format:

```
dhcp-host=90:fb:a6:86:0d:60,xbmc-livingroom,192.168.1.20
dhcp-host=b8:27:eb:e3:4c:5f,xbmc-familyroom,192.168.1.21
dhcp-host=b8:27:eb:16:d9:08,xbmc-masterbedroom,192.168.1.22
dhcp-host=00:1b:a9:fa:98:a9,officelaser,192.168.1.100
dhcp-host=04:46:65:d4:e8:c9,birdcam,192.168.1.201
```

It's also valid to leave the hostname out of the static declaration, but adding it to the DHCP reservation adds it to the DNS server's list of known addresses, even if the client itself doesn't tell the DHCP server its hostname. You also could just add the hostname to your DNSMasq server's /etc/hosts file, but I prefer to make my static DHCP entries with hostnames, so I can tell at a glance what computer the reservation is for.

And If That's Not Enough...

The above scenarios are all I use DNSMasq for on my local network. It's more incredible than any DHCP/DNS combination I've ever used before, including the Windows and OS X server-based services I've used in large networks. It does provide even more services, however, for those folks needing them.

The TFTP server can be activated via configuration file to serve boot files, configuration files or any other TFTP files you might need served on your network. The service integrates flawlessly with the DHCP server to provide boot filenames, PXE/BOOTP information, and custom DHCP options needed for booting even the most finicky devices. Even if you need TFTP services for a non-boot-related reason, DNSMasq's server is just a standard TFTP service that will work for any computer or device requiring it.

If you've read Kyle Rankin's recent articles on DNSSEC and want to make sure your DNS information is secure, there's no need to install BIND. DNSMasq supports DNSSEC, and once again provides configuration examples in the configuration file.

Truly, DNSMasq is the unsung hero for consumer-grade Internet routers. It allows those tiny devices to provide DNS and DHCP for your entire network. If you install the program on a regular server (or teeny tiny Raspberry Pi or Cubox), however, it can become an extremely robust platform for all your network needs. If it weren't for my need to get a more powerful and reliable router, I never would have learned about just how amazing DNSMasq is. If you've ever been frustrated by BIND, or if you'd just like to have more control over the DNS and DHCP services on your network, I urge you to give DNSMasq a closer look. It's for more than just your DD-WRT router!■

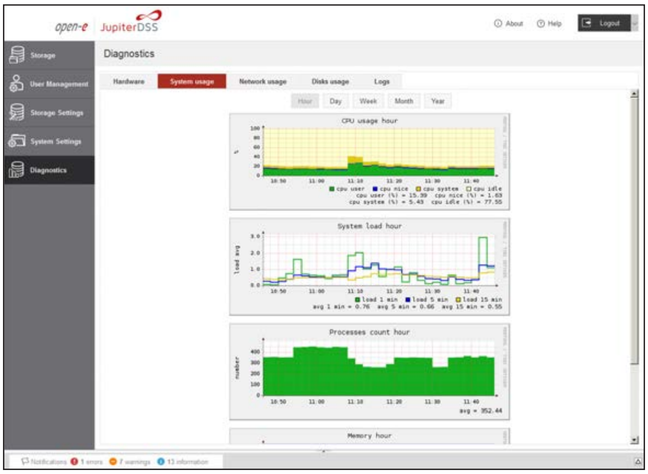
Shawn Powers is the Associate Editor for *Linux Journal*. He's also the Gadget Guy for LinuxJournal.com, and he has an interesting collection of vintage Garfield coffee mugs. Don't let his silly hairdo fool you, he's a pretty ordinary guy and can be reached via e-mail at shawn@linuxjournal.com. Or, swing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

Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.

CoreOS Managed Linux

The team at CoreOS recently announced three big, concurrent news items. First, CoreOS rolled out its two flagship products: CoreOS Managed Linux, which bills as "the world's first OS as a Service", and CoreUpdate, which provides a dashboard for full control of rolling updates. Second, CoreOS announced the receipt of $8 million in funding from the venture capital firm Kleiner Perkins Caulfield and Byer, meaning you are sure to hear more about CoreOS in the future. Capital is following this development because companies like Google, Facebook, Twitter and others must run their services at scale with high resilience. The solution is CoreOS, a new Linux OS that has been re-architected to provide the foundation of warehouse-scale computing. CoreOS customers receive a continuous stream of updates and patches (via CoreUpdate), as well as a high level of commercial support, eliminating the need for major OS migrations every few years. The goal is to make the migration to CoreOS's products the last migration they ever need. Included platforms are Bare Metal, Amazon, Google and Rackspace, among others.

http://www.coreos.com

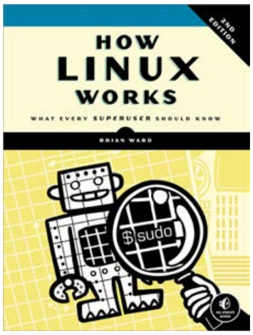


Open-E JupiterDSS

The latest release of the Open-E JupiterDSS—or Defined Data Storage Software—is a result of three years of development, testing, working closely with partners and integrating customer feedback, reported maker Open-E. The firm added that Open-E JupiterDSS provides enterprise users the highest level of performance with unlimited capacity and volume size. Delivered through Open-E certified partners as a software-defined storage system, Open-E JupiterDSS comes complete with advanced features, including thin provisioning, compression and de-duplication. This milestone release of the company's flagship application comes in response to customers demanding ever larger storage environments while maintaining high benchmarks for quality, reliability, performance and price. Open-E JupiterDSS features a ZFS- and Linux-based storage operating system.

http://www.open-e.com

Brian Ward's *How Linux Works* (No Starch Press)



Though there now exists a seemingly limitless list of great Linux books, those like Brian Ward's *How Linux Works: What Every Superuser Should Know* are the kind of books that should go into the "Linux Beginner's Canon". Ward's book contains the essentials that new enthusiasts should know as they embark on their journey of Linux discovery. To master Linux and avoid obstacles, one needs to understand Linux internals like how the system boots, how networking works and what the kernel actually does. In this completely revised second edition, author Ward makes the concepts behind Linux internals accessible to anyone who wants to understand them. Inside these information-packed pages, readers will find the kind of knowledge that normally comes from years of experience doing things the hard way, including essential topics like Linux booting; how the kernel manages devices, device drivers and processes; how networking, interfaces, firewalls and servers work; and how development tools, shared libraries and shell scripts work. Publisher No Starch Press notes that the book's combination of background, theory, real-world examples and patient explanations will teach readers to understand and customize their systems, solve pesky problems and take control of their OS.

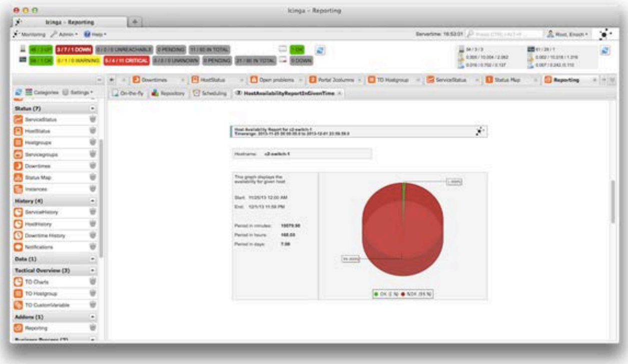
http://www.nostarch.com

QLogic's NetXtreme II 10Gb Ethernet Adapters

As part of its mission to be the most comprehensive supplier of network infrastructure across IBM's entire server portfolio, QLogic recently released the NetXtreme II 10Gb Ethernet adapter line, which the firm claims is the first for IBM Power Systems. Available in 10GBASE-T or SFP+ versions, QLogic 10GbE adapters are critical for achieving desired levels of performance and consolidation for cloud computing, convergence and data-intensive application environments, such as video and social-media content. Migration to the new adapters is seamless, says QLogic, because of the full backward-compatibility with an installed base of millions of 1GbE twisted-pair switch ports. For applications requiring leading-edge networking performance, QLogic NetXtreme II SFP+ low-profile adapters combine advanced, multiprotocol offload technologies with standard Ethernet functionality.

http://www.qlogic.com

The Icinga Project's Icinga 2



Although the Icinga 2 monitoring solution is backward-compatible to Nagios and Icinga 1, the Icinga Project considers its latest masterpiece "a league apart from its predecessors". Icinga 2 is a next-generation open-source monitoring solution designed to meet modern IT infrastructure needs. Built from scratch and based on C++, Icinga 2 boasts multi-threaded architecture for high-performance monitoring, a new dynamic configuration format and distributed monitoring out of the box. Version 2 also features multiple back ends for easy integration of popular add-ons. In contrast to its predecessors, core features and their related libraries are shipped with the application and can be activated as needed via "icinga2-enable-feature" soft link commands, easing installation and extension, enabling multi-functionality and improving performance. Additional new advancements include cluster stacks, a new object-based template-driven configuration format and extensive documentation, as well as configuration validation and syntax highlighting to support troubleshooting.

http://www.icinga.org

Zentyal Server

zentyal
Active Exchange

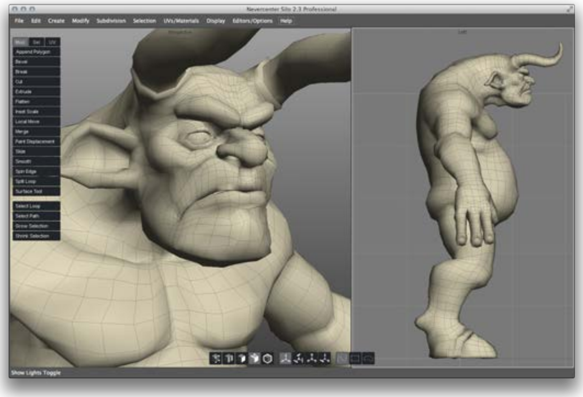
What's special about the upgraded Zentyal Server 3.5 is that it integrates both the complex Samba and OpenChange technologies, making it easy to integrate Zentyal into an existing Windows environment and carry out phased, transparent migration to Linux. In other words, the Zentyal Linux Small Business Server offers a native drop-in replacement for Windows Small Business Server and Microsoft Exchange Server that can be set up in less than 30 minutes and is both easy to use and affordable. Because Zentyal Server's 3.5 release focuses on providing a stable server edition with simplified architecture, it comes with a single LDAP implementation based on Samba4, helping solve a number of synchronization issues caused by having two LDAP implementations in the earlier editions. In addition, a number of modules have been removed in order to advance the core goal of offering the aforementioned drop-in Windows server replacement capabilities.

http://www.zentyal.com

Dave Seff's *Mastering 3D Printing* LiveLessons—Video Training (Addison-Wesley Professional)

The conditions are riper than ever to learn about 3-D printing, especially thanks to a new video training program called *Mastering 3D Printing* LiveLessons from Addison-Wesley Professional. What's special for our readers is that the instructor, Dave Seff, is a modern-day Goethe-esque geek, a Linux/UNIX engineer who also has mastered topics as varied as computer and mechanical design, machining, electronics and computer programming. Seff has even built several of his own 3-D printers and other CNC machines. In the videos, Seff not only teaches the fundamentals of 3-D printing but also does so utilizing the open-source modeling software, Blender. Seff explores how to create a 3-D model (beginner and advanced lessons), how to slice (prepare for printing) and then how to print a 3-D model. Seff also covers troubleshooting problems when they arise.
http://www.informit.com/livelessons

Nevercenter's Silo 3D Modeler



In response to its number-one request, Nevercenter has ported version 2.3 of its Silo 3D Modeler application to Linux, complementing the existing Mac OS and Windows editions. Silo, popular with designers of video games, movies and 3-D architectural applications, can be used either as a standalone tool or as a versatile element of a multi-application 3-D graphics workflow. Nevercenter is finding ever more studios and individuals in this space moving to Linux. Silo's internals also have received significant updates, including an updated windowing system and bug fixes across all platforms, as well as added support for .stl import.
http://www.nevercenter.com/silo

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.

Provisioning X.509 Certificates Using RFC 7030

Learn how to use libest to deploy X.509 certificates across your enterprise.

JOHN FOLEY

Have you ever found yourself in the need of an X.509 certificate when configuring a Linux service? Perhaps you're enabling HTTPS using Apache or NGINX. Maybe you're configuring IPsec between two Linux hosts. It's not uncommon to use on-line forums or developer blogs to find setup instructions to meet these goals quickly. Often these sources of information direct the reader to use a self-signed certificate. Such shortcuts overlook good security practices that should be followed as part of a sound Public Key Infrastructure (PKI) deployment strategy. This article proposes a solution to the problem of widespread deployment of X.509 certificates using Enrollment over Secure Transport (EST). First, I provide a brief primer on PKI. Then, I give an overview of using EST to provision a certificate, demonstrated using both curl and libest. Finally, I show a brief example of an OpenVPN deployment using certificates provisioned with EST.

PKI Primer

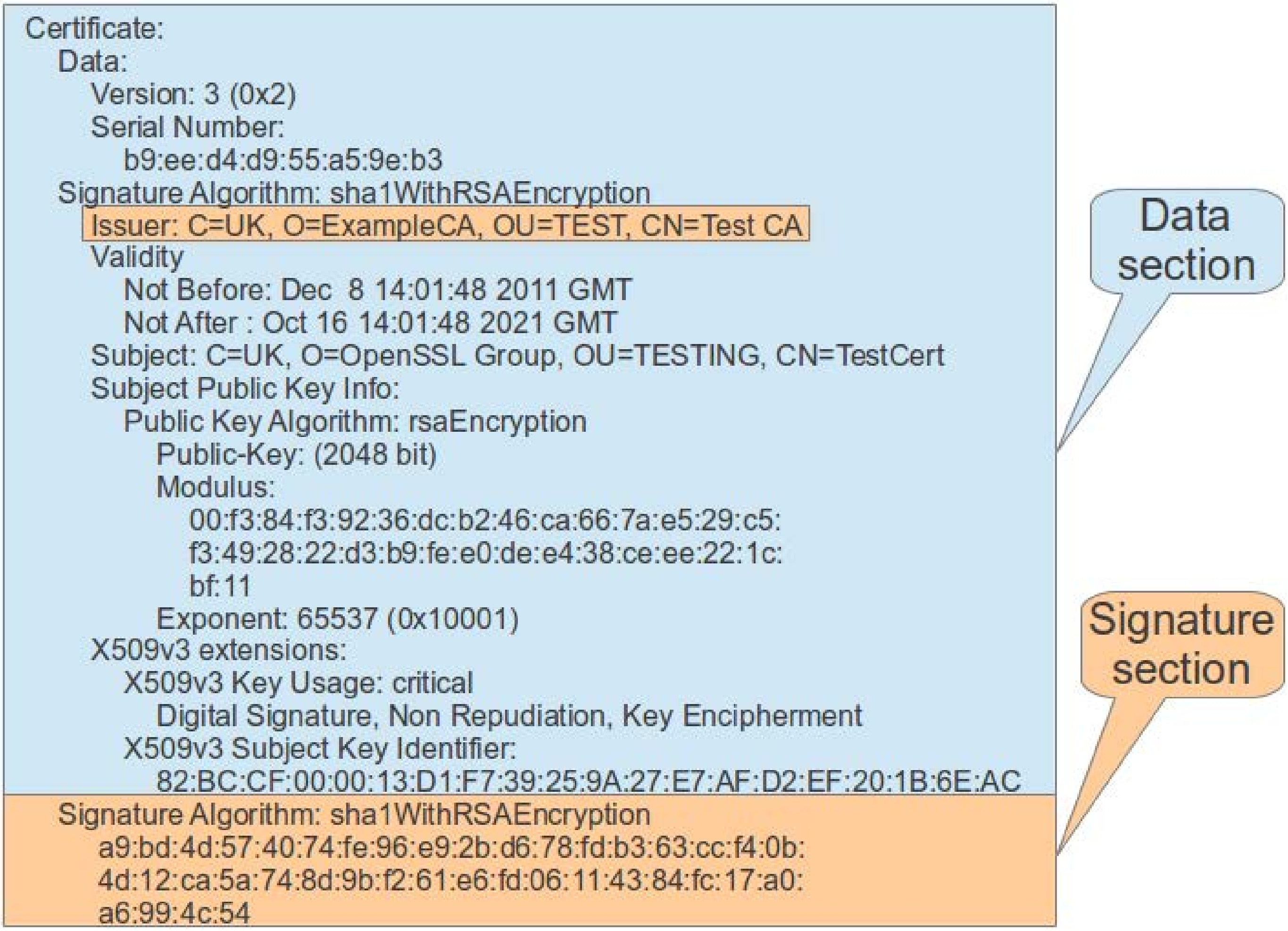
Public Key Infrastructure consists of several building blocks, including X.509 certificates, Certificate Authorities, Registration Authorities, public/private key pairs and certificate revocation lists. X.509 certificates are provisioned by end-entities from either an RA or a CA. An end-entity will use the X.509 certificate to identity itself to a peer when establishing a secure communication channel. The X.509 certificate contains a public key that another entity can use to verify the identity of the entity presenting the X.509 certificate. The owner of an X.509 certificate retains the private key associated with the public key in its X.509 certificate. The private key is used only by the end-entity that owns the X.509 and must remain confidential. Leakage of the private key compromises the security model.

X.509 certificates are the most commonly used approach for verifying peer identity on the Internet today. X.509 certificates are used with TLS, IPsec, SSH and other protocols. For example, a Web browser will use the X.509 certificate from a Web server to verify the identity of the server. This is achieved by using the public key in the CA certificate and the signature in the Web server certificate.

PKI allows for multiple layers of trust, with the root CA at the top of the trust chain. The root CA also is called a trust anchor. The root CA can delegate authority to a sub-CA or an RA. The sub-CA or RA can provision X.509 certificates on behalf of an end-entity, such as a

Web browser or a Web server. For simplicity, this article is limited to showing a single layer of trust where the root CA generates certificates directly for the end-entities.

The multiple layers of trust in the PKI hierarchy are implemented using asymmetric cryptographic algorithms. RSA is the most common asymmetric algorithm used today. RSA is named after its inventors: Ron Rivest, Adi Shamir and Leonard Adleman. RSA uses a public/private key pair. The X.509 certificate generated by a CA contains a digital signature. This signature is the encrypted output from the RSA algorithm. The CA will calculate the hash (for example, SHA1) of the certificate contents. The hash value is then encrypted using the CA's private RSA key. The CA also will include information about itself in the new certificate in the Issuer field, which allows another entity to know which CA generated the certificate (Figure 1). This becomes



```
Certificate:
   Data:
      Version: 3 (0x2)
      Serial Number:
         b9:ee:d4:d9:55:a5:9e:b3
   Signature Algorithm: sha1WithRSAEncryption
      Issuer: C=UK, O=ExampleCA, OU=TEST, CN=Test CA
      Validity
         Not Before: Dec  8 14:01:48 2011 GMT
         Not After : Oct 16 14:01:48 2021 GMT
      Subject: C=UK, O=OpenSSL Group, OU=TESTING, CN=TestCert
      Subject Public Key Info:
         Public Key Algorithm: rsaEncryption
            Public-Key: (2048 bit)
            Modulus:
               00:f3:84:f3:92:36:dc:b2:46:ca:66:7a:e5:29:c5:
               f3:49:28:22:d3:b9:fe:e0:de:e4:38:ce:ee:22:1c:
               bf:11
            Exponent: 65537 (0x10001)
      X509v3 extensions:
         X509v3 Key Usage: critical
            Digital Signature, Non Repudiation, Key Encipherment
         X509v3 Subject Key Identifier:
            82:BC:CF:00:00:13:D1:F7:39:25:9A:27:E7:AF:D2:EF:20:1B:6E:AC
   Signature Algorithm: sha1WithRSAEncryption
       a9:bd:4d:57:40:74:fe:96:e9:2b:d6:78:fd:b3:63:cc:f4:0b:
       4d:12:ca:5a:74:8d:9b:f2:61:e6:fd:06:11:43:84:fc:17:a0:
       a6:99:4c:54
```

Figure 1. Anatomy of an X.509 Certificate

important when another entity needs to verify the authenticity of another entity's certificate.

When another entity needs to verify the identity and authenticity of an X.509 certificate, the public key from the CA certificate is used to verify the X.509 certificate to be verified. The verifying identity calculates the hash (for example, SHA1) of the certificate, similar to how the CA did this when generating the certificate. Instead of using the CA private key, the verifying entity will use the CA public key to encrypt the hash result. If the public key encrypted value of the hash matches the signature that is in the X.509 certificate, the verifying identity is assured that the certificate is valid and was issued by the CA.

Astute readers will observe that the verifying entity needs to have the CA public key to perform this verification process. This problem is commonly solved by deploying the public keys of well-known certificate authorities with the software that will be performing the verification process. This is known as out-of-band trust anchor deployment. Commonly used Web browsers (such as Firefox, Chrome and IE) follow this model. When you install the Web browser on your computer, the well-known CA certificates are installed with the software. When you browse to a secure Web site, the Web browser uses this trust anchor of pre-installed CA certificates

Elliptic Curve Digital Signature Algorithm

Elliptic Curve Digital Signature Algorithm (ECDSA) is an asymmetric cryptographic algorithm that can be used similar to RSA. ECDSA uses a public/private key pair similar to RSA. However, ECDSA provides an equivalent security level to RSA using much smaller key sizes. As of 2014, the NIST minimum recommended RSA key size is 2048 bits. Using an ECDSA 256-bit key provides better security than RSA 2048, which significantly reduces the amount of data that needs to be transmitted between peers during the key exchange process of protocols like TLS, DTLS and IKE. This bandwidth savings is desirable for IoT devices or other end-entities that need to minimize bandwidth consumption over radio interfaces.

(containing public keys) to verify the X.509 certificate presented by the Web server.

FQDN

X.509 certificates should contain the Fully Qualified DNS Domain Name (FQDN) of the entity that owns the certificate. RFC 6125 provides guidance on how FQDN should be implemented in a PKI deployment. The FQDN should be in either the Subject Common Name or the Subject Alternative Name of the X.509 certificate. When an entity, such as a Web browser, verifies the authenticity of the peer certificate, the FQDN should be checked to ensure that it matches the hostname used to initiate the IP connection. For example, when you browse to https://www.foobar.org, the X.509 certificate presented by the Web server should contain www.foobar.org in either the Subject Common Name or Subject Alternative Name. Checking the FQDN helps mitigate an MITM (Man In The Middle) attack.

Enrollment over Secure Transport

Enrollment over Secure Transport (EST) is defined in RFC 7030. This protocol solves the challenge of PKI deployment across a large infrastructure. For example, RFC 7030 defines methods for both provisioning end-entity certificates and deploying CA public keys, which are required for end-entities to verify each other. This article focuses on the client side of solving these two challenges. Let's use the interop test server hosted at **http://testrfc7030.cisco.com** as the EST server for the examples. *Note: the certificates generated by this test server are for demonstration purposes only and should not be used for production deployments.*

RFC 7030 defines a REST interface for various PKI operations. The first operation is the /cacerts method, which is used by an end-entity to retrieve the current CA certificates. This set of CA certificates is called the *explicit* trust anchor. The /cacerts method is the first step invoked by the end-entity to ensure that the latest trust anchor is used for subsequent EST operations. The following steps show how to use curl as an EST client to issue the /cacerts operation.

Step 1: Retrieve the public CA certificate used by testrfc7030.cisco.com. Note: this step emulates the out-of-band deployment of the *implicit* trust anchor:

```
wget http://testrfc7030.cisco.com/DST_Root_CA_X3.pem
```

Step 2: Use curl to retrieve the latest *explicit* trust anchor from the test server:

```
curl https://testrfc7030.cisco.com:8443/.well-known/est/cacerts \
    -o cacerts.p7 --cacert ./DST_Root_CA_X3.pem
```

Note: you can use a Web browser instead of wget/curl for steps 1 and 2. Enter the URL shown in step 2 into your browser. Then save the result from the Web server to your local filesystem using the filename cacerts.p7.

Step 3: Use the OpenSSL command line to convert the trust anchor to PEM format. This is necessary, because the EST specification requires the /cacerts response to be base64-encoded PKCS7. However, the PEM format is more commonly used:

```
openssl base64 -d -in cacerts.p7 | \
   openssl pkcs7 -inform DER -outform PEM \
   -print_certs -out cacerts.pem
```

The certificate in cacerts.pem is the explicit trust anchor. You will use this certificate later to establish a secure communication channel between two entities. However, first you need to provision a certificate for each entity. You'll use OpenSSL to create a certificate request. The certificate request is called a CSR, defined by the PKCS #10 specification. You'll also create your public/private RSA key pair when creating the CSR. You'll use OpenSSL to create both the CSR and the key pair.

Step 4: Generate an RSA public/private key pair for the end-entity and create the CSR. The CSR will become the X.509 certificate after it has been signed by the CA. You will be prompted to supply the values for the Subject Name field to be placed in the X.509 certificate. These values include the country name, state/province, locality, organization name, common name and your e-mail address. The common name should contain the FQDN of the entity that will use the certificate. The challenge password and company name are not required and can be left blank:

```
openssl req -new -sha256 -newkey rsa:2048 \
    -keyout privatekey.pem -keyform PEM -out csr.p10

Generating a 2048 bit RSA private key
........+++
....................................................
....................................................
.................+++
writing new private key to 'privatekey.pem'
```

```
Enter PEM pass phrase:
Verifying - Enter PEM pass phrase:
-----
You are about to be asked to enter information that
will be incorporated
into your certificate request.
What you are about to enter is what is called a
Distinguished Name or a DN.
There are quite a few fields but you can leave some
blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
-----
Country Name (2 letter code) [AU]:US
State or Province Name (full name) [Some-State]:Colorado
Locality Name (eg, city) []:Aspen
Organization Name (eg, company) [Internet Widgits Pty Ltd]:ACME, Inc.
Organizational Unit Name (eg, section) []:Mfg
Common Name (e.g. server FQDN or YOUR name) []:mfgserver1.acme.org
Email Address []:jdoe@acme.org

Please enter the following 'extra' attributes
to be sent with your certificate request
A challenge password []:
An optional company name []:
```

Now that you have the CSR and key pair, you need to send the CSR to the CA to have it signed and returned to you as an X.509 certificate. The EST /simpleenroll REST method is used for this purpose. Curl can be used again to send the CSR to the CA as a RESTful EST operation.

Step 5: Use curl to enroll a new certificate from the test server using the CSR you just generated. Normally the explicit trust anchor is used for this step. However, the test server doesn't use the explicit trust anchor for HTTPS services. Therefore, you'll continue to use the implicit trust anchor (DST_Root_CA_X3.pem) for this step (note: the test CA at testrfc7030.cisco.com uses a well-known user name/password of estuser/estpwd):

```
curl https://testrfc7030.cisco.com:8443/.well-known/est/simpleenroll \
    --anyauth -u estuser:estpwd -o cert.p7 \
    --cacert ./DST_Root_CA_X3.pem --data-binary @csr.p10 \
    -H "Content-Type: application/pkcs10"
```

Step 6: If successful, the curl command should place the new certificate in the cert.p7 file. The EST specification requires the certificate to be base64-encoded PKCS7. Because PEM is a more commonly used format, you'll use OpenSSL to convert the new certificate to PEM format:

```
openssl base64 -d -in cert.p7 | openssl pkcs7 -inform DER \
     -outform PEM -print_certs -out cert.pem
```

Step 7: Finally, use OpenSSL again to confirm the content in the certificate. The Subject Name should contain the values you used to create the CSR earlier:

```
openssl x509 -text -in cert.pem

Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 42 (0x2a)
  Signature Algorithm: ecdsa-with-SHA1
    Issuer: CN=estExampleCA
    Validity
        Not Before: Jun  4 18:42:56 2014 GMT
        Not After : Jun  4 18:42:56 2015 GMT
    Subject: CN=mfgserver1.acme.org
    Subject Public Key Info:
        Public Key Algorithm: rsaEncryption
            Public-Key: (2048 bit)
            Modulus:
                00:c0:4c:65:d1:6c:d2:8b:7d:37:b9:a1:67:da:7a:
                a1:6c:4f:b9:9f:68:e0:9a:44:24:a0:aa:54:55:19:
                c0:fc:6b:35:c5:a7:14:ed:70:e9:99:32:6a:21:19:
                49:2b:8e:42:89:eb:9f:ec:3d:69:75:49:2f:f7:18:
                f6:14:ed:d5:71:54:b5:0a:d0:f3:7b:8e:36:19:f1:
                45:07:37:b9:aa:73:7c:60:bb:e1:f1:ac:b2:75:74:
                22:9e:5d:b5:ee:13:7c:b8:31:61:c5:9a:ef:7e:07:
                24:8d:c8:50:44:89:6d:fe:dd:e0:28:fd:80:1c:b9:
                61:94:8d:63:cd:54:2c:a9:86:7a:3b:35:62:e9:c6:
                76:58:fb:27:c1:bf:db:c2:03:66:e5:dd:cb:75:bc:
                72:6c:ca:27:76:2a:f7:48:d5:3b:42:de:85:8e:3b:
                15:f1:7a:e4:37:3c:96:b2:91:70:6f:97:22:15:c6:
                82:ea:74:8b:f2:80:39:c1:c2:10:78:6e:70:11:78:
                31:2f:4a:c3:c4:2b:ab:2f:4d:f2:87:15:59:88:b3:
                17:12:1d:92:b2:6d:a6:8a:94:3f:b3:76:18:53:f9:
                59:29:e1:9b:8c:81:41:7e:8c:a2:a7:34:c9:b4:07:
                32:77:57:37:59:dd:fb:36:02:59:74:bb:96:6e:e7:
                3f:b7
            Exponent: 65537 (0x10001)
    X509v3 extensions:
        X509v3 Basic Constraints:
            CA:FALSE
        X509v3 Key Usage:
            Digital Signature
        X509v3 Subject Key Identifier:
     E2:C5:FC:55:42:D9:52:D9:81:F7:CC:6C:01:56:BF:10:35:41:7A:D8
        X509v3 Authority Key Identifier:
     keyid:EE:DE:AA:C0:5B:AC:38:7D:F3:08:26:33:73:00:3F:F3:2B:63:41:F8

  Signature Algorithm: ecdsa-with-SHA1
     30:45:02:20:1e:b6:b6:32:fa:79:de:26:c0:34:0d:a5:5c:70:
     cb:27:a3:8f:fc:9f:d2:1f:ca:5c:99:fd:d0:ff:bf:7f:51:e8:
     02:21:00:be:1f:36:b3:f6:46:65:58:eb:57:05:c3:af:4c:4a:
     0e:d1:28:e9:0b:58:e3:ac:3f:db:27:36:33:98:3f:b1:9e
```

At this point, you have the new certificate and associated private key. The certificate is in the file cert.pem. The private key is in the file privatekey.pem. These can be used for a variety of security protocols including TLS, IPsec, SSH and so on.

libest

Curl provides a primitive method to issue the RESTful operations of the EST enrollment process. However, the curl command-line options required to enroll a new certificate securely are cumbersome and error-prone. Additionally, curl is unable to perform the TLS channel binding requirements defined in RFC 7030 section 3.5. There is an open-source alternative called libest. This library supports client-side EST operations required to provision a certificate. The libest library comes with a client-side command-line tool to replace the curl commands described earlier. Additionally, libest exposes an API when EST operations need to be embedded into another application. Next, I demonstrate how use the libest CLI to enroll a certificate from the test server.

libest is available at **https://github.com/cisco/libest**. It's known to work on popular Linux distributions, such as Ubuntu and Centos. You will need to download, configure, compile and install libest to follow along. The default installation location is /usr/local/est. libest requires that you install the OpenSSL devel package prior to configuration. OpenSSL 1.0.1 or newer is required.

You'll use the same implicit trust anchor and CSR that you used earlier when using curl as the EST client. The implicit trust anchor is located in DST_Root_CA_X3.pem, and the CSR is in csr.p10.

Step 1: Configure the trust anchor to use with libest:

```
export EST_OPENSSL_CACERT=DST_Root_CA_X3.pem
```

Step 2: Using the same CSR used earlier with curl in the csr.p10 file, provision a new X.509 certificate for the CSR:

```
estclient -e -s testrfc7030.cisco.com -p 8443 \
    -o . -y csr.p10 -u estuser -h estpwd
```

Heartbleed

The Heartbleed bug in OpenSSL was a severe vulnerability in OpenSSL that was publicly announced in April 2014. The severity of this bug was due to the potential for leaking the X.509 private key of the TLS server. When a private key is leaked, previously recorded communications using the key can be compromised. For example, a compromised TLS server may have had all communications revealed since the private key was issued to it. It's not uncommon for certificates to be issued for a year or longer. Imagine every transaction you conducted with your on-line bank during the past year being compromised. The Heartbleed bug provides motivation to use shorter validity periods when issuing X.509 certificates. EST can be used to automate the certificate renewal process to avoid interruptions in service due to certificate expiry.

Step 3: Similar to the curl example shown earlier, use OpenSSL to convert the new certificate to PEM format and confirm the contents of the certificate:

```
openssl base64 -d -in ./cert-0-0.pkcs7 | \
    openssl pkcs7 -inform DER  -print_certs -out cert.pem
openssl x509 -text -in cert.pem
```

This enrollment procedure can be used on any number of end-entities. These end-entities than can use their certificate along with the explicit trust anchor to verify each other when establishing secure communications. This eliminates the need to generate self-signed certificates and manually copy those certificates among end-entities. The enrollment process could be automated using curl or libest. EST can be used for certificate renewal as well, which can automate the process of renewing certificates that are about to expire. Automating the process can facilitate the shortening of certificate validity periods, which improves the overall security posture of a PKI deployment.

Using the New Certificates

Now that you have provisioned a new certificate for the end-entity, the certificate can be used with a variety of protocols. Next, I show how the certificate can be used with OpenVPN to establish a secure communication channel between two Linux hosts. The certificate enrollment process described earlier needs to be completed on each Linux host.

OpenVPN supports a wide variety of configurations. In this example, let's use the TLS client/server model. Two Linux hosts are required. OpenVPN should be installed on both systems. One host will operate as the TLS server for VPN services. The other host will operate as the TLS client. In this example, the IP address for the physical Ethernet interface on the server is 192.168.1.35. The TAP

Listing 1. TLS Server OpenVPN Config

```
dev tap
ifconfig 10.3.0.1 255.255.255.0
tls-server
dh dh2048.pem
ca cacerts.pem
cert cert.pem
key privatekey.pem
```

Listing 2. TLS Client OpenVPN Config

```
remote 192.168.1.35
dev tap
ifconfig 10.3.0.2 255.255.255.0
tls-client
ca cacerts.pem
cert cert.pem
key privatekey.pem
```

interface is used on both the client and the server for the VPN tunnel. The server TAP interface uses the address 10.3.0.1, while the client uses 10.3.0.2. The certificates provisioned using EST are configured in the OpenVPN configuration file on each system (see Listings 1 and 2).

Step 1: Generate DH parameters for the OpenVPN server:

```
openssl gendh -out dh2048.pem 1024
```

Step 2: Start the OpenVPN server:

```
sudo openvpn vpnserver.conf
```

Step 1: Start the OpenVPN client:

```
sudo openvpn vpnclient.conf
```

Step 2: Ping across the tunnel to ensure that the VPN is working:

```
ping 10.3.0.1
```

Summary

This article has focused on the minimal client-side EST operations required to establish an explicit trust anchor and provision a new certificate. EST provides additional capabilities including certificate renewal, CSR attributes and server-side key generation. EST also provides for various client authentication methods other than using a user name/password. The client can be authenticated using SRP or an existing X.509 certificate. For example, an existing certificate on the EST client should be used when renewing a certificate prior to expiration. A good source of information for learning more about these concepts is the EST specification (RFC 7030).

This article has focused on the client side. In the future, I will look at implementing EST on the server side to front a CA. The EST protocol is a new protocol and not widely adopted at this time. Several well-known commercial CA vendors are currently implementing EST. If you use a commercial CA as part of your PKI infrastructure today, you may want to ask your CA vendor about its plans to support EST in the future.■

John Foley is a Technical Leader in product development at Cisco Systems. John has worked on a variety of projects during the past 14 years at Cisco, including VoIP systems, embedded development, security and multicast. John has spent the past three years working with security protocols, including TLS, SRTP and EST. Prior to this, John worked for seven years in the IT industry doing application development and RDBMS design. John is an active contributor to the libest and libsrtp projects on GitHub.

Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.

drupalize.me

Instant Access to Premium Online Drupal Training

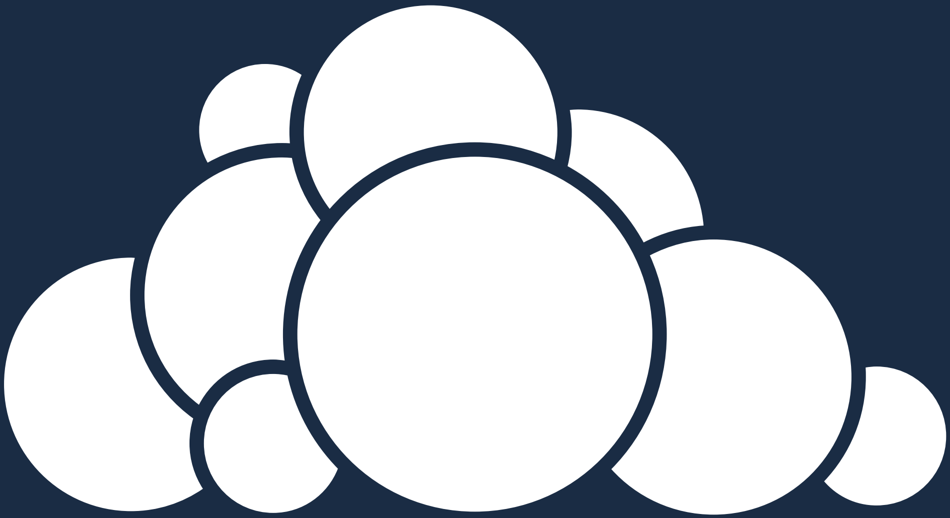
- ✓ *Instant access to hundreds of hours of Drupal training with new videos added every week!*
- ✓ *Learn from industry experts with real world experience building high profile sites*
- ✓ *Learn on the go wherever you are with apps for iOS, Android & Roku*
- ✓ *We also offer group accounts. Give your whole team access at a discounted rate!*

Learn about our latest video releases and offers first by following us on Facebook and Twitter (@drupalizeme)!

Go to http://drupalize.me and get Drupalized today!

```
 *
 * Usage: $(obj).insertAtCursor(text);
 *   obj = a textarea or textfield
 *   text = a string to insert
 */
$.fn.extend({
  insertAtCaret: function(myValue){
    return $(this).each(function(){
      // If target element is hidden, don't do it.
      if ($(this).is(':hidden') || $(this).parents(':hidde
        return;
      }
      if (document.selection) {
        this.fo
        sel = c
        sel.tex
        this.fc
      }
      else if (
        var sta
        var enc
        var scr
        this.va
        this.fc
        // High
```

Synchronize Your Life with ownCloud



How to build a synchronization server and keep your data where you want it.

MIKE DIEHL

Like most families these days, our family is extremely busy. We have four boys who have activities and appointments. My wife and I both have our own businesses as well as outside activities. For years, we've been using eGroupware to help coordinate our schedules and manage contacts. The eGroupware system has served us well for a long time. However, it is starting to show its age. As a Web-based groupware system, it's pretty well polished, but it doesn't hold a candle to Kontact or Thunderbird. Also, my wife finds that she needs to access her calendar from her Android phone, and eGroupware just isn't very mobile-friendly. Sure, we can set up calendar synchronization, but eGroupware seems to have added synchronization as an afterthought, and it really doesn't work as well as we'd like.

So, I started looking for a new groupware system that would allow us to access our calendars and contacts seamlessly from our smartphones, a Web browser or our favorite desktop PIM. Sure, we simply could have uploaded all of our information to a Google server. However, I may be paranoid, but I just don't want an outside corporation having personal information like who my friends are, my wife's recipe for cornbread or what I'm doing next Tuesday at 3:00pm; it's just none of their business. By hosting my own groupware server, I maintain my privacy and don't have to worry about arbitrary changes in service.

The ownCloud system has a calendar, address book, task manager, bookmark manager and file manager, among other features. These services can be accessed from any Web browser. However, ownCloud also supports the calDAV, cardDAV and webDAV standards, so synchronization with other clients should be pretty straightforward.

In practice, there was a slight learning curve, but synchronization works very well. The ownCloud system also allows you to integrate third-party modules (apps) in order to add features. Apps are available that provide music and video streaming, file encryption, e-mail and feature enhancements for existing functions.

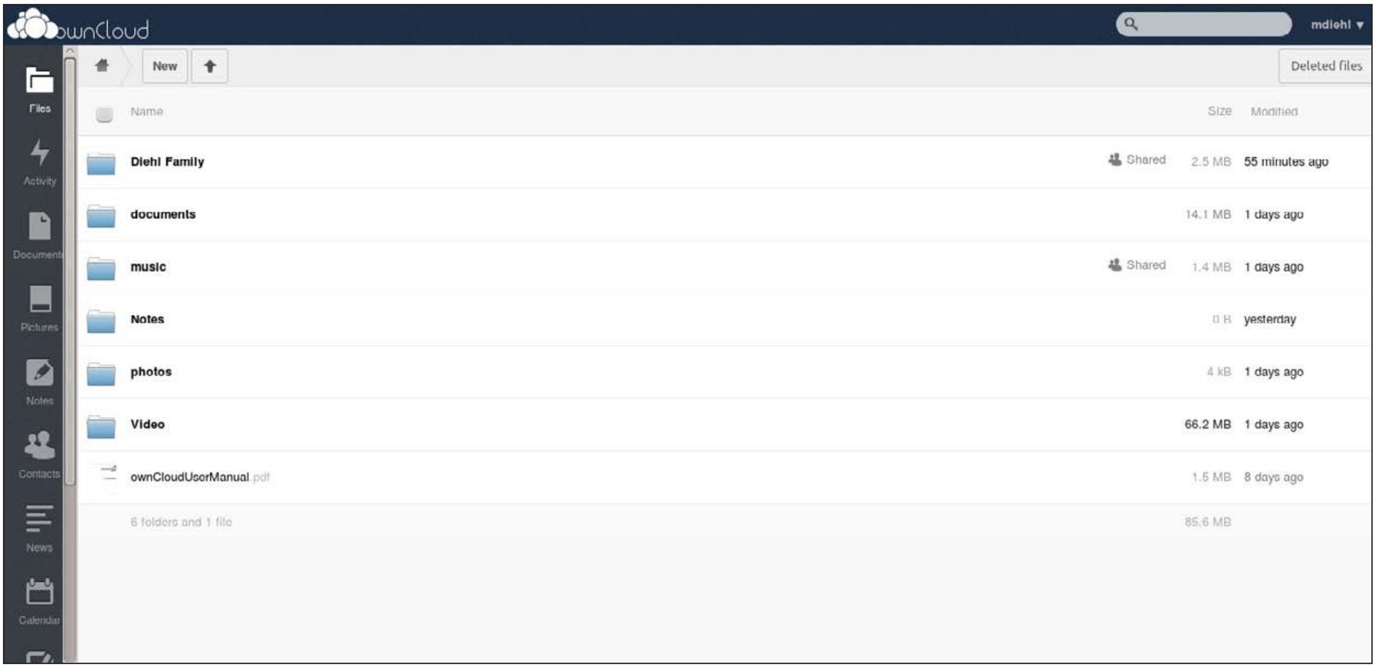
In order to install ownCloud, you need PHP, a Web server and a database server. The installation documentation walks you through configuring the Apache, Lighttpd, Nginx, Yaws or Hiawatha Web servers for use with ownCloud. For a database server, you can choose from MySQL, PostgreSQL or SQLite. It's pretty hard to have a system that doesn't meet those requirements.

The installation process is well documented, so I won't go into too much detail here. Essentially, you download and extract the tarball into a subdirectory under your Web server's htdocs directory. Then you make the Web server configuration changes indicated in the manual and restart the Web server.

Basically, you're setting permissions and enabling cgi execution. Once this is done, you point a Web browser at the new installation and follow the installation wizard. I purposely neglected to make some of the file permission changes, and the wizard notified me that the permissions weren't right. The installation is really pretty straightforward.

After all of the installation is complete, you won't able to access your new ownCloud installation. To resolve this problem, you have to edit ./config/config.php and comment out the `trusted_domains` line. This is a security setting that determines which domains clients are able to connect from, and by default, limits access only to localhost. I happen to think the default values are a bit strict.

After the installation is complete, point a Web browser at your ownCloud server and log in. You will be greeted with a page resembling what is shown in Figure 1. As you can see, the interface is simple. From here, you can access the calendar, contact manager, task list and so on. All of



Name		Size	Modified
Diehl Family	Shared	2.5 MB	55 minutes ago
documents		14.1 MB	1 days ago
music	Shared	1.4 MB	1 days ago
Notes		0 B	yesterday
photos		4 kB	1 days ago
Video		66.2 MB	1 days ago
ownCloudUserManual.pdf		1.5 MB	8 days ago
6 folders and 1 file		85.6 MB	

Figure 1. ownCloud Web Interface

the tools are intuitive to use, but not polished enough that you'd want to use them every day. The intent is that you'd point your favorite PIM at the server and use it as an interface to your shared information.

The initial configuration should be done in a particular order. Since my initial intent was simply to test this system, I managed to do everything in the wrong order. If I had known I would be using the system as a permanent solution, I would have put more thought into its initial implementation. I still ended up with a usable system, but I've made things more complicated and harder to manage than they should have been. Let me share what I did wrong and how it could have been done better.

As soon as you get logged in as the administrator user, you should start creating users and groups. First, I would create the groups. You'll want to create a group for every group of users who need unique access capabilities. For example, I created a group for our family as a whole, and a separate group for each of our businesses. This way, when I create a calendar or address book, I can share it to just my company group, and my wife doesn't have to look at it on her PIM. I initially made the mistake of simply creating a family group and sharing everything to it. But when I created a chore list for the kids, I discovered that they also were able to see my company's calendar, which isn't what I wanted. The moral of the story is to spend the time and keep your groups as granular as possible, because users in the same group can see everything shared to it. Once you've got your groups created, you can create users and assign them to the appropriate group(s) from the pick list. In my case, I created the users first, then I had to go back and assign them to groups, which was tedious.

The moral of the story is to spend the time and keep your groups as granular as possible because users in the same group can see everything shared to it.

Next, you should start creating calendars. I thought I'd be clever and log in as the administrative user and create a family calendar, a calender for each of our businesses and a private calendar for each family member. This sounds reasonable until you discover that each user gets created with his or her own default calendar, which is now redundant. So, use the administrative account to create entity calendars and address books, but let each of your users share their assets themselves.

Then, create a shared documents folder. This is a pretty straightforward process. However, I would recommend that once you've created the shared space, you also create as much of the directory structure as you can reasonably foresee. Otherwise, you end up with a hodge-podge, and users won't be able to find anything when they need it, and that defeats the purpose of a shared file space.

One of the goals of this project is to be able to access the system from the LAN or from the Internet. To make this work from the LAN side, I logged in to my router, which is running OpenWRT, and configured a static hostname, which it was happy to serve to every DNS client on the network. Then, I went to my DNS registrar and configured the same FQDN, but with my router's outside IP address. Then, it was simply a matter of configuring iptables to port-forward TCP/80 to the machine that's hosting ownCloud. A reverse proxy might be more secure, but this works quite well.

I have successfully synchronized my ownCloud with Kontact, Thunderbird, Evolution, my Android phone and our iPad.

Kontact is the easiest to set up. In order to configure address book synchronization, you simply create a new cardDAV address book and point it at http://server.example.com/owncloud/remote.php/carddav/. Kontact happily will discover every shared address book to which your login has access. Similarly, by creating a calDAV calendar and pointing it at http://server.example.com/owncloud/remote.php/caldav/, you'll be able to get all of your calendars configured in one step.

Thunderbird and Evolution are the next easiest clients to configure. However, in these cases, you have to point the client to each *individual* asset. For example, if you have a calendar named "family", you have to point these clients to http://server.example.com/owncloud/remote.php/caldav/calendars/username/family/. You have to do this for each calendar and address book with which you want to synchronize.

To make matters a bit worse, the

structure of the URL changes if the asset was shared by another user. Fortunately, ownCloud will tell you what the correct URL is for each asset. To get this information, simply edit the asset. You will see an icon that looks like a globe. If you click on that, you will be provided with the correct URL.

In order to get the iPad to synchronize, you simply create an account under settings, where it says "Mail, Contact, Calendars", and point it to the same URL mentioned above. This is pretty easy to get working even for a non-Apple user like myself. I don't have an iPhone, but I'm assuming the process is the same.

Synchronizing to the Android device requires additional software. For contact synchronization, I used "CardDAV-Sync free beta". For calendar synchronization, I used "Caldav Sync Free Beta". Once the software is installed, you simply create a corresponding account for each application under Setup. However, you have to point the software to the individual assets, just as you do for Thunderbird and Evolution. There are two potential gotchas though. Automatic synchronization isn't turned on by default, so you have to turn it on and perform an initial synchronization before you will see your calendars and contacts. Also, the Android calendar application supports multiple calendars, but you have to select which ones will be displayed. It doesn't do any good to have a perfectly functioning synchronization system that simply isn't turned on, and don't ask me how I know.

The ownCloud Web site indicates that there is a custom client available that costs $.99. I installed it to see how it works. I was a little disappointed to find that it was simply a webDAV client. I guess I was hoping it would be an integrated calendar, contacts and file client. However, once it was configured, I was able to share files from my Android directly to my file space on my ownCloud server. I did find that the client occasionally lost its configuration and had to be reconfigured, which is a bit tedious. Otherwise, the ownCloud client rounds out almost all of the synchronization features of ownCloud.

I say "almost" because ownCloud also offers a Firefox browser synchronization function. This function is supposed to allow you to synchronize your bookmarks and browser history across multiple machines. However, with the latest version of Firefox, there is no way to point Firefox to the ownCloud server. Perhaps this will be fixed with the next upgrade.

Once everything is configured, there are some operational issues. The obvious issue stems from making concurrent changes to an asset. This results in a conflict, and the various clients handle conflicts differently. To avoid problems, simply synchronize the asset before you modify it, and then re-synchronize when your changes are complete. This will ensure that everyone has the same version of each asset on their client.

I also discovered that it is very difficult to move assets from one calendar or address book to another. The various clients don't seem to do a very good job of this. So far, my attempts at organizing my contacts have resulted in duplicate contacts in different address books. I think the solution is going to involve adding the assets in question to a category, exporting the assets in that category, deleting the assets in that category and the re-importing the assets into the appropriate calendar or address book. This seems like the long way around the block, so I'm going to hold on doing it this way until I know for sure there isn't an easier way to do it.

The rest of the difficulties involve file security. The first problem is that when a user uploads a file into his or her cloud space, that file will be owned by the Web server user. This is okay as long as you don't want to access the file from the filesystem directly or via a Samba share. In those cases, you either have to change the user name that the Web server runs as or the name that the Samba server uses to access the files. Either way, you still won't be able to access the files directly. I've not yet decided on if or how I intend to fix this. I'll probably just access the files via a Samba share or NFS mount.

The ownCloud system supports server-side encryption that can be turned on and off on a per-user basis. This leads to more problems than it's worth, in my opinion. For example, what happens when a user encrypts his or her files and then shares a directory with a user who does not? I happen to know that you get a warning from ownCloud, but I didn't spend the time to find out what actually happens, because I stumbled upon another problem. Server-side encryption pretty much breaks any possible means of file access besides webDAV. I guess that's the point of server-side encryption, but it doesn't work for the way I want/need to access my files. I ended up turning off encryption and decrypting my existing files, which was done seamlessly for me by ownCloud.

The better solution might be to use an encrypted filesystem like Encfs to protect your files. With this solution, you still will be able to use Samba and

NFS to access the plain-text files on the filesystem. Also, you'll be able to upload the encrypted files to another cloud provider, such as Dropbox, as a means of backing up your files without giving up your privacy.

I have found ownCloud to be a very capable and easy-to-manage synchronization server. The actual installation process is pretty simple, so I've spent most of this article pointing out as many of the potential pitfalls as I could. Now that I have it properly configured, I am able to share calendars, contacts and files with other members of my family, no matter where they are or what client they choose to use...and I maintain complete control over my information.■

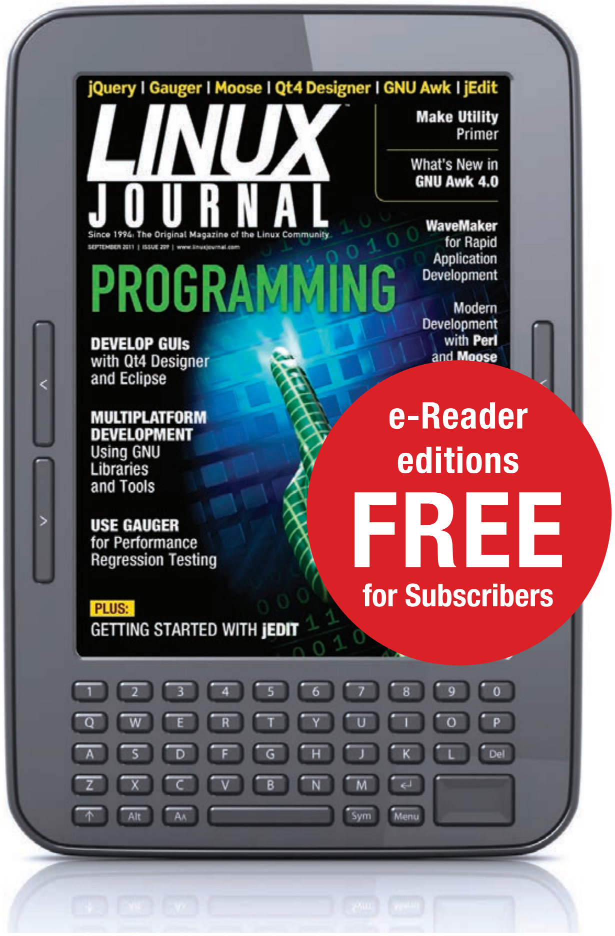
Mike Diehl in an uber-nerd who has been using Linux since the days when Slackware came on 14 5.25" floppy disks and installed kernel version 0.83. He currently operates an Internet telephone company and lives in Blythewood, South Carolina, with his wife and four sons.

Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.

LINUX JOURNAL
on your
e-Reader

Customized
Kindle and Nook
editions
now available

LEARN MORE



WEBCASTS

Learn the 5 Critical Success Factors to Accelerate IT Service Delivery in a Cloud-Enabled Data Center

Today's organizations face an unparalleled rate of change. Cloud-enabled data centers are increasingly seen as a way to accelerate IT service delivery and increase utilization of resources while reducing operating expenses. Building a cloud starts with virtualizing your IT environment, but an end-to-end cloud orchestration solution is key to optimizing the cloud to drive real productivity gains.

> **http://lnxjr.nl/IBM5factors**

Modernizing SAP Environments with Minimum Risk—a Path to Big Data

Sponsor: SAP | Topic: Big Data

Is the data explosion in today's world a liability or a competitive advantage for your business? Exploiting massive amounts of data to make sound business decisions is a business imperative for success and a high priority for many firms. With rapid advances in x86 processing power and storage, enterprise application and database workloads are increasingly being moved from UNIX to Linux as part of IT modernization efforts. Modernizing application environments has numerous TCO and ROI benefits but the transformation needs to be managed carefully and performed with minimal downtime. Join this webinar to hear from top IDC analyst, Richard Villars, about the path you can start taking now to enable your organization to get the benefits of turning data into actionable insights with exciting x86 technology.

> **http://lnxjr.nl/modsap**

WHITE PAPERS

White Paper: JBoss Enterprise Application Platform for OpenShift Enterprise

Sponsor: DLT Solutions

Red Hat's® JBoss Enterprise Application Platform for OpenShift Enterprise offering provides IT organizations with a simple and straightforward way to deploy and manage Java applications. This optional OpenShift Enterprise component further extends the developer and manageability benefits inherent in JBoss Enterprise Application Platform for on-premise cloud environments.

Unlike other multi-product offerings, this is not a bundling of two separate products. JBoss Enterprise Middleware has been hosted on the OpenShift public offering for more than 18 months. And many capabilities and features of JBoss Enterprise Application Platform 6 and JBoss Developer Studio 5 (which is also included in this offering) are based upon that experience.

This real-world understanding of how application servers operate and function in cloud environments is now available in this single on-premise offering, JBoss Enterprise Application Platform for OpenShift Enterprise, for enterprises looking for cloud benefits within their own datacenters.

> **http://lnxjr.nl/jbossapp**

WHITE PAPERS

Linux Management with Red Hat Satellite: Measuring Business Impact and ROI

Sponsor: Red Hat | Topic: Linux Management

Linux has become a key foundation for supporting today's rapidly growing IT environments. Linux is being used to deploy business applications and databases, trading on its reputation as a low-cost operating environment. For many IT organizations, Linux is a mainstay for deploying Web servers and has evolved from handling basic file, print, and utility workloads to running mission-critical applications and databases, physically, virtually, and in the cloud. As Linux grows in importance in terms of value to the business, managing Linux environments to high standards of service quality — availability, security, and performance — becomes an essential requirement for business success.

> **http://lnxjr.nl/RHS-ROI**

Standardized Operating Environments for IT Efficiency

Sponsor: Red Hat

The Red Hat® Standard Operating Environment SOE helps you define, deploy, and maintain Red Hat Enterprise Linux® and third-party applications as an SOE. The SOE is fully aligned with your requirements as an effective and managed process, and fully integrated with your IT environment and processes.

Benefits of an SOE:

SOE is a specification for a tested, standard selection of computer hardware, software, and their configuration for use on computers within an organization. The modular nature of the Red Hat SOE lets you select the most appropriate solutions to address your business' IT needs.

SOE leads to:

- Dramatically reduced deployment time.
- Software deployed and configured in a standardized manner.
- Simplified maintenance due to standardization.
- Increased stability and reduced support and management costs.
- There are many benefits to having an SOE within larger environments, such as:
  - Less total cost of ownership (TCO) for the IT environment.
  - More effective support.
  - Faster deployment times.
  - Standardization.

> **http://lnxjr.nl/RH-SOE**

ZFS and BTRFS

BTRFS and ZFS are two options for protecting against data corruption. Which one should you use, and how should you use it?

RUSSELL COKER

For a long time, the software RAID implementation in the Linux kernel has worked well to protect data against drive failure. It provides great protection against a drive totally failing and against the situation where a drive returns read errors. But what it doesn't offer is protection against silent data corruption (where a disk returns corrupt data and claims it to be good). It also doesn't have good support for the possibility of drive failure during RAID reconstruction.

Drives have been increasing in size significantly, without comparable increases in speed. Modern drives have contiguous read speeds 300 times faster than drives from 1988 but are 40,000 times larger (I'm comparing a recent 4TB SATA disk with a 100M ST-506 disk that can sustain 500K/s reads). So the RAID rebuild time is steadily increasing, while the larger storage probably increases the risk of data corruption.

Currently, there are two filesystems available on Linux that support internal RAID with checksums on all data to prevent silent corruption: ZFS and BTRFS. ZFS is from Sun and has some license issues, so it isn't included in most Linux distributions. It is available from the ZFS On Linux Web site (**http://zfsonlinux.org**). BTRFS has no license problems and is included in most recent distributions, but it is at an earlier stage of development. When discussing BTRFS in this article, I concentrate on the theoretical issues of data integrity and not the practical issues of kernel panics (which happen regularly to me but don't lose any data).

Do Drives Totally Fail?

For a drive totally to fail (that is, be unable to read any data successfully at all), the most common problem used to be "stiction". That is when the heads stick to the platters, and the drive motor is unable to spin the disk. This seems to be very uncommon in recent times. I presume that drive manufacturers fixed most of the problems that caused it.

In my experience, the most common reason for a drive to become totally unavailable is due to motherboard problems, cabling or connectors—that is, problems outside the drive. Such problems usually can be fixed but may cause some downtime, and the RAID array needs to keep working with a disk missing.

Serious physical damage (for example, falling on concrete) can cause a drive to become totally unreadable. But, that isn't a problem that generally happens to a running RAID array. Even when I've seen drives fail due to being in an uncooled room in an Australian summer, the result has been many bad sectors, not a drive that's totally unreadable. It seems that most drive "failures" are really a matter of an increasing number of bad sectors.

There aren't a lot of people who can do research on drive failure. An individual can't just buy a statistically significant number of disks and run them in servers for a few years. I couldn't find any research on the incidence of excessive bad sectors vs. total drive failure. It's widely regarded that the best research on the incidence of hard drive "failure" is the Google Research paper "Failure Trends in a Large Disk Drive Population" (**http://research.google.com/pubs/pub32774.html**), which although very informative, gives no information on the causes of "failure". Google defines "failure" as anything other than an upgrade that causes a drive to be replaced. Not only do they not tell us the number of drives that totally died vs. the number that had some bad sectors, but they also don't tell us how many bad sectors would be cause for drive replacement.

Lakshmi N. Bairavasundaram, Garth R. Goodson, Bianca Schroeder, Andrea C. Arpaci-Dusseau and Remzi H. Arpaci-Dusseau from the University of Wisconsin-Madison wrote a paper titled "An Analysis of Data Corruption in the Storage Stack" (**http://research.cs.wisc.edu/adsl/Publications/corruption-fast08.html**). That paper gives a lot of information about when drives have corrupt data, but it doesn't provide much information about the case of major failure (tens of thousands of errors), as distinct from cases where there are dozens or hundreds of errors. One thing it does say is that the 80th percentile of latent sector errors per disk with errors is "about 50", and the 80th percentile of checksum mismatches for disks with errors is "about 100". So most disks with errors have only a very small number of errors. It's worth noting that this research was performed with data that NetApp obtained by analyzing the operation of its hardware in the field. NetApp has a long history of supporting a large number of disks in many sites with checksums on all stored data.

I think this research indicates that the main risks of data loss are corruption on disk or a small number of read errors, and that total drive failure is an unusual case.

Redundancy on a Single Disk

By default, a BTRFS filesystem that is created for a single device that's not an SSD will use "dup" mode for metadata. This means that every metadata block will be written to two parts of the disk. In practice, this can allow for recovering data from drives with many errors. I recently had a 3TB disk develop about 14,000 errors. In spite of such a large number of errors, the duplication of metadata meant that there was little data loss. About 2,000 errors in metadata blocks were corrected with the duplicates, and the 12,000 errors in data blocks (something less than 48M of data) was a small fraction of a 3TB disk. If an older filesystem was used in that situation, a metadata error could corrupt a directory and send all its entries to lost+found.

ZFS supports even greater redundancy via the `copies=` option. If you specify `copies=2` for a filesystem, then every data block will be written to two different parts of the disk. The number of copies of metadata will be one greater than the number of copies of data, so `copies=2` means that there will be three copies of every metadata block. The maximum number of copies for data blocks in ZFS is three, which means that the maximum number of copies of metadata is four.

The paper "An Analysis of Data Corruption in the Storage Stack" shows that for "nearline" disks (that is, anything that will be in a typical PC or laptop), you can expect a 9.5% probability of read errors (latent sector errors) and a 0.466% probability of silent data corruption (checksum mismatches). Typical *Linux Journal* readers probably can expect to see data loss from hard drive read errors on an annual basis from the PCs owned by their friends and relatives. The probability of silent data corruption is low enough that all users have a

This option of providing extra protection for data is a significant benefit for ZFS when compared to BTRFS.

less than 50% chance of seeing it on their own PCs during their lives—unless they purchased one of the disks with a firmware bug that corrupts data.

If you run BTRFS on a system with a single disk (for example, a laptop), you can expect that if the disk develops any errors, they will result in no metadata loss due to duplicate metadata, and any file data that is lost will be reported to the application by a file read error. If you run ZFS on a single disk, you can set `copies=2` or `copies=3` for the filesystem that contains your most important data (such as /home on a workstation) to decrease significantly the probability that anything less than total disk failure will lose data. This option of providing extra protection for data is a significant benefit for ZFS when compared to BTRFS.

If given a choice between a RAID-1 array with Linux software RAID (or any other RAID implementation that doesn't support checksums) and a single disk using BTRFS, I'd choose the single disk with BTRFS in most cases. That is because on a single disk with BTRFS, the default configuration is to use "dup" for metadata. This means that a small number of disk errors will be unlikely to lose any metadata, and a scrub will tell you which file data has been lost due to errors. Duplicate metadata alone can make the difference between a server failing and continuing to run. It is possible to run with "dup" for data as well, but this isn't a well supported configuration (it requires mixed data and metadata chunks that require you to create a very small filesystem and grow it).

It is possible to run RAID-1 on two partitions on a single disk if you are willing to accept the performance loss. I have a 2TB disk running as a 1TB BTRFS RAID-1, which has about 200 bad sectors and no data loss.

Finally, it's worth noting that a "single disk" from the filesystem perspective can mean a RAID array. There's nothing wrong with running BTRFS or ZFS over a RAID-5 array. The metadata duplication that both those filesystems offer will reduce the damage if a RAID-5 array suffers

When you replace a disk in Linux software RAID, the old disk will be marked as faulty first, and all the data will be reconstructed from other disks.

a read error while replacing a failed disk. A hardware RAID array can offer features that ZFS doesn't offer (such as converting from RAID-1 to RAID-5 and then RAID-6 by adding more disks), and hardware RAID arrays often include a write-back disk cache that can improve performance for RAID-5/6 significantly. There's also nothing stopping you from using BTRFS or ZFS RAID-1 over a pair of hardware RAID-5/6 arrays.

Drive Replacement

When you replace a disk in Linux software RAID, the old disk will be marked as faulty first, and all the data will be reconstructed from other disks. This is fine if the other disks are all good, but if the other disks have read errors or corrupt data, you will lose data. What you really need is to have the new disk directly replace the old disk, so the data for the new disk can be read from the old disk or from redundancy in the array, whichever works.

ZFS has a `zpool replace` command that will rebuild the array from the contents of the old disk and from the other disks in a redundant set. BTRFS supports the same thing with the `btrfs replace` command. In the most common error situations (where a disk has about 50 bad sectors), this will give you the effect of having an extra redundant disk in the array. So a RAID-5 array in BTRFS or in ZFS (which they call a RAID-Z) should give as much protection as a RAID-6 array in a RAID implementation that requires removing the old disk before adding a new disk. At this time, RAID-5 and RAID-6 support in BTRFS is still fairly new, and I don't expect it to be ready to use seriously by the time this article is published. But the design of RAID-5 in BTRFS is comparable to RAID-Z in ZFS, and they should work equally well when BTRFS RAID-5 code has been adequately tested and debugged.

Hot-spare disks are commonly used to allow replacing a disk more quickly than someone can get to the server. The idea is that the RAID array might be reconstructed before anyone even can get to it. But it seems to me that the real benefit of a hot-spare when used with a modern filesystem, such as ZFS or BTRFS, is that the system has the ability to read from the disk with errors as well as the rest of the array while constructing the new disk. If you have a server where every disk bay contains an active disk (which is a very common configuration in my experience), it is unreasonably difficult to support a disk replacement operation that reads from the failing disk (using an eSATA device for the rebuild isn't easy). Note that BTRFS doesn't have automatic hot-spare support yet, but it presumably will get it eventually. In the meantime, a sysadmin has to instruct it to replace the disk manually.

As modern RAID systems (which on Linux servers means ZFS as the only fully functional example at this time) support higher levels of redundancy, one might as well use RAID-Z2 (the ZFS version of RAID-6) instead of RAID-5 with a hot-spare, or a RAID-Z3 instead of a RAID-6 with a hot-spare. When a disk is being replaced in a RAID-6/RAID-Z2 array with no hot-spare, you are down to a RAID-5/RAID-Z array, so there's no reason to use a disk as a hot-spare instead of using it for extra redundancy in the array.

How Much Redundancy Is Necessary?

The way ZFS works is that the `copies=` option (and the related metadata duplication) is applied on top of the RAID level that's used for the storage "pool". So if you use `copies=2` on a ZFS filesystem that runs on a RAID-1, there will be two copies of the data on each of the disks. The allocation of the copies is arranged such that it covers different potential failures to the RAID level, so if you had `copies=3` for data stored on a three-disk RAID-Z pool, each disk in the pool would have a copy of the data (and parity to help regenerate two other copies). The amount of space required for some of these RAID configurations is impractical for most users. For example, a RAID-Z3 array composed of six 1TB disks would have 3TB of RAID-Z3 capacity. If you then made a ZFS filesystem with `copies=3`, you would get 1TB of usable capacity out of 6TB of disks. 5/6 disks is more redundancy than most users need.

If data is duplicated in a RAID-1 array, the probability of two disks having errors on matching blocks from independent random errors is going to be very low. The paper from the University of Wisconsin-Madison notes that firmware bugs

In many deployments, the probability of the server being stolen or the building catching on fire will be greater than the probability of a RAID-Z2 losing data.

can increase the probability of corrupt data on matching blocks and suggests using staggered stripes to cover that case. ZFS does stagger some of its data allocation to deal with that problem. Also, it's fairly common for people to buy disks from two different companies for a RAID-1 array to prevent a firmware bug or common manufacturing defect from corrupting data on two identical drives. The probability of both disks in a BTRFS RAID-1 array having enough errors that data is lost is very low. With ZFS, the probability is even lower due to the mandatory duplication of metadata on top of the RAID-1 configuration and the option of duplication of data. At this time, BTRFS doesn't support duplicate metadata on a RAID array.

The probability of hitting a failure case that can't be handled by RAID-Z2 but that can be handled by RAID-Z3 is probably very low. In many deployments, the probability of the server being stolen or the building catching on fire will be greater than the probability of a RAID-Z2 losing data. So it's worth considering when to spend more money on extra disks and when to spend money on better off-site backups.

In 2007, Val Bercovici of NetApp suggested in a StorageMojo interview that "protecting online data only via RAID-5 today verges on professional malpractice" (**http://storagemojo.com/2007/02/26/netapp-weighs-in-on-disks**). During the past seven years, drives have become bigger, and the difficulties we face in protecting data have increased. While Val's claim is hyperbolic, it does have a basis in fact. If you have only the RAID-5 protection (a single parity block protecting each stripe), there is a risk of having a second error before the replacement disk is brought on-line. However, if you use RAID-Z (the ZFS equivalent of RAID-5), every metadata block is stored at least twice in addition to the RAID-5 type protection, so if a RAID-Z array entirely loses a disk and then has a read error on one of the other disks, you might lose some data but won't lose metadata. For metadata to be lost on a RAID-Z array, you need to have one disk die entirely and then have matching read errors on two other disks. If disk failures are independent, it's a very unlikely scenario. If, however, the disk failures are not independent, you could have a problem with all disks (and lose no matter what type of RAID you use).

Snapshots

One nice feature of BTRFS and ZFS is the ability to make snapshots of BTRFS subvolumes and ZFS filesystems. It's not difficult to write a cron job that makes a snapshot of your important data every hour or even every few minutes. Then when you accidentally delete an important file, you easily can get it back. Both BTRFS and ZFS can be configured such that files can be restored from snapshots without root access so users can recover their own files without involving the sysadmin.

Snapshots aren't strictly related to the the topic of data integrity, but they solve the case of accidental deletion, which is the main reason for using backups. From a sysadmin perspective, snapshots and RAID are entirely separate issues. From the CEO perspective, "is the system working or not?", they are part of the same issue.

Comparing BTRFS and ZFS

For a single disk in a default configuration, both BTRFS and ZFS will store two copies of each metadata block. They also use checksums to detect when data is corrupted, which is much better than just providing corrupt data to an application and allowing errors to propagate. ZFS supports storing as many as three copies of data blocks on a single disk, which is a significant benefit.

For a basic RAID-1 installation, BTRFS and ZFS offer similar features by default (storing data on both devices with checksums to cover silent corruption). ZFS offers duplicate metadata as a mandatory feature and the option of duplicate data on top of the RAID configuration.

BTRFS supports RAID-0, which is a good option to have when you are working with data that is backed up well. The combination of the use of BTRFS checksums to avoid data corruption and RAID-0 for performance would be good for a build server or any other system that needs large amounts of temporary file storage for repeatable jobs but for which avoiding data corruption is important.

BTRFS supports dynamically increasing or decreasing the size of the filesystem. Also, the filesystem can be rebalanced to use a different RAID level (for example, migrating between RAID-1 and RAID-5). ZFS, however, has a very rigid way of managing storage. For example, if you have a RAID-1 array in a pool, you can never remove it, and you can grow it only by replacing all the disks with larger ones. Changing between RAID-1 and RAID-Z in ZFS requires a backup/format/restore operation, while on BTRFS, you can just add new disks and rebalance.

ZFS supports different redundancy levels (via the `copies=` setting) on different "filesystems" within the same "pool" (where a "pool" is group of one or more RAID sets). BTRFS "subvolumes" are equivalent in design to ZFS "filesystems", but BTRFS doesn't support different RAID parameters for subvolumes at this time.

ZFS supports RAID-Z and RAID-Z2, which are equivalent to BTRFS RAID-5, RAID-6—except that RAID-5 and RAID-6 are new on BTRFS, and many people aren't ready to trust important data to them. There is no feature in BTRFS or planned for the near future that compares with RAID-Z3 on ZFS. There are plans for future development of extreme levels of redundancy in BTRFS at some future time, but it probably won't happen soon.

Generally, it seems that ZFS is designed to offer significantly greater redundancy than BTRFS supports, while BTRFS is designed to be easier to manage for smaller systems.

Currently, BTRFS doesn't give good performance. It lacks read optimization for RAID-1 arrays and doesn't have any built-in support for using SSDs to cache data from hard drives. ZFS has many performance features and is as fast as a filesystem that uses so much redundancy can be.

Finally, BTRFS is a new filesystem, and people are still finding bugs in it—usually not data loss bugs but often bugs that interrupt service. I haven't yet deployed BTRFS on any server where I don't have access to the console, but I have Linux servers running ZFS in another country.■

Russell Coker has been working on NSA Security Enhanced Linux since 2001 and has been working on the Bonnie++ benchmark suite since 1999.

Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.

MySQL® CONFERENCE LONDON

PERCONA LIVE

Introducing pi-web-agent, a Raspberry Pi Web App

A Web application allowing everyday users to control the Raspberry Pi.

VASILIS NICOLAOU, ANGELOS GEORGIADIS, GEORGIOS CHAIREPETIS and ANDREAS GALAZIS

The pi-web-agent is a Web application that aims to give new users experience with the Raspberry Pi and potentially with any Linux distribution on any architecture. Raspberry Pi has introduced Linux to a lot of people, and we want to enhance their experience further. This project also demonstrates the extensibility capabilities of Linux.

What we provide enables you to install the pi-web-agent as soon as you have Raspbian (**http://www.raspbian.org**) installed. You'll simply be able to open your usual browser from your everyday machine and start interacting with your Pi.

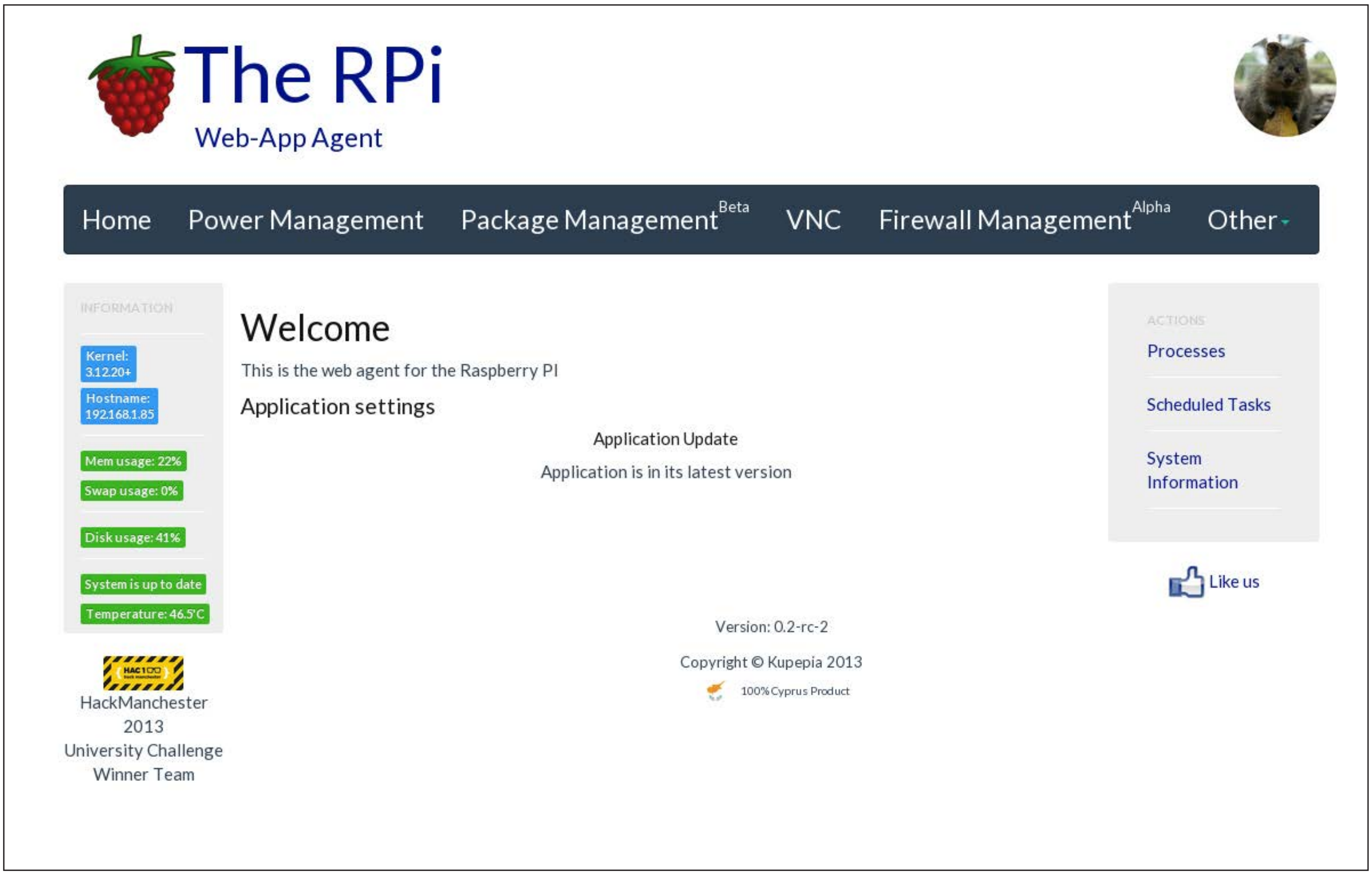
The Product

The idea behind pi-web-agent is to support desktop environment functionality on browsers and provide extensions that behave similarly to mainstream desktop applications. If you have used GNOME or KDE, you will have noticed that each provides its own set of applications.

pi-web-agent is similar to Webmin (**http://www.webmin.com**), with the difference being that pi-web-agent targets everyday users. We want to give users a desktop experience through their browsers.

Main Features

The moment you launch



INFORMATION

Kernel: 3.12.20+

Hostname: 192.168.1.85

Mem usage: 22%

Swap usage: 0%

Disk usage: 41%

System is up to date

Temperature: 46.5'C

HackManchester 2013 University Challenge Winner Team

Welcome

This is the web agent for the Raspberry PI

Application settings

Application Update

Application is in its latest version

Version: 0.2-rc-2

Copyright © Kupepia 2013

100% Cyprus Product

Figure 1. pi-web-agent Home Screen

pi-web-agent (after changing your password of course), you will realize that it already provides a lot of functionality despite its youth. On the left-hand side, you'll see some information concerning your Pi, such as its temperature, kernel version, update notifications, disk and memory usage.

The home screen (Figure 1) will check whether an update for the pi-web-agent is available. If it is, a button will appear that will download and install the new version of the application for you.

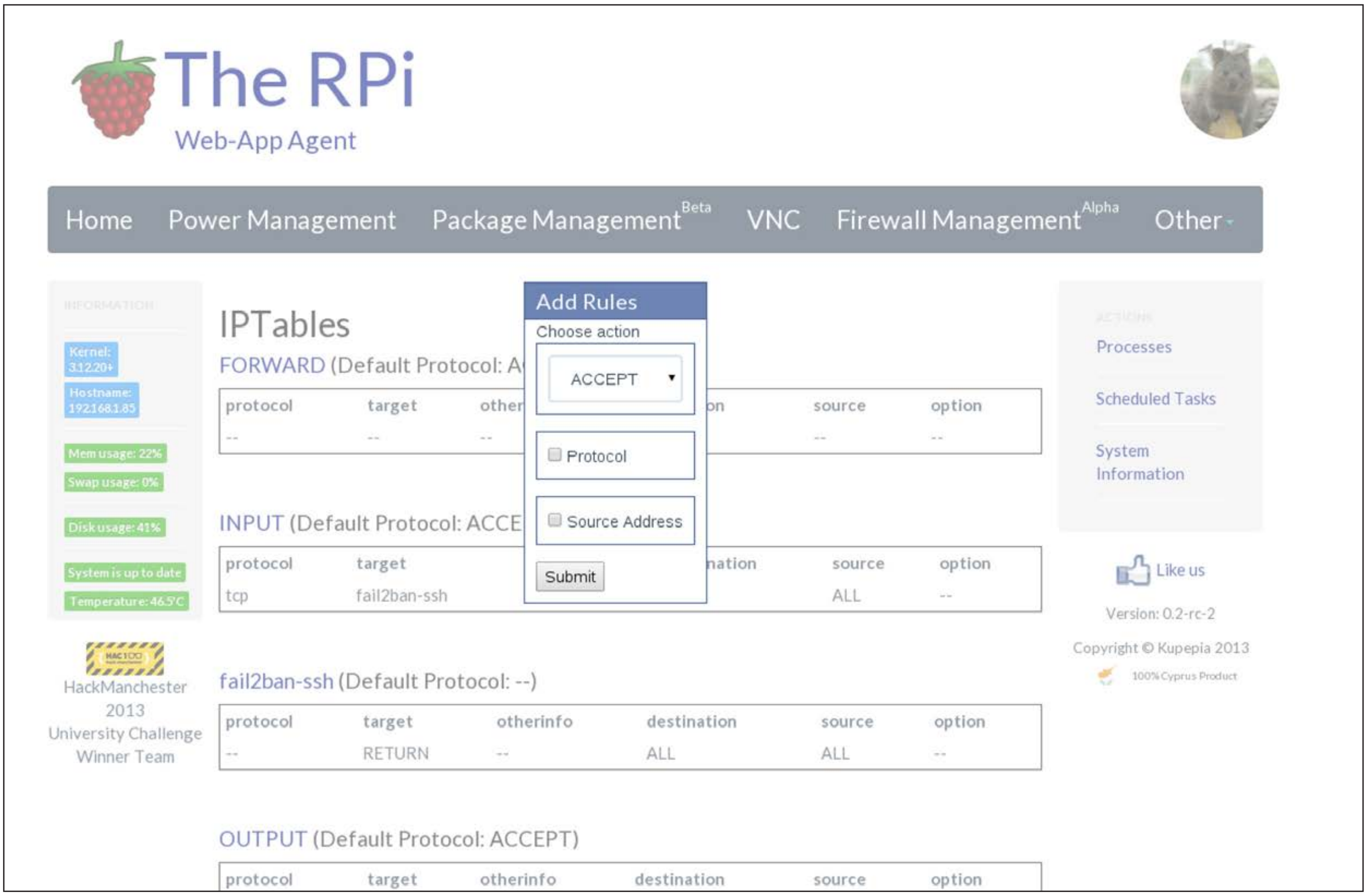
The navigation menu reveals the standard extensions that every desktop environment provides for users to interact with the underlying operating system, including power management for shutting down and restarting, and a package management section where we provide some of our favourite and useful applications with the capability to install them, enabling you to get started fast. One of the most important extensions is VNC, which provides the TightVNC server and gives you access to your real Pi

One of the most important extensions is VNC, which provides the TightVNC server and gives you access to your real Pi desktop with the Glavsoft TightVNC client Java applet.

desktop with the Glavsoft TightVNC client Java applet.

The firewall management, despite its early stage, enables you to avoid the fuss of many complicated options that iptables provides from the command line (until you become an expert). You can set rules for various chains, block certain IP addresses or allow connections through different protocols (Figure 2).

Clicking the "Other" tab will reveal more extensions. The camera controller enables you to take snapshots and



IPTables

FORWARD (Default Protocol: A

protocol	target	other	on	source	option
--	--	--		--	--

INPUT (Default Protocol: ACCE

protocol	target	nation	source	option
tcp	fail2ban-ssh		ALL	--

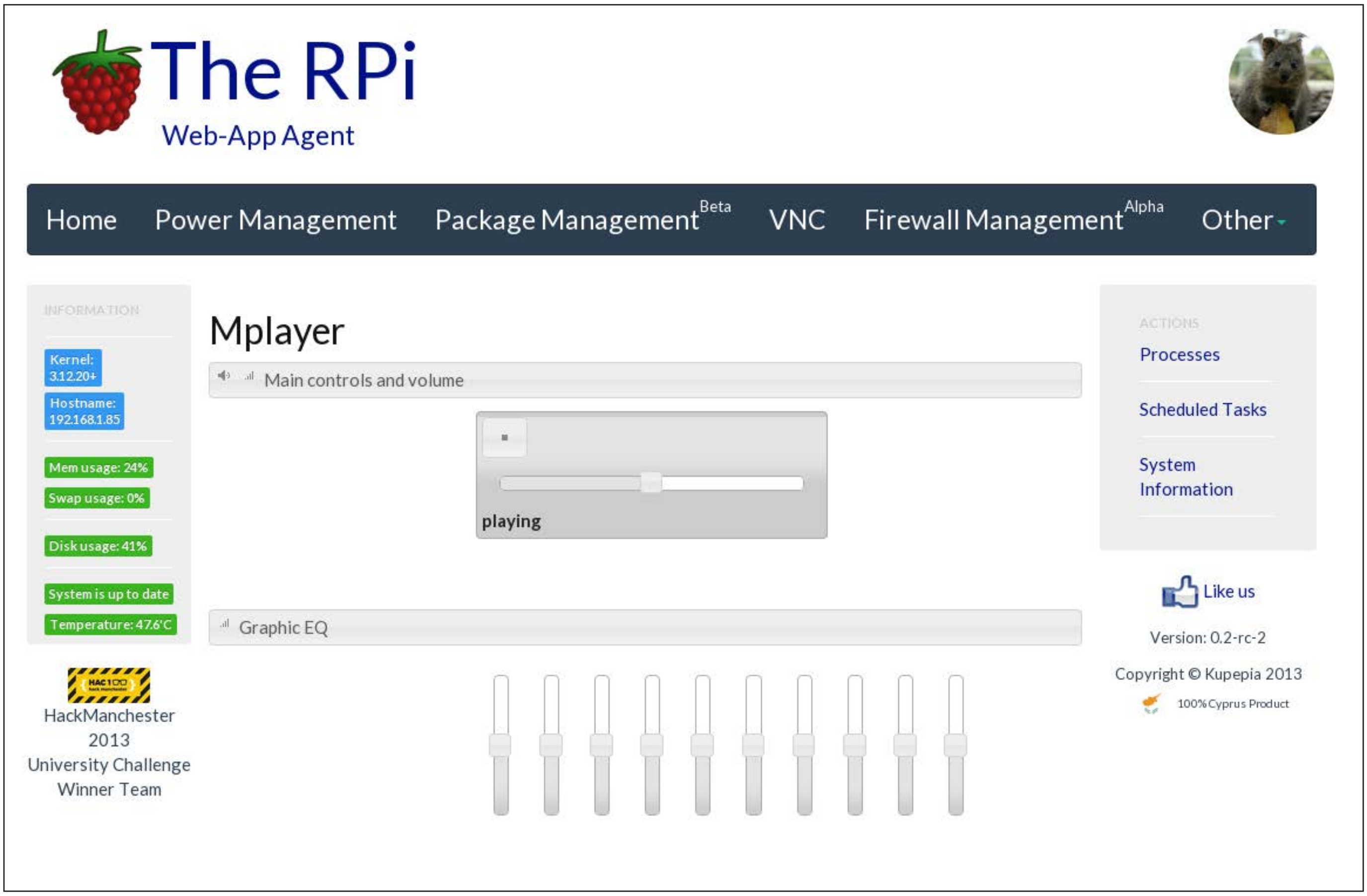
fail2ban-ssh (Default Protocol: --)

protocol	target	otherinfo	destination	source	option
--	RETURN	--	ALL	ALL	--

OUTPUT (Default Protocol: ACCEPT)

protocol	target	otherinfo	destination	source	option

Figure 2. Firewall View When Clicking on a Certain Chain



Kernel: 3.12.20+

Hostname: 192.168.1.85

Mem usage: 24%

Swap usage: 0%

Disk usage: 41%

System is up to date

Temperature: 47.6'C

playing

Version: 0.2-rc-2

Copyright © Kupepia 2013

100% Cyprus Product

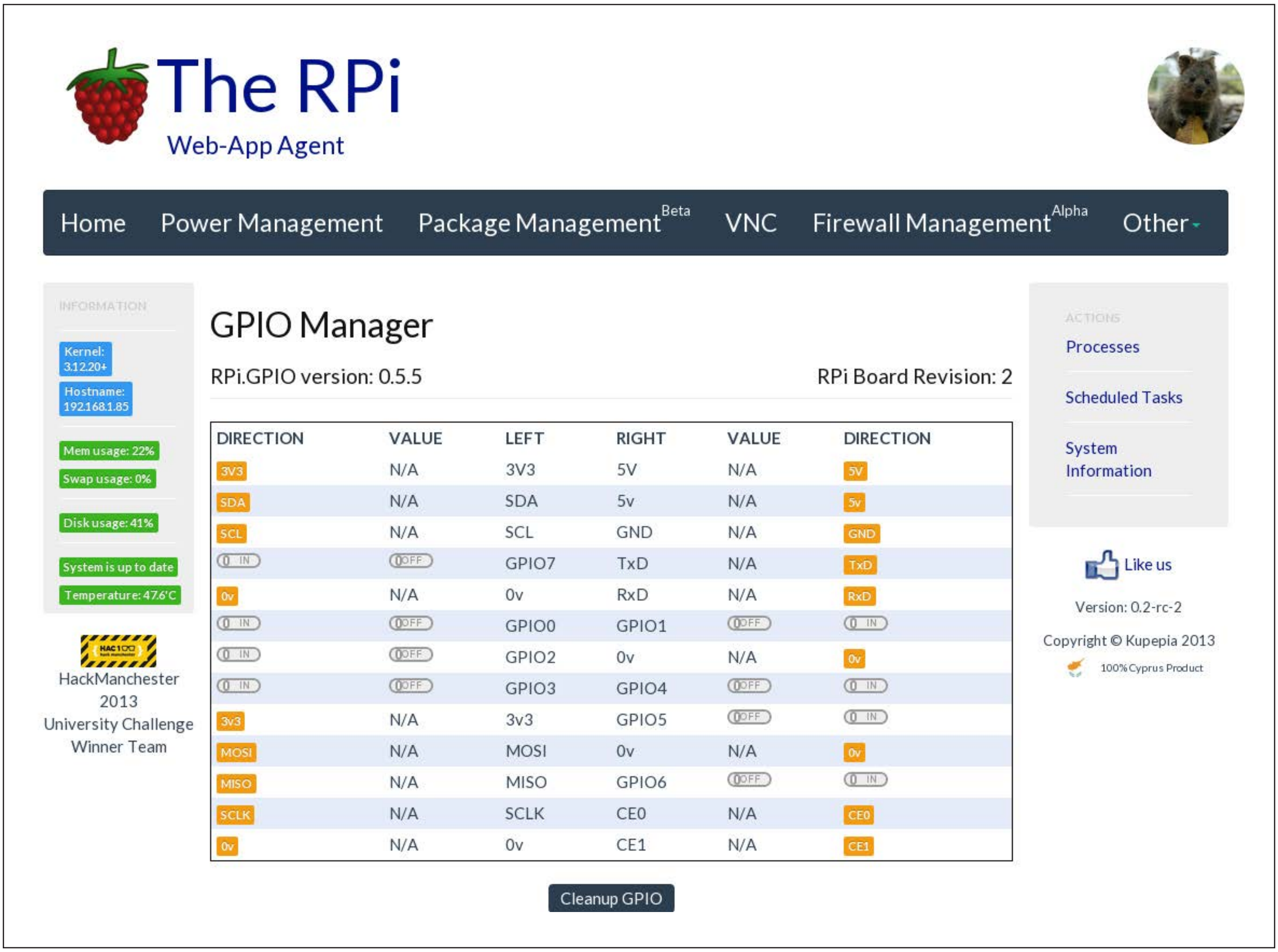
Figure 3. Media Player View When Listening to an Audio File or Streaming

even begin your own live stream! We also provide a media player (tagged as radio, since it started as such), where you can provide a URI of an audio file or a streaming radio channel. Your Raspberry Pi will start playing the audio, so get ready to attach your HD speakers!

You also can play an audio file straight from your Pi, but don't bother typing the URI in the text box. Find it through the file browser we provide. Although it's simple in functionality for now, it enables you to browse through your files, and download or choose to play audio files with the pi-web-agent's media player extension. (Note: this functionality is available only from the development branch on GitHub, but it will be available in version 0.3.)

The media player extension is an Mplayer port that also enables you to control the sound with an equalizer (Figure 3).

If you want to be more hardcore and play with some wires and LEDs, we provide an extension for controlling the GPIO pins on your Pi



Kernel: 3.12.20+

Hostname: 192.168.1.85

Mem usage: 22%

Swap usage: 0%

Disk usage: 41%

System is up to date

Temperature: 47.6'C

HackManchester 2013 University Challenge Winner Team

GPIO Manager

RPi.GPIO version: 0.5.5

RPi Board Revision: 2

DIRECTION	VALUE	LEFT	RIGHT	VALUE	DIRECTION
3V3	N/A	3V3	5V	N/A	5V
SDA	N/A	SDA	5v	N/A	5v
SCL	N/A	SCL	GND	N/A	GND
0 IN	0 OFF	GPIO7	TxD	N/A	TxD
0v	N/A	0v	RxD	N/A	RxD
0 IN	0 OFF	GPIO0	GPIO1	0 OFF	0 IN
0 IN	0 OFF	GPIO2	0v	N/A	0v
0 IN	0 OFF	GPIO3	GPIO4	0 OFF	0 IN
3v3	N/A	3v3	GPIO5	0 OFF	0 IN
MOSI	N/A	MOSI	0v	N/A	0v
MISO	N/A	MISO	GPIO6	0 OFF	0 IN
SCLK	N/A	SCLK	CE0	N/A	CE0
0v	N/A	0v	CE1	N/A	CE1

Version: 0.2-rc-2

Copyright © Kupepia 2013

100% Cyprus Product

Figure 4. The GPIO module gives you control over the Raspberry Pi GPIO pins.

(check out the interface in Figure 4). You also can check for updates or update your system and turn services on or off. There is more to come, so keep yourself posted via our Facebook page (**https://www.facebook.com/piwebagent**). We want you to forget you are using a Web browser and not bother clicking the VNC option.

How It Works

The pi-web-agent is served by Apache (**http://www.apache.org**). Some claim that Apache is big and heavy, but we believe it has the most chances of incorporating cutting-edge technology. If you don't know what this means, check out mod_spdy with the SPDY protocol from Google (**https://code.google.com/p/mod-spdy**). We also can choose among a variety of modules that can increase pi-web-agent's potential. For those of you who want a lighter HTTP server, we are going to release a pi-web-agent modification after the API is complete.

The core is written in Python and interacts via the Common Gateway Interface (CGI) to provide dynamic content. Our goal is also to provide an API, so almost every module is able to generate JSON data, making the application highly extensible both for us and for third-party developers. This will give us a lot of flexibility for future products we plan to deliver.

The styling currently is achieved with bootstrap (**http://getbootstrap.com**) using the Flatly (**http://bootswatch.com/flatly**) theme. User interaction mainly is achieved via JavaScript calls and rendering the document on the client side (most of the time). Our rule of thumb is to move as much processing from the Raspberry (server side) to the more powerful machine that runs the browser (client side).

Current Status

pi-web-agent is in a very early stage. It started in October 2013 at HackManchester (**http://www.hackmanchester.com**), winning the University challenge prize. The first version (0.1) was released on December 27, 2013, and the second release (0.2) followed in April 2014.

The current stage of development has three phases:

- The development of version 0.3, which will introduce framework improvements. This also involves better extension management. As the number of provided extensions grows, we want users to choose what they want, and install and uninstall them at will. This will significantly reduce long installation times caused by dependencies (dependencies of the camera controller introduced 100MB of dependency packages).

- The development of the pi-web-agent for Android, which also includes the optimization of the API.

- The design and development of the pi-web-agent version 1.0. We want this to look like a real desktop environment, and we also want to achieve this in no more than a year.

pi-web-agent is open source, and it's easy for you to get involved—just fork our main git repository (**http://www.github.com/vaslabs/pi-web-agent**), and send us your changes through pull requests.

Using pi-web-agent

Imagine your Raspberry Pi has

just arrived and you have installed Raspbian on your SD card. Even if you don't have much experience with Linux and the command line, worry no more. You can connect to your Pi with SSH and install the pi-web-agent, which will help you in your first steps. While you become more experienced with the Pi and Linux, the pi-web-agent will grow with you, giving you more powerful capabilities and making your interaction with your Pi more enjoyable.

The most difficult task you'll face is the installation process, especially if you run a headless Debian distribution on your Raspberry Pi. You won't be able to avoid executing commands (until we release a Raspbian mod with the pi-web-agent included). You need to connect with your Pi via SSH. There are two ways to install the pi-web-agent, which are described below.

Installing through pistore

If you are using a Linux machine, it's easy. Just do:

```
ssh -X pi@raspberrypi
```

The -X will enable you to execute graphical applications. Provide your password to the prompt, and then launch the pistore by typing the following and then pressing Enter:

```
pistore
```

When pistore opens, just register and search for pi-web-agent. Everything else is straightforward.

Installing via the Command Line

If you are not on a Linux machine, or if your distribution is headless, you still can install pi-web-agent easily. The following commands will fetch and install pi-web-agent:

```
wget https://github.com/vaslabs/\
    pi-web-agent/archive/0.2-rc-1.zip
unzip 0.2-rc-1.zip
cd pi-web-agent-0.2-rc-1
./install.sh
./run.sh
```

Troubleshooting

We've started a discussion on Reddit that covers a lot of troubleshooting, thanks to users' questions (**http://www.reddit.com/r/raspberry_pi/comments/249j4r/piwebagent_control_your_pi_from_the_ease_of_your**). You can find guidelines on how to install under various circumstances and how to resolve problems that others already have faced. All the issues identified in this discussion have been resolved, but if you face a new one,

It is possible to extend the pi-web-agent by adding new Python modules.

just post a new comment.

Supported Platforms

The pi-web-agent framework is based on the micro-CernVM (Scientific Linux) appliance agent framework developed at CERN in summer 2013 (**https://github.com/cernvm/cernvm-appliance-agent**). We developed pi-web-agent based on that framework. We've modified it to work on Raspbian and cover the needs of the Raspberry Pi users. However, it is possible to use it on various Linux distributions with minor modifications concerning Apache configuration and replacing Raspberry Pi-specific modules, such as the Update and the GPIO.

We plan to release pi-web-agent version 1.0 for Raspbian, Pidora and Arch. Until then, the only officially supported platform is Raspbian.

Developing on pi-web-agent

It is possible to extend pi-web-agent by adding new Python modules. Upon creating a Python module, you'll find that the best way to work is to follow the structure below:

```
if 'MY_HOME' not in os.environ:
    os.environ['MY_HOME']=
        '/usr/libexec/pi-web-agent'
sys.path.append(os.environ['MY_HOME']+
                    '/etc/config')
from framework import output


def main():
    output('Title', 'Hello my first module')


if __name__ == "__main__":
    main()
```

There are a lot of modules and methods you can use to get the most out of the framework. The most important is the output, which takes care of what's appearing on your browser. You can give two arguments, the title and the HTML, as the main content of your extension.

Framework Overview

The framework is composed of various Python modules and configuration files. The configuration files initially were in XML format, but they have been converted to JSON format, which reduced the codebase by a significant amount.

The core module is framework.py, placed in the /usr/libexec/pi-web-agent/etc/ config directory. This module uses view.py and menu.py, which also use HTMLPageGenerator.py and BlueprintDesigner.py to construct the Web site skeleton. All the information about which modules are in use is in config.cfg in the same directory with the framework.py.

Adding Features

When you create your first module, you'll need to register it to the config.cfg file in order to be placed in the navigation menu. You'll also find that you can declare its version as Alpha or Beta. More options will be added soon, such as dependencies (next version 0.3) of a corresponding feature.

When a feature is added to the configuration file, the framework places it in the navigation menu with the URL provided in that file. There are two types of URL links: one for reloading the whole page and one for updating just the extension view (append `?type=js`). Since version 0.2, we started using the second format, and the first format is deprecated.

When clicking to select a feature, a JavaScript routine is triggered that loads the content of the extension in the appropriate area, where the user can interact with it. It is important to note that all user interface renderings will be performed on the client side exclusively by version 1.0.

The Future

Reading through this article, you will have noticed that there are a lot of things pending and even more that can be improved. This is our goal: to develop a solid application, not only to satisfy users, but also to provide a good environment for other developers to extend or build on top of the pi-web-agent. That's why we have started multiple spin-off projects.

We have started a bunch of help projects in order to make life easier for us, users and third-party developers. We created a benchmark (**https://github.com/azardilis/testing-fw**) that gives us the loading times of each feature. We also started a plugin for the gedit text editor (the "official" text editor of our dev team) to automate the creation and deployment of pi-web-agent extensions.

Last but not least, we are developing pi-web-agent for Android. Not only have we started this application to increase user satisfaction, but it also is the driver for the pi-web-agent API, which will be given out officially, ready and documented, for third-party developers to build on. In addition, the API will be solely used for the creation of pi-web-agent version 1.0.

What Needs to Be Done

pi-web-agent is in an early stage, but it needs support from special tools to speed up the development process. Now that the framework has started to be more stable in terms of changes, we need to finish the gedit development plugin.

Next, we need to finish the pi-web-agent API very soon, and the pi-web-agent for Android will help shape a well-defined and documented API. Then, we will extend the framework to encapsulate the API modules both on client and server side. We plan to create the client-side framework using Dart (**https://www.dartlang.org**).

We also plan to start a new spin-off project that will be a Web site for hosting extensions for pi-web-agent, which users will be able to install via the Web application.

Our Team's Goals

Following the Android model, we want to build a platform to act as a desktop environment for the Raspberry Pi, which will be able to expand via extensions or Activities. A different Web site will act as a market and host those extensions. Developers will be able to register and publish their own extensions.

Get Involved

The project is small—as it should be at this point. We want you, the users, to get involved before we start over-engineering things and making the pi-web-agent a big ugly piece of code that doesn't meet your needs.

There are a lot of ideas that need to be implemented. The pi-web-agent is already a product that is released in approximately a three-month cycle. You can become involved by sending us e-mail with recommendations, following us on GitHub, and forking and contributing to the repository. If you are new to any of these, don't worry; just send us your questions, and we'll get you started.

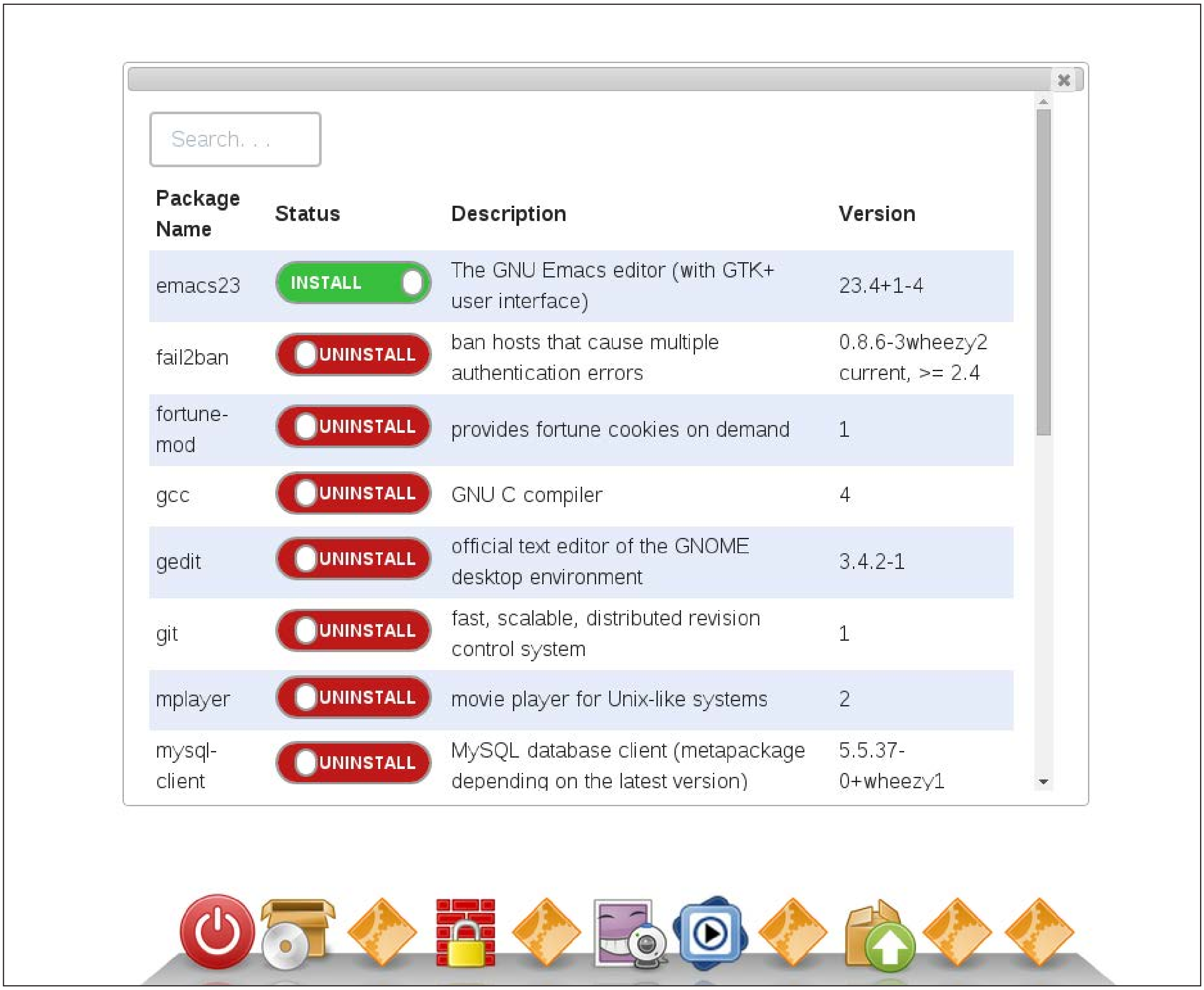
If visualizing a desktop environment in a Web browser is not so exciting for you, you can contribute to any of the spin-off projects that aim to boost the pi-web-agent development process. Here is a list of all the projects and their repositories:

- pi-web-agent: **http://www.github.com/vaslabs/pi-web-agent**.
- Benchmark for testing the framework: **https://github.com/azardilis/testing-fw**.
- pi-web-agent for Android: **https://github.com/vaslabs/pi-android-agent**.

- pi-web-agent gedit development plugin.

Conclusion

The pi-web-agent is a product that aims to replace the desktop environment with a Web-based alternative. HTML5 and CSS3 technology has made this possible. Along with the Linux extensibility, we chose to start with the Raspberry Pi, as it's the perfect platform for educational and experimental purposes. The Raspberry Pi also has the resource limitations we need, with the idea that if it runs fast on a



Package Name	Status	Description	Version
emacs23	INSTALL	The GNU Emacs editor (with GTK+ user interface)	23.4+1-4
fail2ban	UNINSTALL	ban hosts that cause multiple authentication errors	0.8.6-3wheezy2 current, >= 2.4
fortune-mod	UNINSTALL	provides fortune cookies on demand	1
gcc	UNINSTALL	GNU C compiler	4
gedit	UNINSTALL	official text editor of the GNOME desktop environment	3.4.2-1
git	UNINSTALL	fast, scalable, distributed revision control system	1
mplayer	UNINSTALL	movie player for Unix-like systems	2
mysql-client	UNINSTALL	MySQL database client (metapackage depending on the latest version)	5.5.37-0+wheezy1

Figure 5. Sneak peek of the pi-web-agent version 1.0, simple window with content inside and dock as navigation menu.

Raspberry Pi, it will be super-fast on mainstream machines.

We already are developing a new design that brings the feel of a mainstream desktop environment. Figure 5, which demonstrates the dock navigation menu and the windowing system, provides a sneak peek of the new design.

As we mentioned already, we also need a lot of help. If you are a Python/CSS/HTML/JavaScript expert with some free time and a passion for open source, don't hesitate to contact us and join our team.

Acknowledgements

We want to give credit and a kind thank you to all the people that helped shape the pi-web-agent:

- Kyriacos Georgiou
- Maria Charalambous
- Argyris Zardylis
- Iliada Eleftheriou
- Theodoros Pertsas

Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.

Vasilis Nicolaou, pi-web-agent's founder, is maybe the biggest Linux lover on the team, at his job and maybe in other groups and places. He even knows what Linus Torvalds does daily better than Linus himself. Vasilis loves open source, Python, Java (don't tell that to Linus), Raspberry Pi and new technology in general, which (as Linus says) is what keeps him interested in programming. He graduated from the University of Manchester with an MEng degree in Software Engineering.

Angelos Georgiadis is a Java programmer who became a Python expert for the sake of pi-web-agent. He probably is the best chance of keeping pi-web-agent alive if Vasilis is hit by a bus tomorrow. He is the number-one suspect when something breaks in pi-web-agent and is probably responsible, since he has blown the repository quite a few times (17-ish).

Georgios Chairepetis is a young programming enthusiast, currently studying for an MSc in Software Engineering at the University of Manchester. He got involved with the rest of the pi-web-agent team initially by taking part in a hackathon contest and was lucky enough to win an award in the first competition he ever attended. He enjoys staying inside on Saturdays doing some programming with friends, but he also likes to go outside and enjoy the sunshine, maybe with some beer, when he has the chance.

Andreas Galazis has been a junior Web developer for six months. The fact that his favourite Linux application is Mplayer is somewhat misleading, as he spends most of his time coding rather than watching movies or listening to music, but when he does, he wants to do it the proper way. When he heard about pi-web-agent, he decided to join forces to develop an extension to demonstrate the power of his favourite media player.

DOC SEARLS

Stuff That Matters

Are we going to get real about privacy for everybody—or just hunker in our own bunkers?

I'm writing this in a hotel room entered through two doors. The hall door is the normal kind: you stick a card in a slot, a light turns green, and the door unlocks. The inner one is three inches thick, has no lock and serves a single purpose: protection from an explosion. This grace is typical of many war-zone prophylaxes here in Tel Aviv, Israel's second-largest city. The attacks come in cycles, and the one going on now (in mid-July 2014) is at a peak. Sirens go off several times a day, warning of incoming rockets from Gaza. When that happens, people stop what they're doing and head for shelters. If they're driving, they get out of their cars and lie on the ground with their heads covered.

Since I got here, I have joined those throngs three times in small, claustrophobia-inducing shelters—once in my hotel, once in an apartment house where I was visiting friends and once in a bunker entered through a men's room at the beach. After gathering behind a heavy door, everyone in the shelter tensely but cheerfully waits to hear a "boom" or two, then pauses another few minutes to give shrapnel enough time to finish falling to the ground. Then they go outside and return to whatever they were doing before the interruption.

But not everybody bothers with the shelters. Some go outside and look at the sky. I was one of those when I shot the photo shown in Figure 1 from the front porch of my hotel, a few moments after hearing a pair of booms.

The photo tells a story in smoke of two incoming Hamas rockets from Gaza, intercepted by four Israeli missiles. The round puffs of smoke mark the exploded rockets. The parallel trails mark the paths of the interceptors. These are examples of the

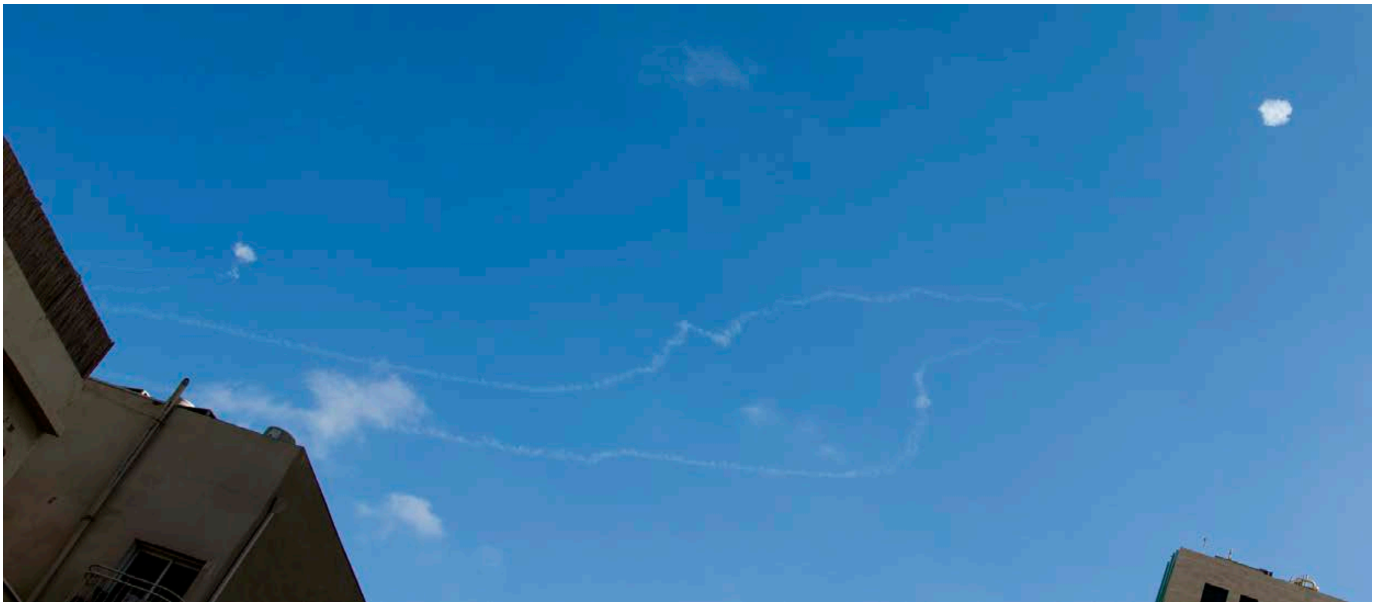


Figure 1. Two incoming Hamas rockets from Gaza, intercepted by four Israeli missiles.

"Iron Dome" (**http://en.wikipedia.org/wiki/Iron_Dome**) at work.

People here monitor rocket attacks using a free mobile app called Red Alert. The one on my phone tells me there were 27 rocket attacks fired from Gaza on Israel yesterday, including the long-range ones I saw intercepted over Tel Aviv. (Most are short-range ones targeted at cities closer to Gaza.)

Meanwhile, *Linux Journal* and its readers also are under an attack (**http://daserste.ndr.de/panorama/xkeyscorerules100.txt**), of sorts, by the NSA. Here are the crosshairs at which we sit, in an NSA surveillance system called XKEYSCORE:

```
// START_DEFINITION
/*
These variables define terms and websites relating to the
TAILs (The Amnesic Incognito Live System) software program,
a comsec mechanism advocated by extremists on extremist forums.
*/

$TAILS_terms=word('tails' or 'Amnesiac Incognito Live System')
 ➥and word('linux'or ' USB ' or ' CD ' or 'secure
 ➥desktop' or ' IRC ' or 'truecrypt' or ' tor');
$TAILS_websites=('tails.boum.org/') or
 ➥('linuxjournal.com/content/linux*');
// END_DEFINITION

// START_DEFINITION
/*
This fingerprint identifies users searching for the TAILs
(The Amnesic Incognito Live System) software program, viewing
documents relating to TAILs, or viewing websites that
detail TAILs.
*/
fingerprint('ct_mo/TAILS')=
fingerprint('documents/comsec/tails_doc') or
 ➥web_search($TAILS_terms) or
 ➥url($TAILS_websites) or html_title($TAILS_websites);
// END_DEFINITION
```

Those details come via a story on the German site Tagesshau (**http://www.tagesschau.de/inland/nsa-xkeyscore-100.html**). On our Web site, Kyle Rankin explains what's going on (**http://www.linuxjournal.com/content/nsa-linux-journal-extremist-forum-and-its-readers-get-flagged-extra-surveillance**):

> XKEYSCORE uses specific selectors to flag traffic, and the article reveals that Web searches for Tor and Tails—software I've covered here in *Linux Journal* that helps to protect a user's anonymity and privacy on the Internet—are among the selectors that will flag you as "extremist" and targeted for further surveillance. If you just consider how many *Linux Journal* readers have read our Tor and Tails coverage in the magazine, that alone would flag quite a few innocent people as extremist.

BoingBoing also was targeted. Writes Cory Doctorow (**http://boingboing.net/2014/07/03/if-you-read-boing-boing-the-n.html**), "Tor and Tails have been part of the mainstream discussion of online security, surveillance and privacy for years. It's nothing short of bizarre to place people under suspicion for searching for these terms."

Both kinds of attacks bring defensive responses. In the physical space above Israel, Iron Dome is fully developed and amazingly effective. In the cyber space surrounding us all on the Net, we have little to protect us from unwelcome surveillance. As the XKEYSCORE story shows, just looking for effective privacy help (such as Tor and Tails provide) places one under suspicion.

My current road trip began in London, where there are surveillance cameras in nearly all public spaces. These were used to identify the perpetrators of the bombings on July 21, 2005 (**http://en.wikipedia.org/wiki/21_July_2005_London_bombings**). Similar cameras also identified suspects in the Boston Marathon bombings of April 15, 2013 (**http://en.wikipedia.org/wiki/Boston_bombing**). It is essential to note, however, that these cameras are not in people's homes, cars and other private spaces (at least not yet).

Our problem with the Net is that it is an entirely public space. In that space, Tor (**https://www.torproject.org**) and Tails (**https://tails.boum.org**) are invisibility cloaks, rather than the cyber equivalents of clothing, doors, windows and other well-understood and widely used ways of creating and maintaining private spaces.

We do have some degree of privacy on our computers and hard drives when they are disconnected from the Net and through public key cryptography (**http://en.wikipedia.org/wiki/Public_key_cryptography**) when we communicate with each other via the Net. But both are primitive stuff—the cyber equivalents of cave dwellings and sneaking about at night wearing bear skins.

It should help to recognize that we've been developing privacy technologies in the physical world for dozens of thousands of years, while we've had today's version of cyber space only since 1995, when ISPs, graphical browsers and the commercial Web first came together.

But living in early days is no excuse for not thinking outside the boxes we've already built for ourselves. Maybe now that wizards are in the crosshairs—and not just the muggles—we'll do that.

Got some examples? Let's have them.■

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.

Send comments or feedback via http://www.linuxjournal.com/contact or to ljeditor@linuxjournal.com.

Advertiser Index

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

ADVERTISER	URL	PAGE #
All Things Open	http://allthingsopen.org	2
AnDevCon	http:/www.AnDevCon.com	45
Drupalize.me	http://www.drupalize.me	73
EmperorLinux	http://www.emperorlinux.com	37
High Performance Computing Shows 2014	http://www.flaggmgmt.com/hpc	7
Percona	http://www.percona.com/live	93
Silicon Mechanics	http://www.siliconmechanics.com	3
SPTechCon	http://www.sptechcon.com	27

ATTENTION ADVERTISERS

The *Linux Journal* brand's following has grown to a monthly readership nearly one million strong. Encompassing the magazine, Web site, newsletters and much more, *Linux Journal* offers the ideal content environment to help you reach your marketing objectives. For more information, please visit http://www.linuxjournal.com/advertising.