LINUX™

JOURNAL

Since 1994: The Original Magazine of the Linux Community

STUNNEL SECURITY

for Databases

Protect Your Desktop Environment with Qubes

Intro to Pandas
The Python Data Analysis Library
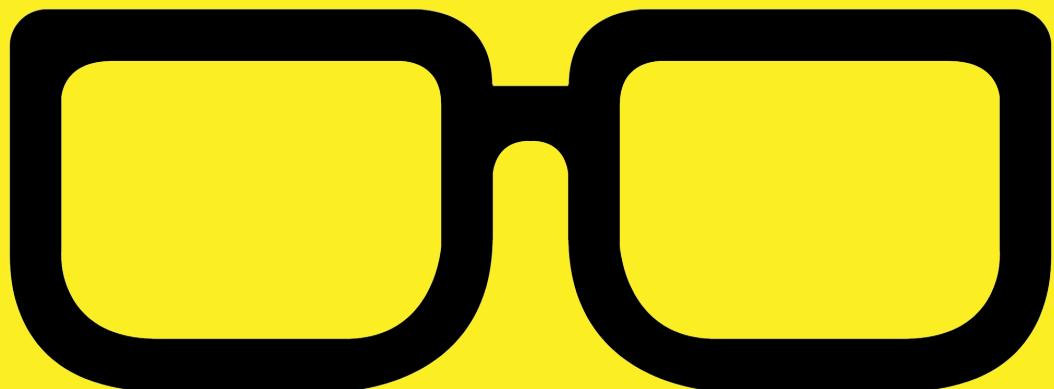
A Look at printf
A Super-Useful Scripting Command

What's the Kernel Space of Democracy?

# BE SMART ABOUT CREATING A SMART HOME
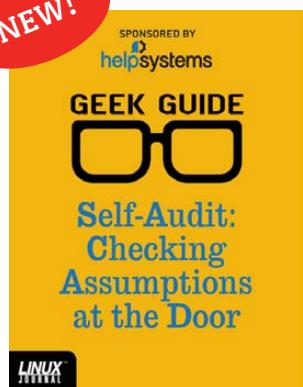
NEW!

Self-Audit: Checking Assumptions at the Door

Author:
Greg Bledsoe
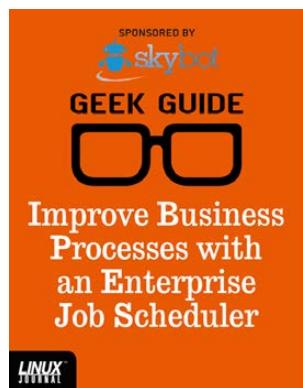
Sponsor:
HelpSystems

Agile Product Development

Author:
Ted Schmidt
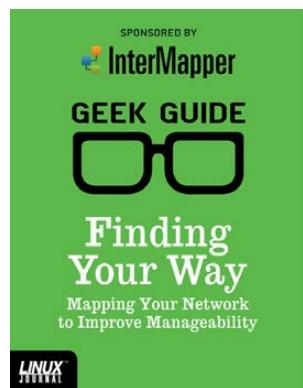
Sponsor: IBM

Improve Business Processes with an Enterprise Job Scheduler

Author:
Mike Diehl

Sponsor:
Skybot

Finding Your Way: Mapping Your Network to Improve Manageability

Author:
Bill Childers

Sponsor:
InterMapper

DIY Commerce Site

Author:
Reuven M. Lerner

Sponsor: GeoTrust

Combating Infrastructure Sprawl

Author:
Bill Childers
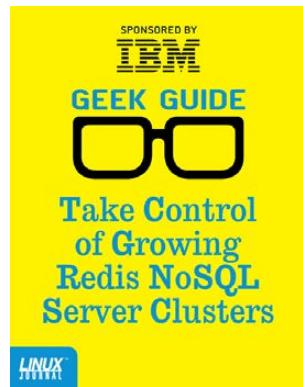
Sponsor:
Puppet Labs

Get in the Fast Lane with NVMe

Author:
Mike Diehl

Sponsor:
Silicon Mechanics & Intel

Take Control of Growing Redis NoSQL Server Clusters

Author:
Reuven M. Lerner

Sponsor: IBM

CONTENTS APRIL 2016
ISSUE 264

FEATURES

CONTENTS

COLUMNS

24

IN EVERY ISSUE

60

NEW ORLEANS

DRUPALCON 2016

ERNEST N. MORIAL CONVENTION CENTER
MAY 9 - 13, 2016

Join us in the Big Easy.

With Drupal 8 newly released and thousands of community members in attendance, DrupalCon New Orleans promises to be an event to remember.

See you in New Orleans this May.
Laissez les Bon Temps Rouler!

neworleans2016.drupal.org

Drupal™
ASSOCIATION

Linux Does Stuff

SHAWN POWERS

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.

We're huge fans of open source here at *Linux Journal*, which I'm sure comes as no surprise to anyone. The best part about Linux itself, however, is that it's the concept of open source realized. It has permeated every aspect of IT, and it has proven that being open doesn't equate to being insecure. In fact, it's quite the opposite. When you have nothing to hide, there aren't any dirty secrets waiting to be leaked. In the spirit of "doing things", this month we've got a bunch of really cool topics that show open source in action.

We start with Reuven M. Lerner. Last month he talked about navigating data, and this month he talks about Pandas. Specifically, Reuven talks about parsing and analyzing CSV (comma-separated values) files with Python. If you're a data nerd and want to get the most from your CSV data files, you won't want to miss Reuven's column this issue. Dave Taylor follows with a look at some powerful scripting commands borrowed from the C library. If you want to tighten your code, the printf command is incredibly powerful and thankfully available for scripting.

Kyle Rankin keeps his security head firmly in place and starts a series on Qubes this month. Qubes is a distribution focused on security. With all the publicity encryption and privacy is getting thanks to the Apple/FBI case, it's important to understand how

VIDEO: Shawn Powers runs through the latest issue.

security on your devices functions. Kyle starts his series by describing how Qubes compartmentalizes applications, isolating them from each other and the OS itself. Whether or not you want to beef up your desktop security, his article is a fascinating look at an awesome technology.

I go in a very different direction this month, and rather than talk about security, I focus on what sometimes can be the opposite of security—convenience. I've always wanted a smart house, and thanks to SmartThings and the Amazon Echo, I finally have one—or at least the start of one. If you've ever wanted to talk to your house like it was the computer on the *Starship Enterprise*, you'll want to check out my column.

SSH is arguably my favorite command-line tool in Linux. It's secure, and it's so versatile. Ramon Crichlow explains how to stream video securely through an SSH tunnel this month. Not only will you learn how to accomplish a cool video streaming task, but along the way, you'll learn a lot about how SSH works and what tunneling really means. You'll also learn how to tweak it so it's not more frustrating than useful!

Charles Fisher follows Ramon with a very in-depth look at using stunnel as a tool for authentication, isolation and privacy of data stored in an Oracle database. If you've ever managed an Oracle database and had concerns about its security implementations (even considering recent improvements), using the open-source stunnel tool can add a solid layer of security that is regularly updated and offers the peace of mind that comes with FOSS.

Whether you want to improve the security of your desktop environment, use Linux as a tool to accomplish a necessary function, or just turn on your bedroom lights by talking to a Pringles-can-shaped robot, this month is an issue worth reading. As always, it's also full of product announcements, time-saving tips and other Linux-related goodies. Whether this is your first issue of *Linux Journal* or you've been one of us for years, we hope you enjoy this issue as much as we enjoyed putting it together.■

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

RETURN TO CONTENTS

Dave Taylor's Article on getopt

Regarding Dave Taylor's "Working with Command Arguments" in the February 2016 issue: it's a worthy article but let's expand on it a bit. Long arguments like `--help` certainly deserve a mention.

But my main gripe with using getopt in bash is the lack of a wrapper function. Python, C, C++, Ruby, etc., all have wrappers that simplify using getopt enormously.

I help maintain hundreds of scripts at work (and home!), and I find the biggest source of errors are those that creep in during maintenance.

The problem is that there are multiple places that need to be kept in sync—the call to getopt itself, the case statement where the options are processed and the help output that documents it all.

So—time for shameless self publicity—I wrote a wrapper for bash that fixes this shortcoming. It's at http://bhepple.com/doku/doku.php?id=argp.sh.

To use it, you put all the information about the various options in a simple variable (or here-document) and run a bit of mumbo-jumbo to process the arguments. For that low, low price, you get the options parsed, a call made to getopt and variables set for you as well as a help screen—all automatically from the single source. Here's an example:

```
ARGS="
##############################################################
#OPTIONS:
#name=default sname arg        type range   description
##############################################################
CD=''         c     ''         b    ''      foobar
SLOT=''       s     'n'        s    ''      option that takes a value
TOKEN=''      t     'number'   s    ''      option that takes a value
LONG=''       ''    'long'     s    ''      a long option without a
                                            short one
SILENT=''     ''    ''         b    ''
##############################################################
"
    exec 4>&1
    eval $(echo "$ARGS" | argp.sh "$@" 3>&1 1>&4 || echo exit $? )
    exec 4>&-
```

That last bit looks scary, but it does the job—if the user invoked the script with `-s 5`, the bash variable `SLOT` has that value, etc., etc.

Hope it helps—it certainly has made my life much easier.

—Bob Hepple

CSV Files and the Comma

This is with reference to Dave Taylor's article in the December 2015 issue about dealing with CSV files. Yes, for people skilled with the power of the shell, doing taxes and such accounting things with scripting is very easy and full of enjoyment. I do not know who invented the comma as a delimiter or the format called "csv". Instead, that person should have used tab as a delimiter, and the format should have been called "tsv"! I use tab as a delimiter (when exporting from spreadsheets using OpenOffice) and face no problems whatsoever. Besides ease of processing, the file becomes so much more readable with the suitable tab-stop setting.

—Mayuresh

Dave Taylor replies: Mayuresh, I love your suggestion, and you're right, the use of any punctuation symbol that can occur within the data fields themselves is really pretty stupid as a format. When I convert delimiters back and forth, I use sequences like ^^ that are incredibly unlikely to show up in any prose or data set.

More bizarre—CSV is a standards-body-approved format:
http://www.digitalpreservation.gov/formats/fdd/fdd000323.shtml.

My only comment: tabs can make things more readable, but if you've wrestled with data where the fields vary from less than to greater than a tab's width (typically eight characters), you know how annoying that can be to align perfectly.

Thanks for writing in!

Does Every Year Have a Friday the 13th?

The "Command-Line Tutorial" in the February 2016 issue was a fun article by Sol Lederman. It got me playing around with the commands, and I started manually checking with `cal` and `cksum` for unique leap years. When I found the eighth unique one, I started to worry if the world was ending.

After digging around a bit, I discovered that my mis-matched pair was 2016 and 2044. For some reason, in Debian Jessie (32 and 64)—okay in Fedora 23—the year 2016 is generated in a strange fashion.

The numerical dates are displayed with digits separated by one or more spaces (0x20) as appropriate, except for the prior and current date when the cal is generated; in this case the 2nd, 3rd last night, and the 3rd, 4th this morning.

The date of generation and prior day's date are separated by "space", "underbar", "backspace" (0x20 0x5f 0x08) sequences instead of just "space". Now the `cksum` and `wc  -c` just went out the window

and don't match anything, even though the appearance of the two calendars is identical except for the year.

So I would surmise this applies to any Debian derivatives and only for the current year, and possibly for only single-digit dates. I'll have to wait to see what happens on a two-digit date! Perhaps it's a remnant of syntax to highlight the current date.

—Wally Olson

Sol Lederman replies: I'm glad you had fun with the calendar puzzle. Thanks for pointing out that different flavors of Linux render calendars differently. I ran into differences as well, and pointed it out in the article, hoping to give readers a heads up in case they got unexpected results. Perhaps other readers will find even more differences. And, maybe the lesson here is that a command as simple as `cal` *has different output on different systems. Hopefully, others who run into different flavors of* `cal` *will be inspired to dig in and figure out what's wrong as you did. Happy computing!*

Google Blocks the Inclusion of APNG in Blink/Chromium

I believe it's newsworthy that Google is effectively blocking the inclusion of an already-ready patch to include APNG support in Chromium/Blink.

This reaction is so blatantly against the community and so clearly in protection of Google's own WebP that nobody uses that I can't help but clench my fists.

I would very much appreciate if you could spread the news to your readers and colleagues so that the Internet may finally be free of the legacy of GIF.

Here's the thread: https://groups.google.com/a/chromium.org/forum/#!topic/blink-dev/KcMjmFOgG2w.

—OlegM

The Powers That Be

Regarding Shawn Powers' "The Powers That Be" in the February 2016 issue: I really liked the article; it reminded me of a similar problem we had a few years ago. All of a sudden, every evening at 10pm our power would drop for a moment, and every computer in the house would drop—no explanation. We got the power company out, and they found nothing wrong with the line (but they were testing only during the day). Anyhow, three visits later, I mentioned the major hospital three blocks away, and I noticed that there was power-line work happening on the main road. (We lived on a back street three streets from the main road.) It turns out that while the power company had been doing power work on the main line, they switched the power for the hospital's incinerators to the sub-lines on the back streets and had forgotten to switch them back. The problem was that the hospital incinerators were auto-set to start every night at 10, and this was putting too much drain on the line and causing minor drop outs. The moral: it pays to look around.

—Trevor Furnell

Shawn Powers replies: I can only imagine your irritation! Great job figuring out what was causing the problem. Finding a resolution like that is almost worth the frustration it caused in the first place. Almost.

The Powers That Be, II

I had power problems like Shawn's in my house for some time. Although the house was new in 2013, the power drop from the utility pole into my electric meter was very old, as the previous house was demolished to build a new one. I plugged an analog AC voltmeter in to an outlet and watched the voltage fluctuate from 90 volts to 120 volts. My lights would flicker also. After many months of this, I called the electric company. They checked the connection at the utility pole end of my drop and found a badly corroded splice. The lineman cut off the corroded ends and made a new connection

at the pole, and my problems disappeared.
This might be the cause of your problem. The
fluctuations were worse on windy days since
the drop wires were swaying in the wind
making the fluctuations even more apparent.

I hope this helps.
—Eric

Shawn Powers replies: *Thanks for the suggestion! During the next year, our city is moving overhead lines to under ground. Hopefully during that process, the new lines will make for more stable electricity. Of course, now it's a moot point for me, but still, stability is nice!*

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

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

RETURN TO CONTENTS

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 if you want to stop maintaining a piece of software but no one will take it over, all you have to do is simply announce that you're stepping down, and everyone will jump for it.

That was **Neil Brown**'s experience with **software RAID**. After 15 years as maintainer, he wanted out, but he couldn't get anyone to commit to take it over. So, he announced his departure and offered up a description of what he saw for software RAID going forward: a small team of maintainers who would gather and review patches, resolve bugs and feed patches upstream.

Lo and behold, lots of people expressed interest in taking over maintainership or at least in participating in a team. After sifting through many volunteers, he settled on **Jes Sorensen** for **mdadm** and **Shaohua Li** for the **kernel/md** side of things. Neil documented some of the basics for Jes and Shaohua to consider, saying:

The first question is where do you send your patches to get the appropriate review and upstream acceptance. Alasdair or Mike (DM), Jens (Block), Andrew Morton (anything) and Linus (everything) are all defensible choices for upstreaming (I've submitted through Andrew in the past, but through Linus exclusively once I figured out git). That is really something you and

they would need to negotiate though. [...] I plan to submit a pull request to Linus for the 4.5 merge window and then stop queuing patches.

And Jes also announced a new git tree for mdadm, at http://git.kernel.org/pub/scm/utils/mdadm/mdadm.git.

Kernel documentation has traditionally been written in **DocBook**, an XML-based system that's been falling behind the increasingly popular forms of readable markup, like **Markdown**, **AsciiDoc** and others. Recently, **Jonathan Corbet** and **Jani Nikula** did some overlapping work to convert the kernel to use AsciiDoc for all documentation.

The goal was not just to adopt the new hotness, but also to reduce some of the many tool dependencies that were needed for DocBook processing and speed up overall doc production. Along the way, however, any new system would have to be at least as good as DocBook and support large files, cross references, a big pile of output formats and so on.

There turned out to be significant problems doing the whole migration. One of the most viable options, at least temporarily, turned out to be migrating the source files to AsciiDoc, but having the makefiles process that into DocBook and from there into whatever output was needed. This would not reduce the number of tool dependencies, but it would at least produce reliable output, while accepting the new more preferable input.

Ultimately, DocBook would be eliminated, but for now it seems there will be this intermediate step. It's also possible that AsciiDoc would need to be modified upstream, before it would be able to handle the kernel docs without DocBook fully.—Zack Brown

THEY SAID IT

Giving is a necessity sometimes... more urgent, indeed, than having.
—Margaret Lee Runbeck

Life is full of obstacle illusions.
—Grant Frazier

The future, according to some scientists, will be exactly like the past, only far more expensive.
—John Sladek

I think the world is run by "C" students.
—Al McGuire

Real freedom lies in wildness, not in civilization.
—Charles Lindbergh

Back to Backups

In my Open-Source Classroom column last month ["Back It Up, Buster", March 2016], I talked about backups and got some really fascinating feedback. In fact, at the time of this writing, it's still pretty early in the month, so I expect to get even more ideas and suggestions for backup options. Here's a few of the ideas worth checking into:

■ **Carlos Baptista** wrote in as someone who also has struggled with lost data. His current solution is to use rsnapshot (http://rsnapshot.org) on a pair of 4TB drives. Every week he swaps the drives, taking one to his parent's house. This solution gives him low-tech off-site storage, a maximum of one week of lost data in the event of a total failure, plus an excuse to visit his parents on a regular basis. Awesome job!

■ **Harald Nipen** takes the interesting step of making sure the duplication process for his backups is *not* automated. That may seem like a silly thing to do, but as someone who accidentally reversed the source and destination on his rsync backup script before, I can assure you there is some peace of mind that comes from manually seeing your backup take place. Harald does, of course, automate his regular backups, but the duplication process for off-site storage is a manual process using the unison program.

■ **Nicola Larosa** pointed me to an interesting project that uses "content-defined chunking" to back up data efficiently. It's the fastest backup system he's ever used and worth checking out if you have large amounts of data to secure: http://restic.github.io/blog/2015-09-12/restic-foundation1-cdc.

■ Finally (for now), **Johann Schoonees** wrote in about rdiff-backup. It's a program I'd heard of but never really looked into using. That's unfortunate though, because it really is a neat concept. If you've

ever used BackupPC to keep rsync snapshots hard-linked to save space, it's a little like that. The program is an all-in-one solution, however, that keeps a current snapshot of a filesystem while also keeping diff files of previous changes. That allows older versions of files to be recovered without the complexity of setting up the entire BackupPC system.

The most encouraging part about getting followup e-mail messages from readers about their backup solutions is to hear that lots of folks actually have backup solutions! Regardless of the complexity of your backup process or the level of automation you deem appropriate for your data, apart from creating the memories in the first place, few things are as important as backing them up!—Shawn Powers
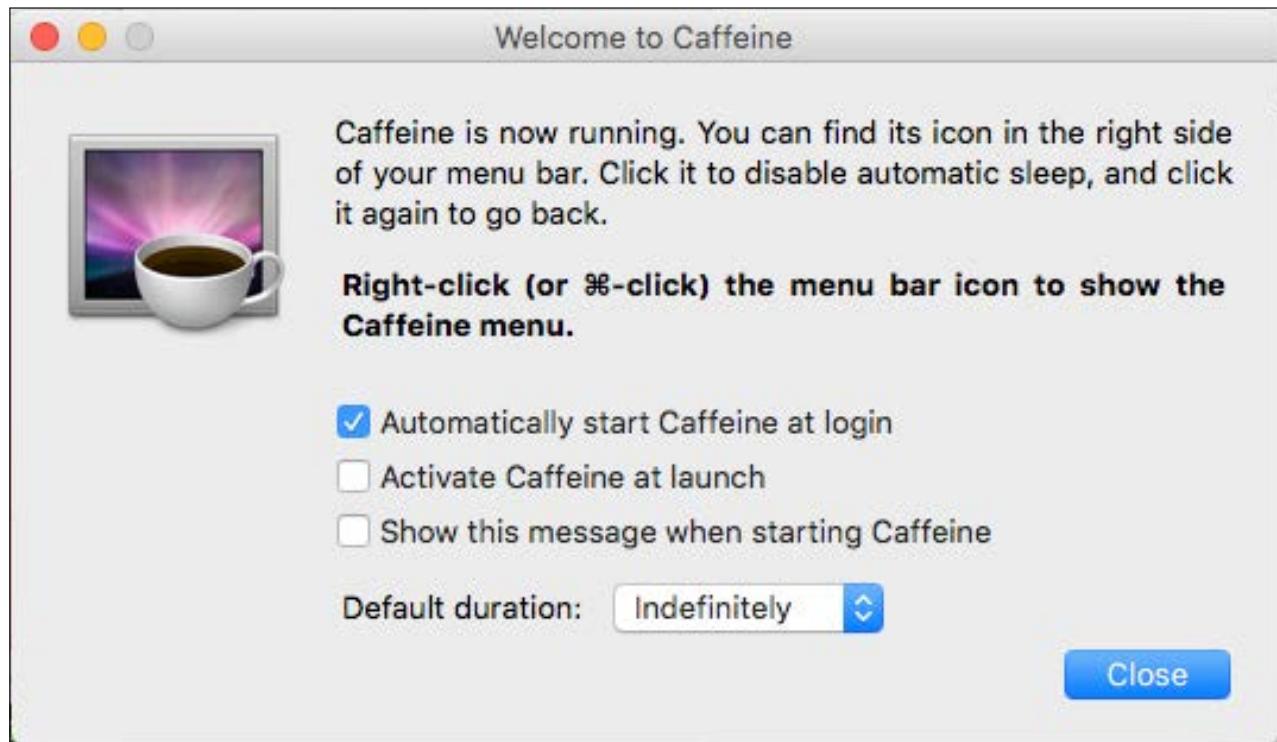
Non-Linux FOSS: Caffeine!
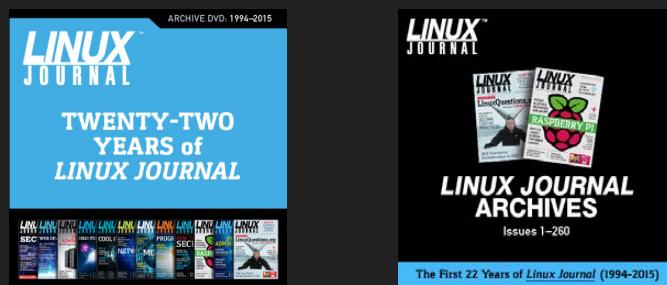
Okay, this program is free (beer), but not Free (speech). I wouldn't normally include a freeware application in the "Non-Linux FOSS" section, because quite frankly, it isn't FOSS. But, I decided to break the rules a bit here because I realized how often I use a freeware program when I'm on OS X that I couldn't imagine doing without.

   If you use OS X for presentations or demonstrations, you've probably had your screen shut off while you explained a slide or dialog box. Then the screen probably locked, and you had to hurry over to the keyboard so you could unlock, and so on and so on. Caffeine is a simple app that does nothing more than keep your Macintosh (or Hackintosh) computer awake. It runs

as a cute little icon in your menu bar, and it disables screen savers and sleep mode, even if you have aggressive power-saving settings enabled.

The danger is that you could leave Caffeine running accidentally and completely drain your battery. I've had that happen only one time, however, and I learned quickly to take my computer off Caffeine when I was done presenting. Even with that risk, I find it's worth it to no longer need to wiggle the mouse pointer every minute to make sure my laptop doesn't fall asleep!

Caffeine is available for free in the App Store, or you can get it from its Web site: http://lightheadsw.com/caffeine.—**Shawn Powers**
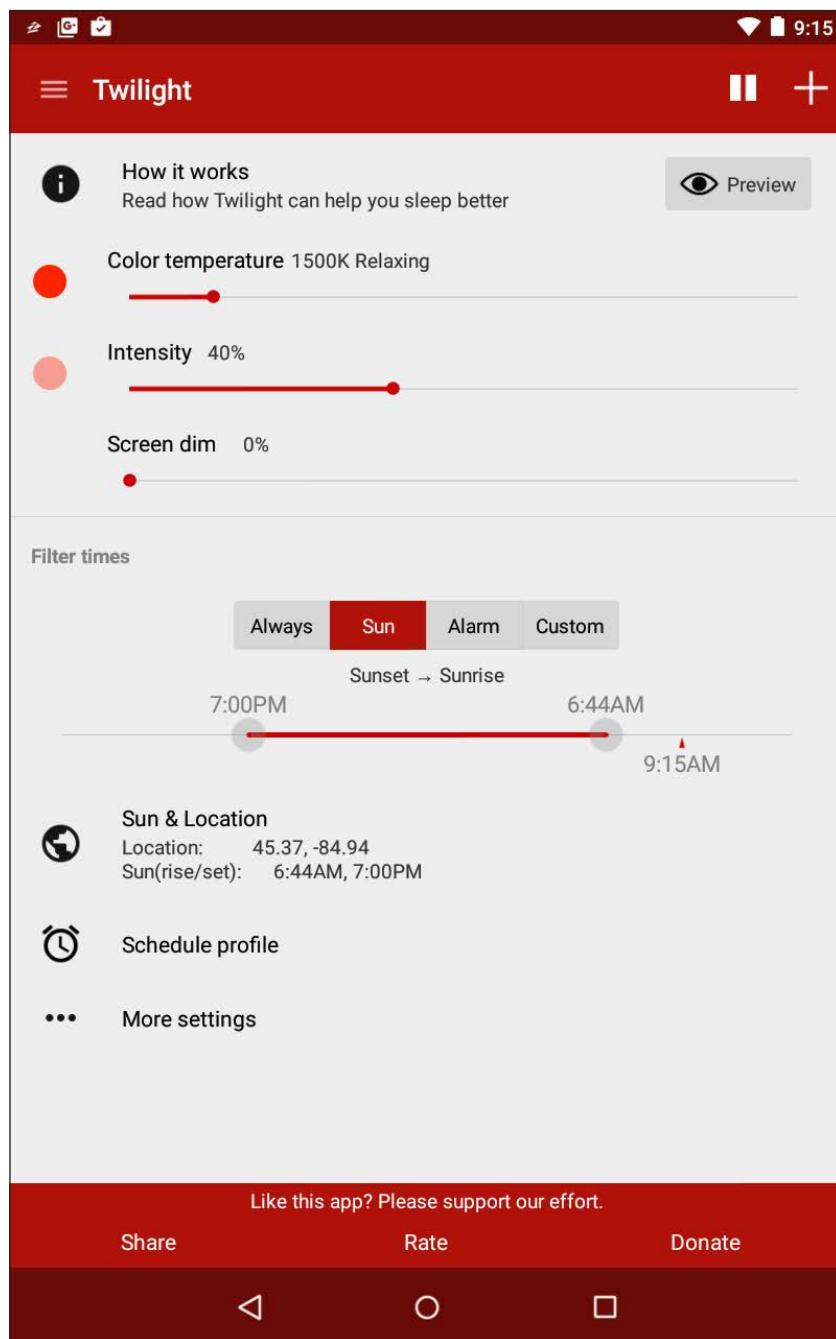
Android Candy: Seeing Red and Getting Sleep

I'm always leery when I hear, "Recent studies show…". But the idea that looking at electronic device screens before bed can cause sleep issues seems to be fairly accepted. The fascinating part for me is that it isn't really the screen itself, but the blue part of the color spectrum that contributes to the sleeplessness. In a purely anecdotal experiment, I find that it's much more difficult for me to fall asleep in the kitchen (cool-colored lighting, closer to blue in the spectrum) than it is to fall asleep on the living-room couch (warmer, less blue lights). Granted, part of that might be general comfort on the couch and more sharp objects in the kitchen, but in

general, warmer lighting tends to be more relaxing.

Based on the idea that less blue and more red will make for better sleeping, the "Twilight" app for Android shifts the color of your screen as bedtime approaches. It's free for basic functionality, but for a small price, you can get a timed, gradual transition on your Android devices.

I have no idea whether it helps me sleep better, but if nothing else, it reminds me that it's getting late as my ebook becomes redder and redder as I read. If you struggle with sleeplessness especially when you first go to bed, give Twilight a try. It's free in the Google Play Store, and if it works, the gradual shift option with the paid version is well worth the cost. And as a bonus, the red screen won't hinder your night vision during those late evenings of summer stargazing!

—Shawn Powers

Night Sky Tools on Android

In previous articles, I've looked at several different astronomy programs that you can run on your Linux machines. Those are great when you are doing work indoors, but most laptops and Netbooks are still a bit of a pain to carry around with you if you are going out into the field. In those cases, having something more portable is definitely nice. And, since I'm beginning to look at Android apps in this column, this is a perfect opportunity. Loads of astronomy applications are available within the Android environment that are well worth a look. In this article in particular, I'm exploring Night Sky Tools. The application is available in the Google Play store, and it should run on most versions of Android.

Once you have it installed, open it to see a very complete menu of all of the functionality available within Night Sky Tools. Many of the functions are updated over the Internet automatically, so you are sure to have the latest information for whatever objects you are trying to observe in the night sky.

The first category is general astronomical information. You can see lists of upcoming astronomical events, as well as
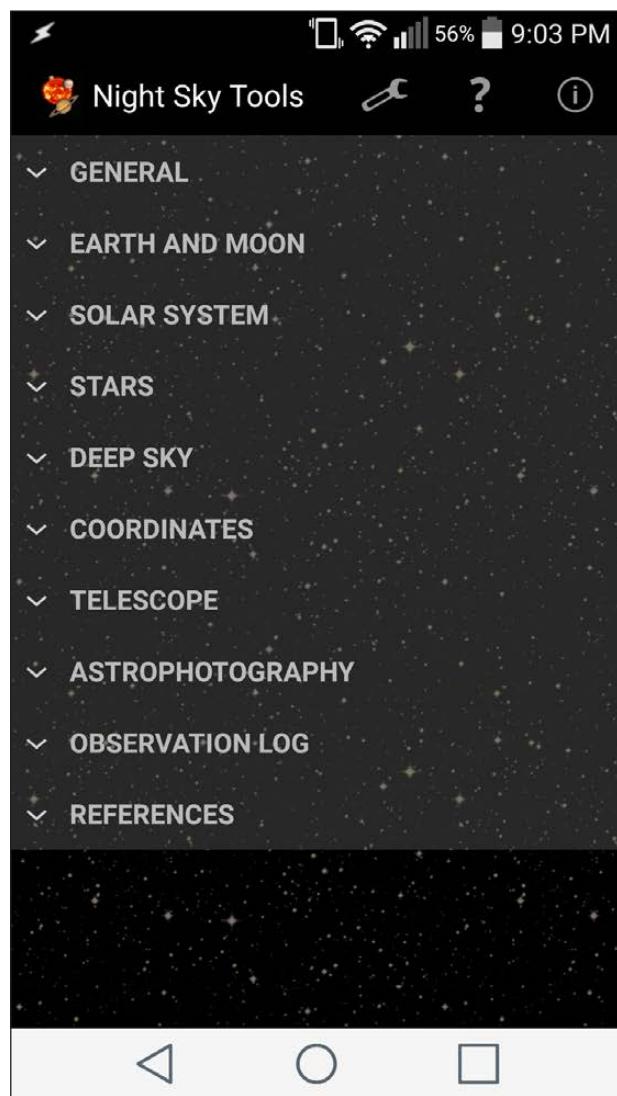
Figure 1. The opening screen displays a menu of the various categories of functions available.

what is up right now and what will be coming up tonight. There also are entries for a compass and a page of the various astronomical times that you may need. The astronomical time page gives the moonrise/moonset and sunrise/sunset times. There even is a page that lets you calculate the visual limiting magnitude based on the actual environmental conditions like temperature and humidity. The last page in this section is the sky map. The information provided in the sky map is rather complete. You can see the stars and constellations, along with the formal constellation boundaries. It even displays artwork for each of the constellations, showing you what they are supposed to look like.

The next section has information on the Earth and Moon. (Note: when moving between sections, be aware that

Figure 2. The sky map provides a display of what the sky looks like at the current time.

the other sections do not automatically close.) You also can pull up a daylight map, showing what parts of the Earth are in daylight and which are in night.

There are pages that show when the solar and lunar eclipse happen, along with dates for the equinoxes and solstices. The meteor page gives a listing of all of the known meteor showers, with the start, peak and end dates. It also, helpfully, gives the percentage of the moon phase so that you can tell right away whether the night will be dark enough to have a good observing period. The moon map, by

Figure 3. The daylight map shows you where on Earth it is day and night.
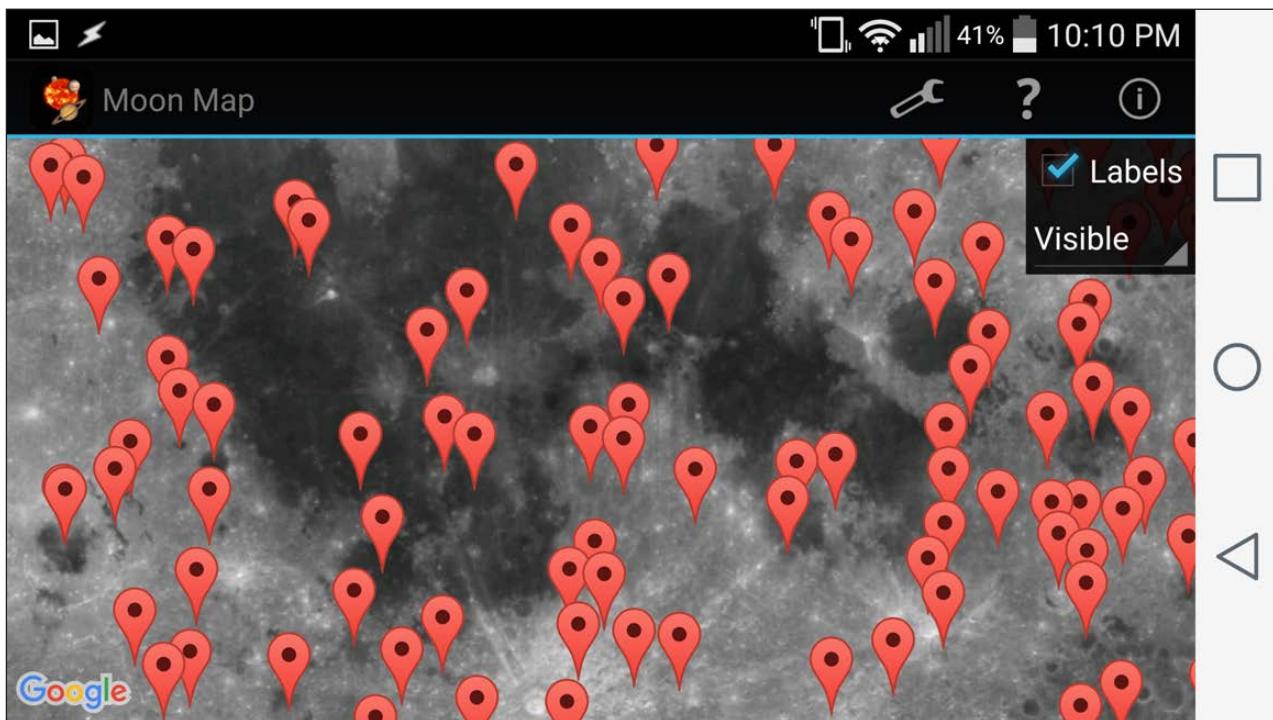
Figure 4. The moon map shows points of interest with pins on the map.

default, shows a full listing of sites of interest. Tapping on one of these points brings up a text label with the name of the site.

The solar system category extends available information farther out into space. The Conjunction/Opposition page provides a list of all times this year when planets either form a conjunction or an opposition. There are two pages, one for comets and one for near-Earth asteroids, where you can search for detailed information on specific comets or asteroids. You also can click the update button to pull a fresh listing from the Internet of what objects are known.

There are four large moons orbiting Jupiter that are visible in a large
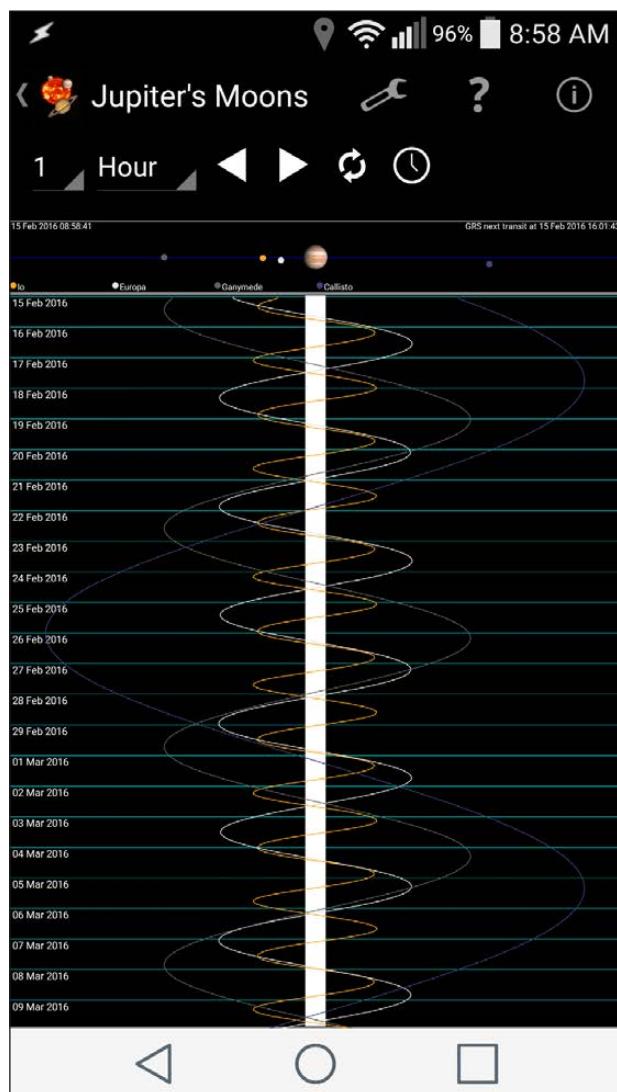
Figure 5. You can get a map of the locations of the four main Jovian moons.
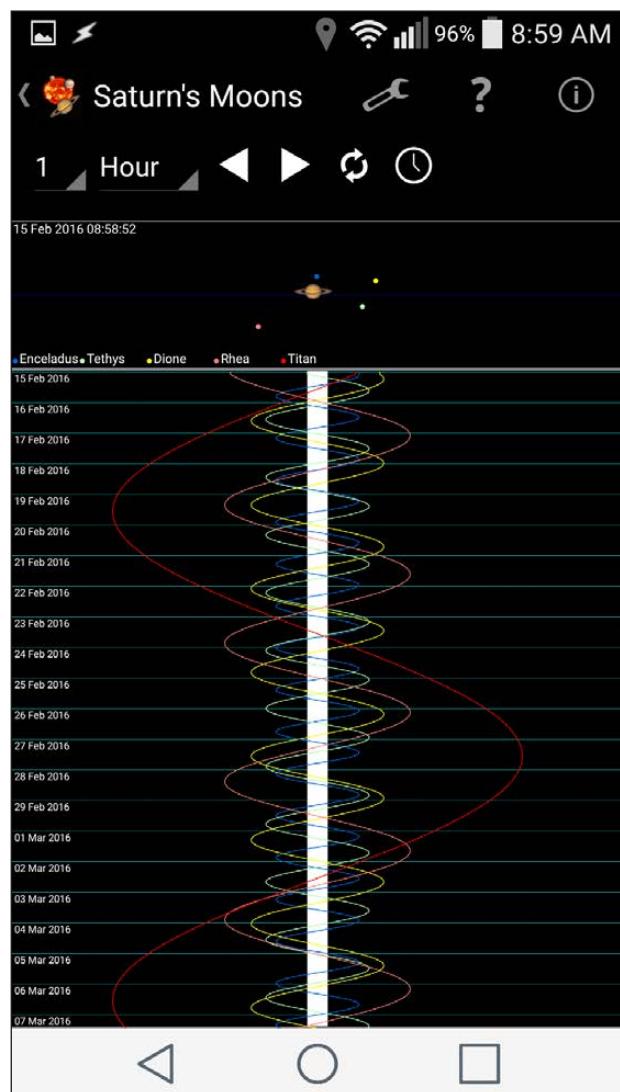
Figure 6. You can get a map of the largest of Saturn's moons as well.

pair of binoculars. Clicking on the Jupiter's Moons page brings up a map of their relative locations around Jupiter. A related page gives you the positions of the largest of Saturn's moons too. Viewing them will require at least a small telescope though.

The Planetary Orbits page takes things even farther out to see the relative positions of all of the planets within the solar system. As with the Moon map described previously, this section includes a Mars map, also with pins at locations of interest.

The stars category takes you even farther out into space. The first selection provides a list of the 300 brightest stars in the sky. The list includes the name, magnitude and location for each of these stars. The entire sky is divided up into
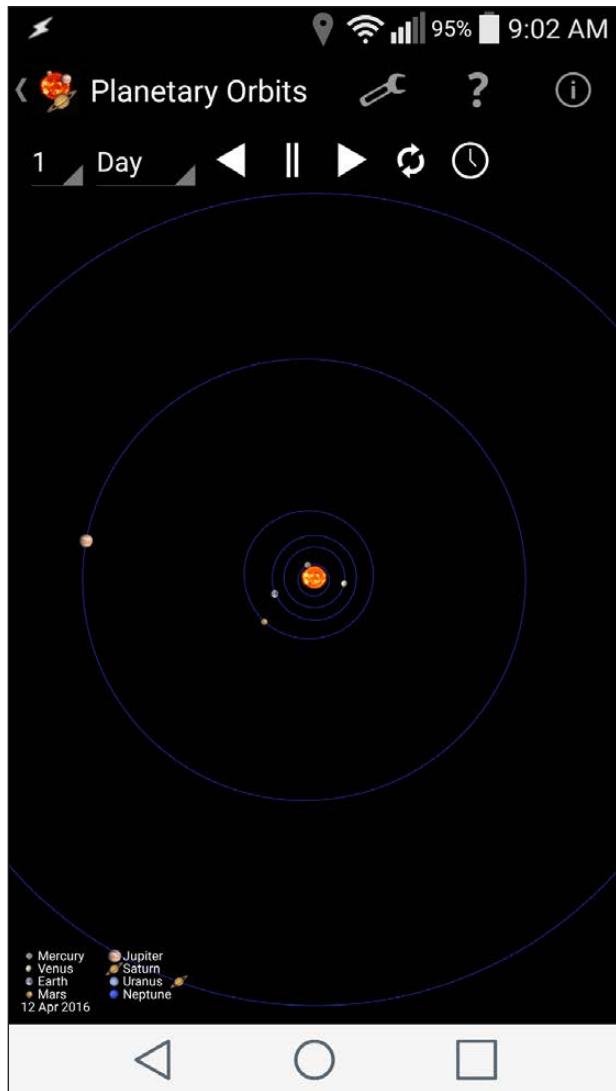
Figure 7. The relative locations of all of the planets are available on the Planetary Orbits page.
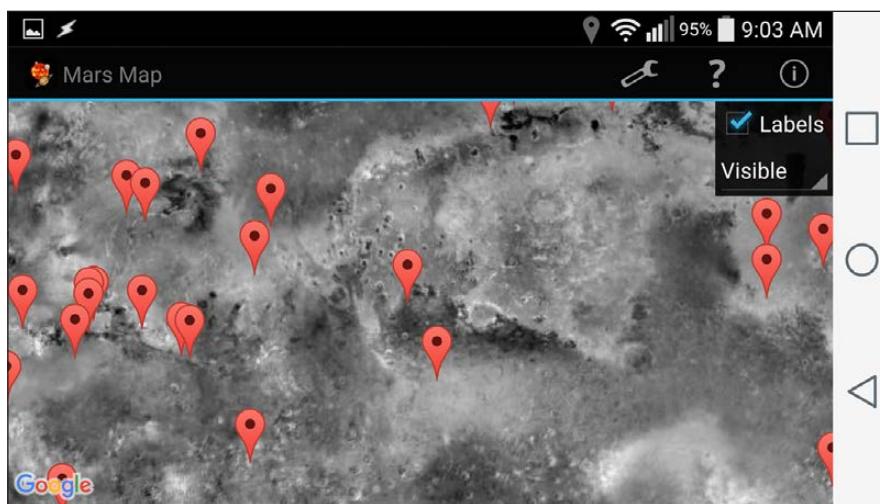
Figure 8. There is a Mars map with locations of interest.

constellations. The constellations page pulls up detailed information for the selected constellation.

There also are pages with theoretical information about astronomy. For example, you can pull up a Hertzsprung-Russell Diagram showing how stars are categorized. The stellar classification page describes the ten classes of stars with their temperature, mass, radius and luminosity characteristics. The last two pages in this section let you search for information on variable stars and visual binary stars.

The deep sky section contains pages for several of the deep sky catalogs. The Caldwell and Messier catalogs are displayed as a list of all of the objects within the catalog. You can click on an object of
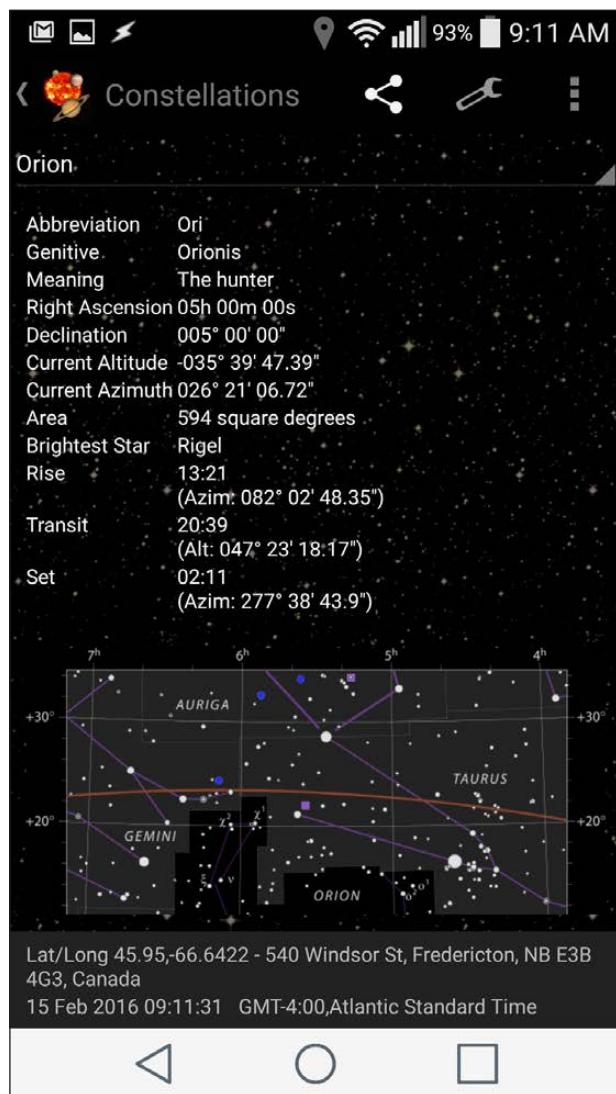
Figure 9. You can get detailed information on constellations, including a map.
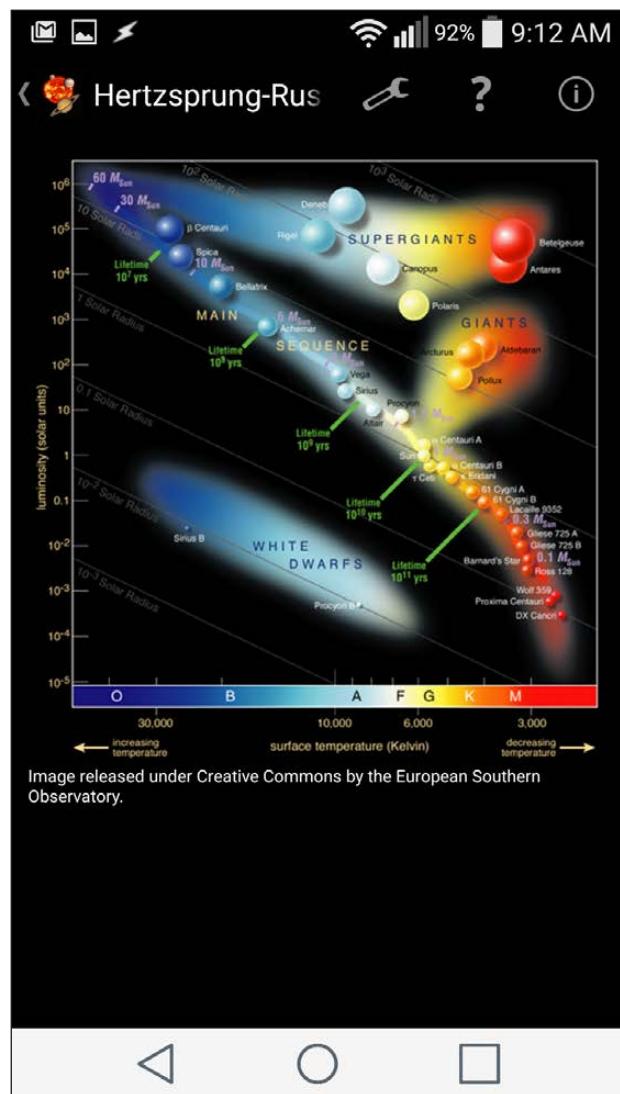
Figure 10. The Hertzsprung-Russell diagram shows how stars are categorized.

interest and pull up detailed information for it.

There also are sections where you can do searches for exoplanets and NGC/IC catalog objects. The remainder of the sections provide functions to do common astronomy calculations. You can handle coordinate calculations, astrophotography and telescope calculations.

The observation log helps you track your own information. There is a page to manage all of your astronomical equipment, like telescopes, eyepieces, filters and cameras. You also can log all of your observations, recording all of the details of interest. You can export your log, including whatever sections you need, so that you can incorporate it into some other database of your research.

Now you have no excuse for not going out and exploring the skies above you. In my next few articles, I plan to look at several other scientific applications that you can use on your Android devices for doing portable science.—Joey Bernard
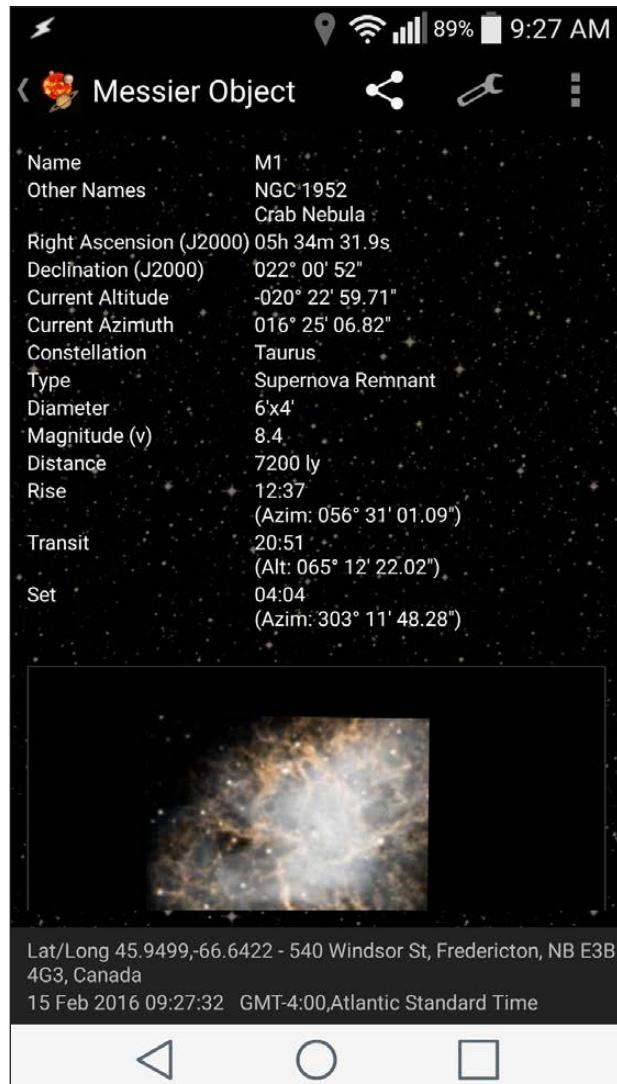
Figure 11. You can get detailed information on Messier catalog objects.

RETURN TO CONTENTS

2016
WOMEN IN TECHNOLOGY

SUMMIT

Executive women, entrepreneurs, and technology thought leaders from around the world will converge in Silicon Valley... *Join Us!*

June 5th - 7th
DoubleTree by Hilton
San Jose, California

witi.com/summit

WITI

Special Offer: Use promo code **WOMEN** by May 15th and **Save $100** Off Registration!

My +1 Sword of Productivity

If I'm being completely honest, I think the game-ification of a daily task list is a dumb idea. I also love it, and can't stress enough how well it works. Habitica might just be the way I get things done from now on.

I'm a perfectionist. If you've ever seen photos of my office (or hairdo), you might not think that's the case, but I assure you, it's true. Unfortunately, one of the big side effects of being a perfectionist is procrastination. Not laziness, but delaying or redoing tasks until you can get them just right. It can be crippling for productivity, and ironically, the rushed product that results often is sub-par to what would have been created in the first place.

Habitica turns the struggles with perfectionism back on the perfectionist. Although I honestly don't care very much about the swords and shields I earn by completing tasks, for some reason, the idea of losing HP for skipping a task is difficult for me to accept. I find myself doing extra "good habits" throughout the day just so my character is as good as he can be. Honestly, I'm surprised Habitica works for me. I still think it's dumb. I also can't stop striving for new experience levels and early task completions.

There's a great Web version of the free program at http://habitica.com, plus you can get mobile versions for iOS or Android via their

respective app stores. In fact, Habitica is so unique and surprisingly effective, it's easily my pick for Editors' Choice this month. If you think it sounds like a dumb idea, I completely agree. I also urge you to try it, because I find it incredibly awesome!—Shawn Powers

RETURN TO CONTENTS

Pandas

Reading data from CSV files, and then analyzing the data, is easy with Pandas.

REUVEN M. LERNER

Reuven M. Lerner offers training in Python, Git and PostgreSQL to companies around the world. He blogs at http://blog.lerner.co.il, tweets at @reuvenmlerner and curates http://DailyTechVideo.com. Reuven lives in Modi'in, Israel, with his wife and three children.

IN MY LAST ARTICLE, I started discussing the amazing world of data science—in which you explore and navigate through data, trying to find correlations that might be of interest to your business and/or point to trends you should consider.

Serious practitioners of data science use the full scientific method, starting with a question and a hypothesis, followed by an exploration of the data to determine whether the hypothesis holds up. But in many cases, such as when you aren't quite sure what your data contains, it helps to perform some exploratory data analysis—just looking around, trying to see if you can find something.

And, that's what I'm going to cover this month, using tools provided by the amazing Python ecosystem for data science, sometimes known as the SciPy stack. It's hard to overstate the number of people I've met in the past year or two who are learning Python specifically for data science needs. Back when I was analyzing data for my PhD dissertation, just two years

ago, I was told that Python wasn't yet mature enough to do the sorts of things I needed, and that I should use the R language instead. I do have to wonder whether the tables have turned by now; the number of contributors and contributions to the SciPy stack is phenomenal, making it a more compelling platform for data analysis.

In my last article, I described how to filter through logfiles, turning them into CSV files containing the information that was of interest. Here, I explain how to import that data into Pandas, which provides an additional layer of flexibility and will let you explore the data in all sorts of ways—including graphically. Although I won't necessarily reach any amazing conclusions, you'll at least see how you can import data into Pandas, slice and dice it in various ways, and then produce some basic plots.

Pandas

NumPy is a Python package, downloadable from the Python Package Index (PyPI: http://PyPI.python.org), which provides a data structure known an a NumPy array. These arrays, although accessible from Python, are mainly implemented in C for maximum speed and efficiency. They also operate on a vector basis, so if you add 1 to a NumPy array, you're adding 1 to every single element in that array. It takes a while to get used to this way of thinking, and to the fact that the array should have a uniform data type.

Now, what can you do with your NumPy array? You could apply any number of functions to it. Fortunately, SciPy has an enormous number of functions defined and available, suitable for nearly every kind of scientific and mathematical investigation you might want to perform.

But in this case, and in many cases in the data science world, what I really want to do is read data from a variety of formats and then explore that data. The perfect tool for that is Pandas, an extensive library designed for data analysis within Python.

The most basic data structure in Pandas is a "series", which is basically a wrapper around a NumPy array. A series can contain any number of elements, all of which should be of the same type for maximum efficiency (and reasonableness). The big deal with a series is that you can set whatever indexes you want, giving you more expressive power than would be possible in a NumPy array. Pandas also provides some additional functionality for series objects in the form of a large number of methods.

But the real powerhouse of Pandas is the "data frame", which is something like an Excel spreadsheet implemented inside of Python. Once you get a table of information inside a data frame, you can perform a wide variety of manipulations and calculations, often working in similar ways to a relational database. Indeed, many of the methods you can invoke on a data frame are similar or identical in name to the operations you can invoke in SQL.

Installing Pandas isn't very difficult, if you have a working Python installation already. It's easiest to use `pip`, the standard Python installation program, to do so:

```
sudo pip install -U numpy matplotlib pandas
```

The above will install a number of different packages, overwriting the existing installation if an older version of a package is installed.

As good as Pandas is, it's even better when it is integrated with the rest of the SciPy stack and inside of the Jupyter (that is, IPython) notebook. You can install this as well:

```
sudo pip install -U 'jupyter[notebook]'
```

Don't forget the quotes, which ensure that the shell doesn't try to interpret the square brackets as a form of shell globbing. Now, once you have installed this, run the Jupyter notebook:

```
jupyter notebook
```

If all goes well, the shell window should fill with some logfile output. But soon after that, your Web browser will open, giving you a chance (using the menu on the right side of the page) to open a new Python page. The idea is that you'll then interact with this document, entering Python code inside the individual cells, rather than putting them in a file. To execute the code inside a cell, just press Shift-Enter; the cell will execute, and the result of evaluating the final line will be displayed.

Even if I wasn't working in the area of data science, I would find the Jupyter Notebook to be an extremely clean, easy-to-use and convenient way to work with my Python code. It has replaced my use of the

text-based Python interactive shell. If nothing else, the fact that I can save and return to cells across sessions means that I spend less time re-creating where I was the previous time I worked on a project.

Inside Jupyter Notebook, you'll want to load NumPy, Pandas and a variety of related functionality. The easiest way to do so is to use a combination of Python `import` statements and the `%pylab` magic function within the notebook:

```
%pylab inline
import pandas as pd
from pandas import Series, DataFrame
```

The above ensures that everything you'll need is defined. In theory, you don't need to alias Pandas to `pd`, but everyone else in the Pandas world does so. I must admit that I avoided this alias for some time, but finally decided that if I want my code to integrate nicely with other people's projects, I really should follow their conventions.

Reading the CSV

Now let's read the CSV file that I created for last month's article. As you might remember, the file contains a number of columns, separated by tabs, which were created from an Apache logfile. It turns out that CSV, although a seemingly primitive format for exchanging information, is one of the most popular methods for doing so in the data science world. As a result, Pandas provides a variety of functions that let you turn a CSV file into a data frame.

The easiest and most common such function is `read_csv`. As you might expect, `read_csv` can be handed a filename as a parameter, which it'll read and turn into a data frame. But `read_csv`, like many other of the `read_*` functions in Pandas, also can take a file object or even a URL.

I started by trying to read access.csv, the CSV file from last month's article, with the `read_csv` method:

```
df = pd.read_csv('access.csv')
```

Unfortunately, this failed, with a very strange error message, indicating that different lines of the file contained different numbers of fields. After a bit of

thought and debugging, it turns out that this error is because the file contains tab-separated values, and that the default setting of pd.read_csv is to assume comma separators. So, you can retry your load, passing the `sep` parameter:

```
df = pd.read_csv('access.csv', sep='\t')
```

And sure enough, that worked! Moreover, if you ask for the keys of the Pandas data frame you have just created, you get the headers as they were defined at the top of the file. You can see those by asking the data frame to show you its keys:

```
df.keys()
```

Now, you can think of a data frame as a Python version of an Excel spreadsheet or of a table in a two-dimensional relational database, but you also can think of it as a set of Pandas series objects, with each series providing a particular column.

I should note that `read_csv` (and the other `read_*` functions in Pandas) are truly amazing pieces of software. If you're trying to read from a CSV file and Pandas isn't handling it correctly, you either have an extremely strange file format, or you haven't found the right option yet.

Navigating through the Data Frame

Now that you've loaded the CSV file into a data frame, what can you do with it? First, you can ask to see the entire thing, but in the case of this example CSV file, there are more than 27,000 rows, which means that printing it out and looking through it is probably a bad idea. (That said, when you look at a data frame inside Jupyter, you will see only the first few rows and last few rows, making it easier to deal with.)

If you think of your data frame as a spreadsheet, you can look at individual rows, columns and combinations of those.

You can ask for an entire column by using the column (key) name in square brackets or even as an attribute. Thus, you can get all of the requested URLs by asking for the "r" column, as follows:

```
df['r']
```

Or like this:

```
df.r
```

Of course, this still will result in the printing of a very large number of rows. You can ask for only the first five rows by using Python slice syntax—something that's often quite confusing for people when they start with Pandas, but which becomes natural after a short while. (Remember that using an individual column name inside square brackets produces one column, whereas using a slice inside square brackets produces one or more rows.)

So, to see the first ten rows, you can say:

```
df[:10]
```

And of course, if you're interested only in seeing the first ten HTTP requests that came into the server, then you can say:

```
df.r[:10]
```

When you ask for a single column from a data frame, you're really getting a Pandas series, with all of its abilities.

One of the things you often will want to do with a data frame is figure out the most popular data. This is especially true when working with logfiles, which are supposed to give you some insights into your work. For example, perhaps you want to find out which URLs were most popular. You can ask to count all of the rows in `df`:

```
df.count()
```

This will give you a total of all rows. But, you also can retrieve a single column (which is a Pandas series) and ask it to count the number of times each value appears:

```
df['r'].value_counts()
```

The resulting series has indexes that are the values (that is, URLs)

themselves and also a count (in descending order) of the number of times each one appeared.

Plotting

This is already great, but you can do even better and plot the results. For example, you might want to have a bar graph indicating how many times each of the top ten URLs was invoked. You can say:

```
df['r'].value_counts()[:10].plot.bar()
```

Notice how you take the original data frame, count the number of times each value appears, take the top ten of those, and then invoke methods for plotting via Matplotlib, producing a simple, but effective, bar chart. If you're using Jupyter and invoked `%pylab inline`, this actually will appear in your browser window, rather than an external program.

You similarly can make a pie chart:

```
df['r'].value_counts()[:10].plot.pie()
```

But wait a second. This chart indicates that the most popular URL by a long shot was /feed/, a URL used by RSS readers to access my blog. Although that's flattering, it masks the other data I'm interested in. You thus can use "boolean indexing" to retrieve a subset of rows from `df` and then plot only those rows:

```
df[~df.r.str.contains('/feed/')]['r'].value_counts()[:10].plot.pie()
```

Whoa…that looks huge and complicated. Let's break it apart to understand what's going on:

■ This used boolean indexing to retrieve some rows and get rid of others. The conditions are expressed using a combination of generic Python and NumPy/Pandas-specific syntax and code.

■ This example used the `str.contains` method provided by Pandas, which enables you to find all of the rows where the URL contained "/feed/".

■ Then, the example used the (normally) bitwise operator ~ to invert the logic of what you're trying to find.

■ Finally, the result is plotted, providing a picture of which URLs were and were not popular.

Reading the data from CSV and into a data frame gives great flexibility in manipulating the data and, eventually, in plotting it.

Conclusion

In this article, I described how to read logfile data into Pandas and even executed a few small plots with it. Next month, I explain how you can transform data even more to provide insights for everyone interested in the logfile.■

RESOURCES

Data science is a hot topic, and many people have been writing good books on the subject. I've most recently been reading and enjoying an early release of the *Python Data Science Handbook* by Jake VanderPlas, which contains great information on data science as well as its use from within Python. Cathy O'Neil and Rachel Schutt's slightly older book, *Doing Data Science*, also is excellent, approaching problems from a different angle. Both are published by O'Reilly, and both are worth reading if you're interested in data science.

To learn more about the Python tools used in data science, check out the sites for NumPy (http://numpy.org), SciPy (http://SciPy.org), Pandas (http://pandas.pydata.org) and IPython (http://IPython.org). There is a lot to learn, so be prepared for a deep dive and lots of reading.

Pandas is available from, and documented at, http://pandas.pydata.org.

Python itself is available from http://python.org, and the PyPI package index, from which you can download all of the packages mentioned here, is at http://PyPI.python.org.

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

RETURN TO CONTENTS

All about printf

Dave describes a super-useful scripting command stolen from the C standard I/O library.

DAVE TAYLOR

Dave Taylor has been hacking shell scripts since the dawn of the computer era. Well, not really, but still, 30 years is a long time! He's the author of the popular *Wicked Cool Shell Scripts* and *Teach Yourself Unix in 24 Hours* (new edition just released!). He can be found on Twitter as @DaveTaylor and at his tech site: www.AskDaveTaylor.com.

PREVIOUS Reuven M. Lerner's At the Forge	**NEXT** Kyle Rankin's Hack and /

IN MY LAST ARTICLE, I explored the surprising ability of the Linux shell to convert numeric bases on the fly, including this sweet little snippet that converts FF hexadecimal into decimal notation:

```
$ echo $(( 0xFF ))
255
```

And, I discussed how you even could use the handy `printf` command within scripts too, such as this command to display decimal numbers in octal and hexadecimal:

```
$ printf "octal: %o\nhex: %x\n" 42 42
octal: 52
hex: 2a
```

It's pretty neat stuff, but to be honest, I rarely find myself needing to convert numeric bases nowadays, so it's really something I file under "funky shell tricks". Your experience may be different, so it's still well worth learning anyway.

In this article, I thought it would be interesting to take a closer look at the `printf` command, because it is so darn powerful, but before going there, here's a quickie: some neat ways you can make your if-then statements be more succinct.

If/Then Statements

If you're like me, then you find yourself frequently writing conditional statement blocks in your shell scripts. Um, I mean:

```
if [ you're like me ] ; then
    you find yourself...
```

Well, you get the idea. In fact, conditional expressions are where sequences of code turn into more sophisticated programs, whether they're a half-dozen lines long or hundreds of lines.

A typical conditional expression actually might look like this:

```
if [ $(date +%w) -eq 0 ]; then
    echo "It's Sunday"
else
    echo "It's not Sunday"
fi
```

This is clear and readable, but it sure takes up a lot of vertical space in a shell script.

Fortunately, there are some ways you can tighten up things by using the && and || notations in your shell scripts.

The && notation means if what's invoked prior to the && ends with a success return code, do what's subsequent—for example:

```
test $(date +%w) -eq 0 && echo "Sunday"
```

If it's Monday afternoon when I run this code, I'll get no output, and the echo statement isn't even evaluated. But if it's Sunday, the above command will output appropriately.

The || notation offers the same basic functionality but with the opposite logic: if the return code of the command prior to the || returns a fail (non-zero) return code, then the subsequent command will be invoked:

```
test $(date +%w) -eq 0 || echo "It's not Sunday yet"
```

You also can make this even more succinct by using the [] notational shortcut for a test—just remember to include the closing] to ensure it's all well formed:

```
[ $(date +%w -eq 0 ] || echo "it's not Sunday yet"
```

The biggest limitation with this notation is that there's really no reliable and properly interpreted way to add an else clause.

You can try something like this:

```
cmd1 && cmd2 || cmd3
```

But because of precedence interpretation, it's likely to have cmd3 invoked if either cmd1 or cmd2 have a non-zero return code, which makes it functionality different from this:

```
if cmd1 ; then
  cmd2
else
  cmd3
fi
```

All is not lost, however, because you always can use a lot of semicolons to move that onto a single line:

```
if cmd1 ; then cmd2 ; else cmd3 ; fi
```

But, is it more readable? Is it really how you want to write your commands? Maybe. At least now you know!

The Ever-Helpful printf Command

Now, let's look at a completely different type of command, a command that is a built-in C programming language function that's so darn useful, it's now included in Linux as a standalone command.

In C and its brethren, the command shows up like this:

```
printf(formatstring, arg, arg);
```

This actually is a shortcut for the more general `fprintf()` command, which prepends the file handle and would look more like the following:

```
fprintf(stdio, formatstring, arg, arg);
```

It's not really relevant to this discussion, but hey, you should know this C programming nuance just so you know what's going on, right?

Okay, okay, back to the shell.

The `printf` command is basically the same, just without the parentheses and commas:

```
printf formatstring arg arg
```

Unlike the `echo` command, `printf` doesn't automatically append a carriage-return line-feed sequence, so you can end up with odd results like this:

```
$ printf "hello"
hello$
```

The format string allows a number of backslash-escaped sequences to alleviate this problem, notably `\n` to produce the end-of-line carriage return.

Indeed, go back to the first few paragraphs of this column, and you'll

Where things get more interesting is with the specifics of the format string.

notice I included this sequence:

```
printf "octal: %o\nhex: %x\n" 42 42
```

Now you know what those \n sequences mean: each produces an end-of-line sequence.

Additional escape sequences include \a for a bell (try it!), \b for a backspace, \t for a tab and \\ for a backslash character itself.

Where things get more interesting is with the specifics of the format string. All of these are denoted with the % symbol followed by the specific letter that specifies how the associated argument should be interpreted and displayed. Give it a decimal value but use %o, and it'll be output as octal (as shown earlier).

The most important sequences are:

■ %c for a character.

■ %s for a string (a sequence of characters).

■ %d for a decimal value.

■ %f for a floating-point non-integer value.

There are nuances, of course, and in particular, displaying floating-point numbers can be quite complicated because of the various notational conventions used. You can read the printf man page for much more detail on that.

Just about every format sequence also allows you to specify a field width and a precision, which is where all of this gets both complicated and interesting.

Let's consider the floating-point number 3.141597 and how `printf` might display it in different ways:

```
$ pi=3.141597
$ printf "%d\n" $pi
-bash: printf: 3.141597: invalid number
0
```

That shouldn't be a surprise; you can't interpret a floating-point number as an integer. Use `%f` instead:

```
$ printf "%f\n" $pi
3.141597
```

That's the default, and `printf` is showing its default precision for the floating-point value.

Let's see what happens if you specify a zero precision (that is, zero digits subsequent to the decimal point):

```
$ printf "%.0f\n" $pi
3
```

That makes sense. But, what if it's actually currency you're working with and you want to be able to ensure that you don't get weird values like $20.4342434 as a value:

```
$ printf "%.2f\n" $pi
3.14
```

Where this really gets interesting is when you want to line up values in columns, allocating 10, 15, 20 or more characters of space per field. That's the field width, and it appears prior to the decimal point on the formatting string specifier or by itself if there's no decimal point:

```
$ printf "X%15fX\n" $pi
X       3.141597X
```

You can combine things too:

```
$ printf "X%10.2fX\n" $pi
X      3.14X
```

You also can use field width specifiers with strings, which is particularly interesting:

```
$ printf "|%20s|%20s|\n" "one" "two"; printf "|%20s|%20s|\n"
"three" "four"
|                 one|                 two|
|               three|                four|
$
```

I'm running out of space, but I encourage you to check out the `printf` command and its many tricks to help you create more attractive output from your shell scripts!

And don't forget, if you have an idea for a shell script I should tackle—or a game I should consider—please don't hesitate to send an e-mail to ljeditor@linuxjournal.com.■

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

RETURN TO CONTENTS

Attend

SPTechCon
The SharePoint Technology Conference

June 27-30, 2016
The Sheraton Boston

BOSTON!

"This is the most informative conference I have been to in years. The technical discussions gave me a much better understanding of direction, advantages and challenges we face with this massive platform."
—Jamie Tyndall, Manager, Application Development, Business Information Group
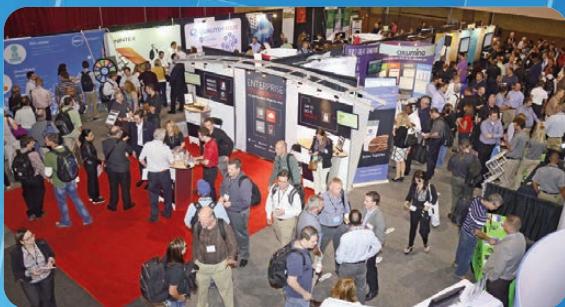
Learn what's new in SharePoint and Office 365!

Whether you want to learn about what's coming in SharePoint 2016, are still making the most out of SharePoint 2013 or even 2010, or getting started with Office 365, you will find the SharePoint and Office 365 training you need at SPTechCon.

"This was a great conference that addresses all levels, roles and abilities. Great variety of classes, great presenters, and I learned many practical things that I can take back and start implementing next week."
—Kathy Mincey, Collaboration Specialist, FHI 360

"As a newcomer to SharePoint, SPTechCon was an excellent way to begin learning more of its vast capabilities. This conference is a great way to hear about technical features and success stories of the product. Great vendors too. I will be following up with several of them about their products."
—Jeffrey Wahl, IT Services Manager, Carbonite, Inc.

www.sptechcon.com

Secure Desktops with Qubes: Introduction

Learn about next-generation desktop security with Qubes.

KYLE RANKIN

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.

THIS IS THE FIRST in a multipart series on Qubes OS, a security-focused operating system that is fundamentally different from any other Linux desktop I've ever used and one I personally switched to during the past couple months. In this first article, I provide an overview of what Qubes is, some of the approaches it takes that are completely different from what you might be used to on a Linux desktop and some of its particularly interesting security features. In future articles, I'll give more how-to guides on installing and

configuring it and how to use some of its more-advanced features.

When it comes to Linux security, server security tends to get the most attention. When you are hardening servers, you generally try to limit what any individual server does and use firewalls to restrict access between servers to only what is necessary. In a modern environment where a server is running only SSH plus maybe one or two other networked services, there are only a few ways for an attacker to get in. If a particular server does get hacked, ideally you can detect it, isolate that server and respond to the emergency while the rest of your environment stays up.

Desktop Linux security is a completely different challenge because of just how many *different* things you do with your desktop. Each action you take with your desktop computer opens up a new way to be compromised. Web browsing, especially if you still have certain risky plugins like Flash installed, is one major way a desktop can be compromised. E-mail is another popular attack vector since you need to open only one malicious e-mail attachment or click on one malicious phishing link for an attack to succeed. Linux desktops also often are used as development platforms, which means users might be downloading, building and executing someone else's code or running services directly on their desktop to test out their own code. Although some Linux users are smug when they think about all of the malware on other platforms, the fact is that the days when Windows was the only desktop OS in town are over, and these days, much of the malware is written in a cross-platform way so that it can run on many different operating systems.

The biggest issue with desktop Linux security is what's at risk if you do get hacked: all of your personal data. This could be anything from user names and passwords to important accounts like your bank or credit-card accounts, your social-media accounts, your domain registrar or Web sites you shopped at in the past that have your credit-card data cached. An attack could expose all of your personal photos or access to private e-mail messages. Attackers could leave behind a Remote Access Trojan that lets them get back into your machine whenever they want, and in the meantime, they could snoop on you with your Webcam and microphone. They even could compromise your SSH, VPN and GPG keys, which opens up access to other computers.

The core idea behind how Qubes provides security is an approach called

security by compartmentalization. This approach focuses on limiting the damage an attacker can do by separating your activities and their related files to separate virtual machines (VMs). You then assign each VM a certain level of trust based on the level of risk that VM presents. For instance, you may create an "untrusted" VM that you use for your generic, unauthenticated Web browsing. You then might have a separate, more-trusted VM that you use only to access your bank. You may decide to create a third highly trusted VM that has no network access at all that you use to manage off-line documents. If you also work from your personal computer, you may create separate VMs for personal versus work activities, with the work VM being more trusted. If you browse to a malicious Web site with your untrusted Web browser, the attacker won't have access to your banking credentials or personal files since you store those on different VMs. Qubes even provides disposable VMs: one-time-use VMs that are deleted completely from disk after the application closes.

How Qubes Works

Although you certainly could use any of the virtual machine technologies out there to set up multiple VMs on your regular Linux desktop, that kind of arrangement can end up being pretty clunky, especially if you don't want multiple desktop environments running inside their own windows. There also are all kinds of mistakes you could make with that kind of set up that would eliminate any security benefits you might get. For instance, how should you share files or copy and paste between VMs securely, and how do you keep all of those VMs up to date with security patches?

Where a traditional Linux distribution made it easy for you to get all of the software you wanted to use without having to download and compile it all, Qubes provides a number of extra tools that makes it easy to manage a desktop full of different virtual machines all with different levels of trust. Qubes also approaches all aspects of the desktop with security at the forefront and uses secure defaults throughout the OS. In doing so, Qubes makes it more difficult (but not impossible) for you to shoot yourself in the foot.

Qubes uses Xen to provide all of its virtualization (if you want to know why Qubes chose that over other technologies, see the FAQ on the Qubes site). Instead of each VM having its own complete desktop

environment, Qubes uses the more-privileged dom0 Xen VM as a host for the desktop environment (currently Qubes gives you the choice of KDE or XFCE, although the community has contributed others), and the other VMs display individual application windows within dom0's desktop environment.

So, launching Firefox in Qubes behaves much like you would expect in any other desktop distribution. The main difference, however, is that Qubes lets you color-code each of your VMs based on level of trust ranging from red (untrusted) to black (ultimately trusted) with a number of different rainbow colors in between.

When you launch an application from an application VM (appVM, in Qubes parlance), the VM starts up if it wasn't started before, then the application appears with a window border that is colorized based on the color you assigned its appVM. So, if you have two instances of Firefox on your desktop at the same time, you can tell your untrusted Web browser from your banking Web browser, because the untrusted one might be colored red while your banking browser might be colored green. Figure 1 provides a screenshot from Qubes' documentation that
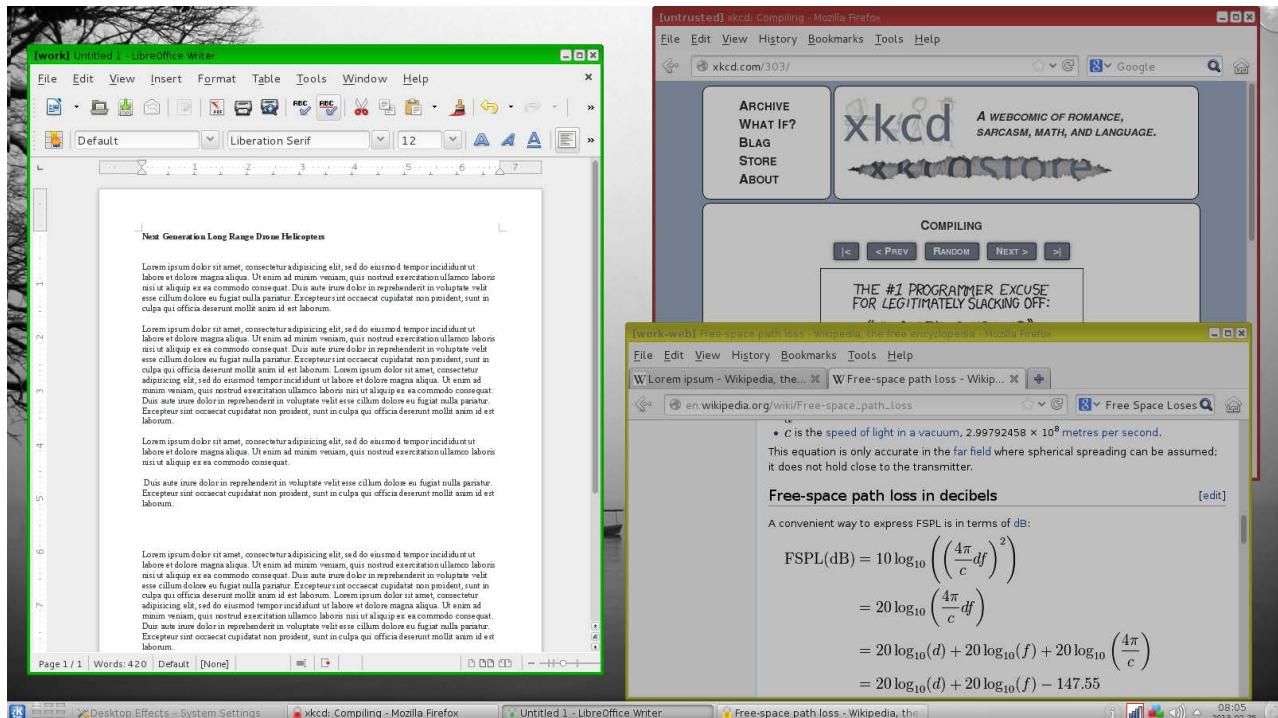
Figure 1. Multiple Windows with Different Colors
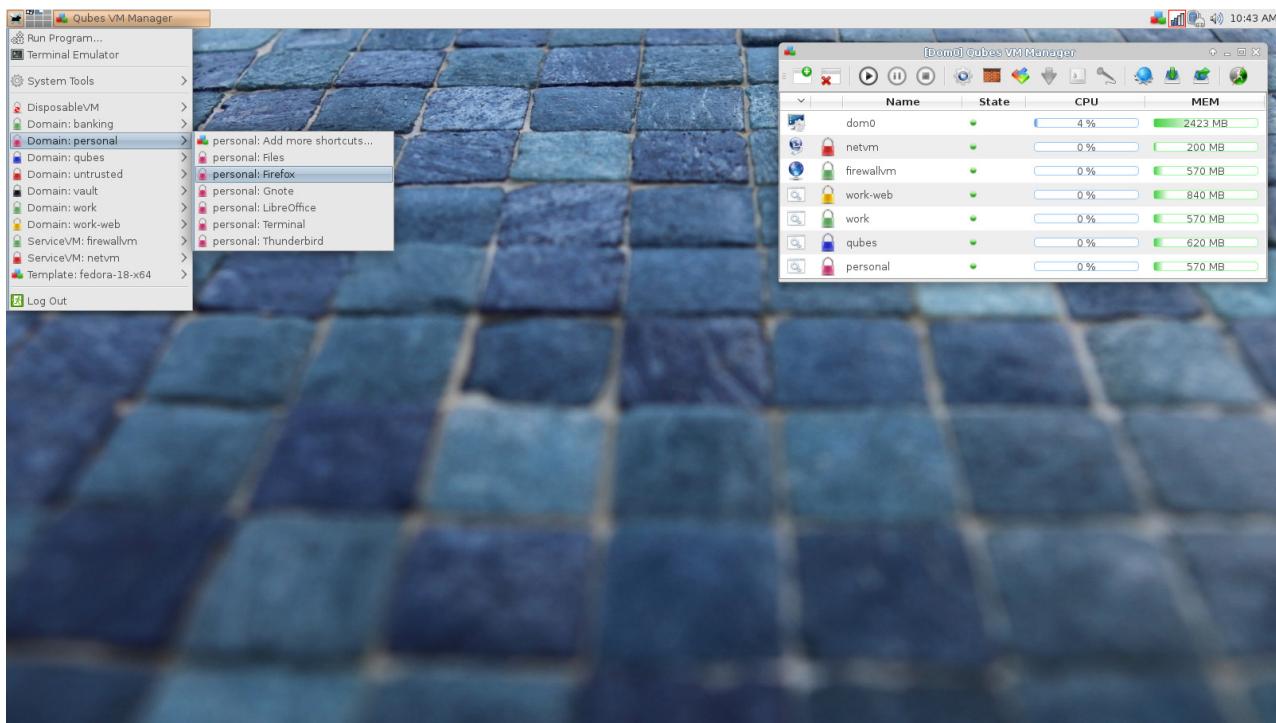
Figure 2. Qubes Application Menu

demonstrates the point.

Since the dom0 VM has privileged access to data about the other VMs in Xen, Qubes goes to extra lengths to protect it by having only the desktop environment run from it and by removing all network access from dom0. You are encouraged to do as little as possible in dom0, and instead, you should use appVMs for any applications you want to run. Qubes even intentionally makes it more difficult to copy files to or from dom0 compared to copying them between appVMs. In the dom0 desktop environment's application menu, each VM has its own submenu where you can launch each of its applications (Figure 2). Qubes provides tools so all of those submenus don't become too unwieldy, and you can select which applications appear under which appVM's menu.

Sharing Information between AppVMs

When you have multiple windows open, however, that raises the question of how do you copy and paste? An insecure approach might be to share the clipboard between all windows, but then the risk

would be that if you logged in to a Web site in a trusted Web browser by copying and pasting from your password manager, that password would be readable by any other appVMs that happened to be running. Instead, Qubes provides a two-tier approach to clipboards. Each appVM has its own clipboard, and you can copy and paste within that appVM as normal. If you want to copy from one appVM and paste to another, once you have put the data in one appVM's clipboard, you press Ctrl-Shift-c to put it in the global clipboard, then highlight the window you want to paste into and press Ctrl-Shift-v to paste that data into that VM's clipboard and wipe it from the global clipboard. Then you can paste inside that application as normal. It's definitely an extra cumbersome step, but you would be surprised at how quickly you adapt to: Ctrl-c, Ctrl-Shift-c, change window, Ctrl-Shift-v, Ctrl-v. It definitely helps you prevent accidentally pasting information into the wrong window.

Qubes also provides a command-line tool and right-click menu options in the GUI file manager so you can copy or move a file between appVMs. When you attempt this, you get a prompt in a black-bordered window the appVM doesn't control so you can accept this file transfer. Even then, the file doesn't appear wherever you want on the destination VM (otherwise an attacker could overwrite important files with backdoored versions). Instead, the files show up in a QubesIncoming directory inside your home directory.

TemplateVMs, Persistence and Backdoor Protection
Another area where Qubes provides an extra level of protection for a desktop user is in how it handles persistence. If attackers compromise a normal desktop, they can install backdoored versions of utilities, like ls or bash, or add extra programs that are triggered to start at boot. With Qubes, appVMs are based off templateVMs that have base installs of Fedora, Debian or Whonix by default (the community has provided templates for other popular distributions). When you create a new appVM, you choose which template it is based from, and when you start it, that appVM gets a read-only version of that template's root filesystem. Although the user inside the appVM still can install software or change the root filesystem, when that appVM shuts down,

This means your browser history and settings will stick around, but if an attacker did compromise your browser and tried to install a backdoor into bash or Firefox, the next time you rebooted that appVM, the backdoor would be gone.

all of those changes are erased. Only the /rw, /usr/local and /home directories persist. This means your browser history and settings will stick around, but if an attacker did compromise your browser and tried to install a backdoor into bash or Firefox, the next time you rebooted that appVM, the backdoor would be gone.

Also by default, appVMs do not launch any common init services like cron automatically. That means attackers also couldn't just add a user cron entry that launched the backdoor. Although it's true that attackers could store a malicious program in your appVM's home directory, the next time you reboot the appVM, the program no longer would be running, and they would have no way to launch it again automatically.

So, how do you install software? Because each appVM uses a root filesystem based on its templateVM, when you want to install new software, you launch the software manager from the templateVM and install the application with yum, apt-get, the GUI equivalent or whatever other method you normally would use to install the software. Qubes then detects any new application menu items you've added and makes them available to the appVMs based on that template.

The only gotcha is that those newly installed applications are unavailable to appVMs until those appVMs restart. Because compromising the templateVM compromises every appVM based on it, Qubes generally encourages you to leave templateVMs off, to not run general applications from them and to turn them on only when you add trusted software. Although this does add an extra bit of work when you want to install software, it also provides a nice benefit

in that when you need to apply a security update, you just need to update the templateVM, and when you restart each appVM, it will get the update.

Network Security with netVMs

Another way Qubes provides security is by compartmentalizing the network. Upon installation, Qubes will create a few special system VMs called network VMs (netVMs), named sys-net, sys-firewall and sys-whonix. The sys-net netVM is assigned any networking hardware on your host, so it's unavailable to any other VM. Because this netVM is the only one with an IP on the external network, it's considered untrusted and colored red. You use Network Manager to configure this netVM with any credentials it needs to connect to wireless networks, and its Network Manager applet shows up on your desktop as normal. The sys-firewall VM (technically classified as a proxyVM) is colored green and connects to sys-net for its network access. By default, any appVMs you create then use sys-firewall for their network access.

Why all this complexity? First, sys-firewall acts as a true firewall for all of your appVMs. Although by default all appVMs can talk to the Internet unrestricted, Qubes provides a GUI tool that makes it easy to lock down individual appVMs so that they can access only certain hosts on the network. For instance, you could restrict your banking appVM so that it can talk only to port 443 on your banking Web site or restrict an e-mail appVM to talk only to your remote mail server. You even could restrict other VMs so that they could talk only to hosts on your internal network. Anyone who wants to attack one of your appVMs has to go through sys-net and sys-firewall. This also means if attackers do compromise an appVM, they don't have direct access to network hardware, so they can't, for instance, automatically connect to a different wireless access point.

The sys-whonix VM acts like sys-firewall except that it automatically sets up a secure Tor router. Any appVMs that use sys-whonix instead of sys-firewall or sys-net for their network have all of their traffic routed over Tor automatically. Qubes also provides an anon-whonix appVM by default that uses the security and anonymity-focused

distribution Whonix, and it includes the Tor browser and routes all traffic through sys-whonix by default.

I'm sure you already can see a number of areas where Qubes provides greater security than you would find in a regular Linux desktop. Hopefully, you have a sense of what a different approach Qubes takes from what you might be used to. With Qubes, you find yourself thinking much more about how you should isolate files and information and what attackers could get if they successfully compromised one of your appVMs. Even the extra copy-and-paste and file-copy steps force you to confront whether you are transferring information between an untrusted VM to a trusted one and think through the implications. I've found the extra security measures actually let me relax a little bit more than I would otherwise, because for instance, I know an e-mail attachment I open in a disposable VM can't do me much harm, or a malicious Web site in my untrusted Web browser can't access anything of value.

I've touched on only some of the higher-level security features in Qubes with this article. In my next article, I will describe how to download and install Qubes, explain how to use Qubes as a desktop OS, including some of the basic features of the Qubes VM Manager and other Qubes-specific tools, and give some examples for how you might organize your day-to-day desktop use across appVMs. I'll follow up with an article describing more-advanced Qubes features, including split-GPG (a method that allows appVMs to use your GPG private key without having direct access to it), how to manage links more securely with default application handlers, how to open e-mail attachments automatically in disposable VMs, and how to create a usbVM that isolates all of your USB devices for you (and why you would want to do that).■

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

RETURN TO CONTENTS

Jarvis, Please Lock the Front Door

I'm like Tony Stark, but my Jarvis is named Alexa.

SHAWN POWERS

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.

YEARS AGO, we put out a request for articles on home automation. About the time Eureka came out on TV, people wanted to have their very own SARAH (Self Actuated Residential Automated Habitat), and it seemed like the perfect time for nerds everywhere to make their houses smart. The problem was, although a few programs existed (MisterHouse for example), the hardware wasn't really reliable or highly available. The X10 company was about the only game in town hardware-wise, and it tended to be glitchy without much advantage over traditional switches.

In recent years, a glut of products have been dumped onto the market, all toting options for

automated lighting, wireless switches and so on. Unfortunately, most were very closed and proprietary, forcing users to stick to a specific brand. That probably was the goal, but it backfired, because the concept of branding my house with proprietary hardware and software was anathema. Thankfully, times are changing, and the product that made me jump into the home automation pool with both feet is a surprisingly proprietary one: Amazon Echo (but, more on that later).

Wireless Communication

Several brands of home automation devices use standard Wi-Fi (2.4GHz) to communicate. At first glance, that seems like a good idea. Unfortunately, the 2.4GHz frequency is so cluttered, adding more devices might be counterproductive. It's also a high-bandwidth type protocol, which is just not needed for simple switching and communication.

Most home automation devices, regardless of brand, focus on the 900MHz spectrum. You might remember 900MHz from the days of cordless phones (not cell phones, rather the old cordless phones from the 1990s). For several reasons, 900MHz network devices have never really gone mainstream, which means the frequency isn't oversaturated. It also penetrates walls better, making it perfect for connecting devices around
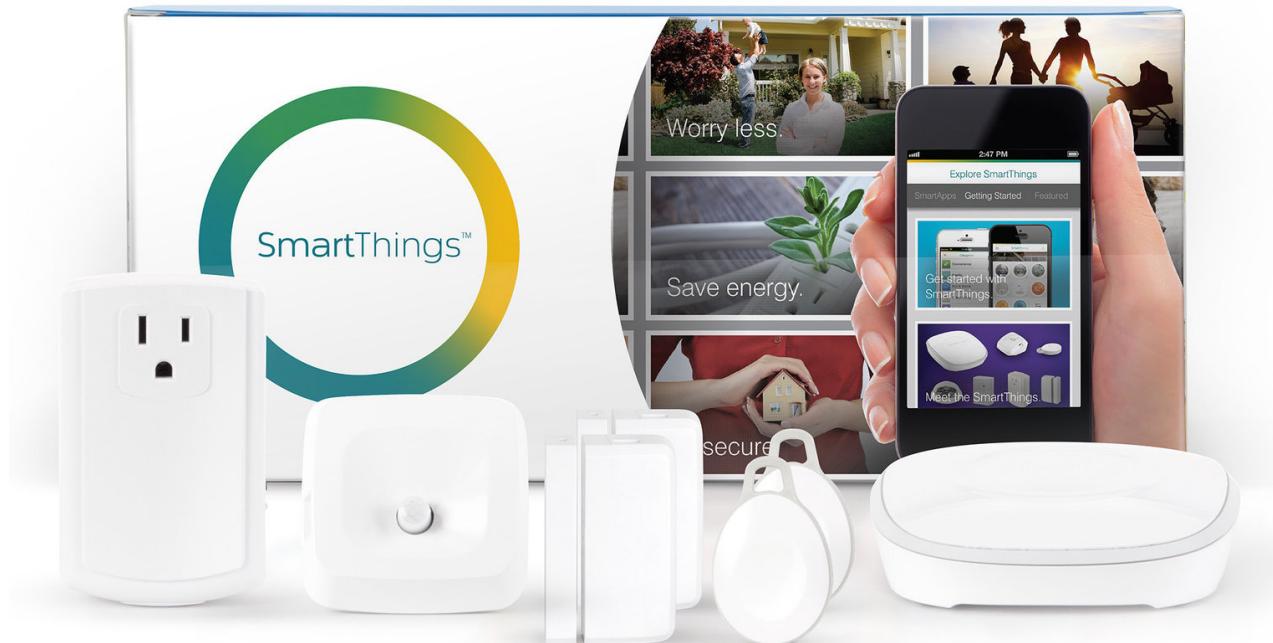
Figure 1. SmartThings is my choice for the most flexible platform upon which to build.

Unfortunately again, even though SmartThings supports Z-Wave and ZigBee, that doesn't mean it has native support for all devices that use Z-Wave or ZigBee.

your home.

Unfortunately, everyone has been trying to become "the standard" in home automation, making the various brand names often incompatible with each other. When I decided to start using home automation devices, I wanted something that was compatible with the most products. For me, that meant SmartThings from Samsung. It supports the very common Z-Wave protocol and the ZigBee protocol, which is similar, but is based on an actual IEEE standard (IEEE 802.15.4).

Unfortunately again, even though SmartThings supports Z-Wave and ZigBee, that doesn't mean it has native support for all devices that use Z-Wave or ZigBee. It might be able to communicate with them wirelessly, but it's sorta like using a standard phone line. Just because I can call someone in Germany doesn't mean we'll be able to understand each other once connected. That downfall is actually another reason I chose SmartThings over the alternatives. Even though it doesn't support all devices, it does have a *very* open development platform that allows users to write device drivers for any product the SmartThings hub can find. That even includes devices it can't communicate with directly, like my Nest thermostat. The developer community is very active, and drivers for devices are usually easy to implement. Be sure to google the device before you buy it though, because some products are just so closed, communicating with them is currently not possible. (I'm looking at you, Insteon.)

The Actual Automation

This is starting to feel like an advertisement for SmartThings, but really it's just my opinion based on lots of research and time using it. When it comes to using your phone to turn lights on and off, or lock

doors, most brands work just fine. With SmartThings, however, you can go one step further and write programs that have actual intelligence. Those programs can be shared, and many are available in the SmartThings Marketplace. You can add these "SmartApps" to your system and provide a wide variety of actions based on events.

For example, if SmartThings detects my front door opening (due to either detecting motion on the front porch via motion sensor, door opening via door sensor or lock unlocking via manual code entry on my Z-Wave deadbolt), it checks the current time of day and compares that to the sunrise/sunset. If it's dark, it turns my entryway light on for five minutes, then turns it back off. That might seem like a fairly complicated event for a simple action, but that's the beauty of programmatically dealing with mundane house activities. It requires no thought, and the house responds intelligently every time, without any interaction on my part.

If you're not a programmer, that doesn't mean you're left out of the automation game. SmartThings (and in all fairness, several other platforms too) integrates with If This Then That (http://ifttt.com) for trigger-based actions that will interact with your home. Want to get a call or text if your house senses motion? You could pay for an expensive security system, or you could just have IFTTT call you
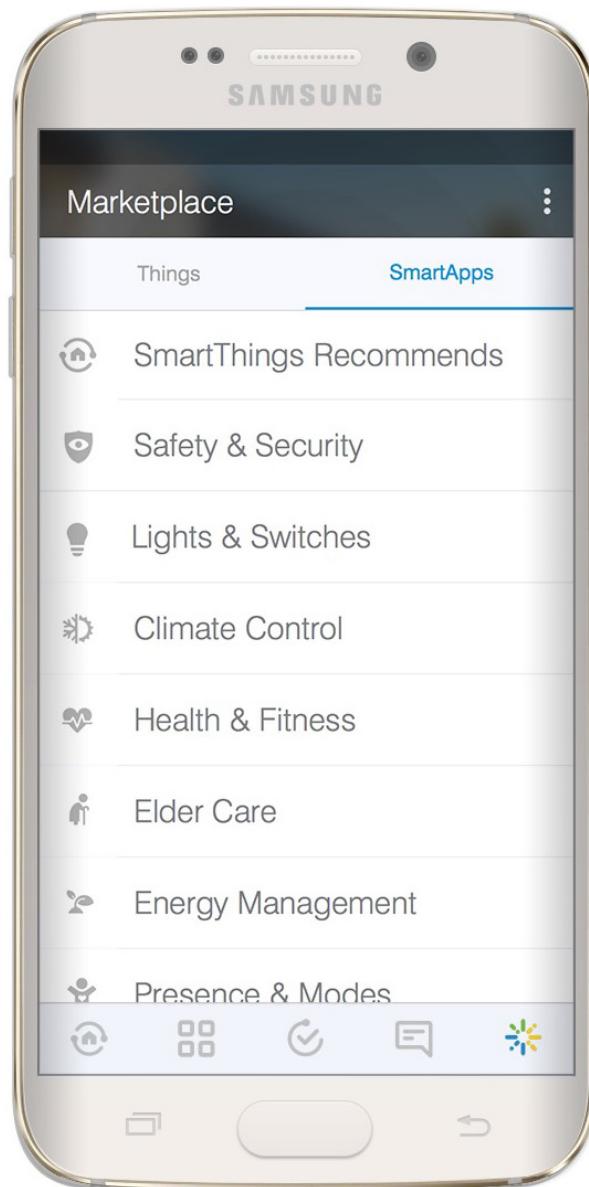
Figure 2. The Marketplace is full of SmartApps ready to download.

Figure 3. IFTTT with SmartThings makes your smart house even smarter.

when SmartThings triggers a motion event. With the flexibility of IFTTT, your smart house becomes one more thing you can add to your recipes.

Call Me Alexa

I mentioned earlier that the Amazon Echo is really what convinced me to start delving into the home automation world. That's largely because although the Android app for SmartThings is very nice, it's not very nice if you're a guest. I do have many Z-Wave physical switches installed so lights can be manipulated in the traditional way, but if you have a smart house, you want it to be convenient for people. That's where the integration with Amazon Echo comes into play.

Interacting with SmartThings via Alexa can happen in two basic

Figure 4. Alexa can do IFTTT things even if you don't use SmartThings!

ways. First, there is direct integration with SmartThings. Unfortunately, the direct integration is limited to turning switches on and off. That sounds great, and it is, but there's so much more I'd like Alexa to do. That's where Alexa and IFTTT comes into play. In fact, I do just as much integration with Alexa and IFTTT as I do with SmartThings.

Alexa isn't perfect, and sometimes the verbiage has to be perfect in order for it to function properly. That said, it's nice to crawl into bed and say:

- "Alexa, turn off all lights." ("All lights" is an Alexa group of SmartThings switches all around our house.)

- "Alexa, trigger door locks." (This starts an IFTTT recipe that tells SmartThings to lock all the deadbolts.)

- "Alexa, bedroom lamps to 20." (This turns on our reading lamps and dims them to 20%.)

- "Alexa, turn on box fan." (This activates an outlet, turning on the fan my wife needs in order to sleep at night.)

- "Alexa, turn on night mode." (This triggers a SmartThings virtual switch, which actually activates a certain string of events, which happens to include all of the above actions.)

Don't Skimp on Physical Switches

If I lived alone and never had houseguests, I wouldn't need any physical switches in my house at all. (I'd also smell worse and probably be unshaven.) But because I live in a house with my family, it's very important that our "smart house" is adding value instead of adding unneeded complexity. For example, it's sometimes difficult to get Alexa to do exactly what I want. For some tasks, it's required to say, "Alexa, trigger <ACTION>", and for other things, it's, "Alexa, turn on <ITEM_OR_TASK>."

   Plus, if you have a house full of people, or if the television is playing, Alexa often can't hear what you're saying clearly. In situations like that, it's vitally important to have a plain-old switch on the wall to turn things on and off. Thankfully, adding a smart switch in place of a traditional light switch often gives you the best of both worlds. You can use traditional (or LED) light bulbs and get the automation by utilizing a wall switch that will turn the lights on and off via physical switching or wireless control. In fact, rather than buying smart light bulbs, I generally try to replace switches instead. A smart bulb is effective only if it's powered on—and if you keep your old switch, you inevitably will shut off the power, making your smart bulbs as dumb as ever.

Getting Started

Rather than just telling you to go buy a specific product, it's important to figure out what you want to accomplish with home automation. I eventually want every possible aspect of my house to be automatic, scriptable or voice-controlled. My wife would be just as happy with

Alexa and a few smart light bulbs. When you stick to something basic, like an Amazon Alexa and some Phillips Hue lights, the integration is really simple.

For my personal goals, using a central hub like the Samsung SmartThings Hub was ideal. I can add devices to it. It supports a wide variety of brands and technologies. The open nature of the system means customizations can be made, even some specific to my needs, assuming I learn the programming language. Rather than taking my word for it, I urge you to research the various brands and see what fits into your world the best. Insteon, for example, has the nicest-looking switches available and a wide variety of products for sale. As long as you're happy being limited to those products, it's hard to beat the quality.

Show and Tell

After all the warnings about researching for yourself, it seems only fair to list my experiences with the handful of products I'm currently using (in no particular order).

Samsung SmartThings—Home Monitoring Kit ($249): This is a pricy kit, but it includes not only the hub (only one hub required

Figure 5. The Samsung SmartThings Home Monitoring Kit is a great way to start your adventure.

per house, regardless of the number of devices you have), but also a smart outlet, motion sensor (with temp sensor) and two door sensors (with vibration detection). This is a great way to start, because it gives you multiple devices that sense things and an outlet to "do" something. You easily can add a few smart light bulbs and make it a complete system.

Amazon Echo ($179): There's actually a few other options for using Alexa. The new Echo Dot is $89, and it has all the same home automation features, it just doesn't have the nice sound system included with the full-blown Echo. It does have audio-out, however, so you potentially could build yourself an even better Alexa. Also, Amazon Fire TV includes Alexa. That version doesn't have all the functionality of the Echo or Echo Dot, but it does support all the home automation features.

Figure 6. The original Amazon Echo or the new Echo Dot both work well for home automation.

Figure 7. This is a great device for dimming lamps. Unfortunately, it's also great for ruining televisions. Be careful!

Leviton DZPD3-1LW Z-Wave Lamp Dimmer ($33.95): Rather than install a bunch of expensive light bulbs in our living room lamps, I just bought this dimmer and plugged all the lamps (with extension cords) into the single dimmer module. It allows for on/off functionality, plus it dims all the lamps as well. (Note: it's *very* important not to plug electronic equipment into the lamp dimmer. It basically provides a brown-out situation and can ruin things like televisions!)

Enerwave ZWN-SC7-W 7-Button Scene Controller ($42.99): The problem with the lamp dimmer module is that it doesn't include any switches.

Figure 8. Enerwave 7-Button Scene Controller—I use only the single big button now, but all seven are usable.

Since I replaced a wall-mounted outlet switch with the dimmer module, I just used that junction box and wired this controller in place of the old switch. It required custom programming in order to make it work with SmartThings, but the code is stable and freely available from the SmartThings development community. I use only one of the seven buttons at this point, but I'll use the others for things like lowering my projector screen in the future.

Kwikset 910 Z-Wave Smartcode Deadbolt ($150): This was an expensive purchase, but since we had to replace our locks anyway (we recently moved in and have no idea who has keys to the old locks), it was a good time to make the investment. I'm really glad we did. Using one of the door sensors from the kit above, I have the deadbolt set to lock automatically when the door has been closed for a minute. That means as long as the door is shut, it's effectively locked as well. I sleep a lot better knowing the doors are locked whether I checked them or not. Plus, the programable codes means we won't have to re-key the lock later, just change codes if they are compromised.

Aeon Labs Aeotec Z-Wave Home Energy Meter ($94.90): I can't get this to work. I basically want to monitor the usage on our home circuit breaker to see when and what electricity we're using. Unfortunately, I can't get the dumb thing to work with my SmartThings hub. Others have succeeded, so I just need to spend more time on it. I mention

Figure 9. This Kwikset deadbolt was expensive, but I think it was worth it. There's a more expensive version available too, with individual buttons for the numbers.

Figure 10. This Aeon Labs Home Energy Meter will be cool, if I can ever get it working!

it because I don't want everyone to think things always go smoothly. This is technology after all, and technology is frustrating!

GE 12727 Z-Wave Lighting Control Smart Toggle Switch ($40): These look at a quick glance like a standard light switch. The "flipper", however, is always in the center position and can be bumped up or down to trigger the on/off action. It doesn't actually switch to up or down, but activates a switch and then returns to the center position. This is so that if you turn the light on with the switch, then off

Figure 11. It feels weird, but it works like any other switch—sorta.

Figure 12. Perfect for dimming lights, but make sure they're dimmable bulbs!

with electronic automation, the switch isn't in the "wrong" position. It takes a little getting used to, but the function is straightforward even for folks who don't know it's a smart switch. (They just think it's weird!)

GE 12724 Z-Wave Smart Dimmer ($40): This is the same as the above toggle switch, but includes dimming technology. It's also a wide "paddle" type switch. It functions the same way, in that it returns to a middle position, but it's less noticeable with this switch than with the toggle switch. You basically tap the top to turn lights on and the bottom to turn lights off. To dim/undim, you just hold the button up or down.

SmartenIt Three-Button ZigBee Switch ($49.99): This is a battery-operated switch, which functions a bit like the seven-button switch mentioned earlier, but with four fewer buttons. Also, since it's battery-operated, you can place it anywhere. We actually put this

Figure 13. The SmartenIt Three-Button ZigBee Swtich is great for sticking places that don't have power lines.

by our bed so we can turn our lamps on/off/20%-dim with the touch of a button. We can do the same with Alexa, but it's nice to have the tactile option as well. This is one of the only available battery-operated switches that I've found, which is frustrating because battery-operated switches are perfect for a smart house that doesn't require your switches to be hard-wired into the house power.

GE Link Wireless A19 Smart Dimmable LED Light Bulb, 60-Watt Equivalent ($15): These bulbs are cheap. At $15, they're hardly more expensive than a non-smart LED bulb, but they are dimmable and work with SmartTools. I use these in our bedroom lamps. In fact, smart bulbs are best used in lamps where the cords are plugged in so the bulbs have constant power. I used bulbs instead of a common dimmer for both lamps next to our bed so we can have them on/off independently. The three-button switch mentioned previously manages both lamps at once, but Alexa can manage them individually or as a group. It's win/win. The only thing I don't like about them is that although they do dim, it's not an even distribution from 1–100%. There's hardly any variation

between 20–100%, and the bulbs really get dim only when you go to 10% or below. That's just a nitpick, however, and the price of the bulbs makes it a complaint I can happily live with.

Is Your Home Smart?

As with most of my articles, I write about things I love. If you have implemented smart technology in your house, I'd love to hear about it. Just put something like [SMART HOME] in the subject line, and drop me a message at shawn@linuxjournal.com. I'm sure I'll follow up with cool things I do or horrible mistakes I make along the path to my future robot army—er, I mean smart home. I'll be sure to share the tips you send if I find them useful.■

Figure 14. These GE LinkWireless Smart Dimmable LED bulbs are cheap, but occasionally difficult to pair properly. The cost makes them worth the hassle.

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

RETURN TO CONTENTS

Linaro Announces Software Reference Platform for ARM

With the launch of its Software Reference Platform for ARMv8-A processors, Linaro is proud to enable both a complete end-to-end open-source server software stack and access to enterprise-class ARM-based server hardware for developers. The build for the Linaro Enterprise Group is a complete reference implementation for ARM servers, including open-source boot software and firmware implementing the ARM Trusted Firmware, UEFI and ACPI standards, a Linux 4.4 kernel, tested latest Debian and CentOS distributions, OpenStack, OpenJDK, Hadoop and Spark. A build for the Linaro Mobile Group also is available. Linaro expects the platform to be utilized by Linaro members and the wider community for enterprise products and cloud-instance development and deployment. During 2016, the Linaro Software Reference Platform releases will provide market-segment-specific application stacks to support an increasing range of data-center, networking and home-gateway applications.
http://linaro.org

Kolab Systems AG and Collabora's CloudSuite

The chemistry created by the Kolab Systems-Collabora Productivity partnership enabled CloudSuite, the first 100% open-source, enterprise-grade cloud office suite. Kolab Systems' contribution is its Kolab open-source groupware and collaboration framework; Collabora Productivity is the architect behind LibreOffice Online, the cloud-based office productivity suite. The integration of CloudSuite into Kolab allows users to work on documents simultaneously using a full-featured on-line office suite. Collaboratively, they can compose text documents, fill spreadsheets and design presentations, even from different locations. Documents can be saved in Microsoft-compatible and Open Document formats. The CloudSuite offering also includes Collabora Office, a professional LibreOffice distribution, for off-line use on the desktop. An important motivation for both firms in this effort involves a desire to move away from closed and insecure solutions to ones that respect users' freedoms, protect their privacy and guarantee their work will not be locked away in proprietary formats.
http://kolabsys.com, http://collaboraoffice.com

AdaCore's SPARK Pro

With this new version of the SPARK Pro toolset, AdaCore comes one step closer to its goal of making the writing of proven software both efficient and pleasant. As part of its new SPARK Pro 16 integrated development and verification environment, AdaCore further simplifies software engineers' transition to greater reliance on static verification and formal proofs sans need for expertise in mathematical logic. SPARK Pro 16 also provides enhanced coverage of the SPARK 2014 language features and now supports the Ravenscar tasking profile, thus extending the benefits of formal verification methods to a safe subset of Ada 2012 concurrent programming features. This new SPARK Pro can generate counter-examples to verification conditions that cannot be proved, making it easier for developers to find defects in the functional code or in the supplied contracts. Finally, SPARK Pro 16 also improves the handling of bitwise/modular operations, and the product's proof engine now includes the Z3 SMT solver.
http://adacore.com
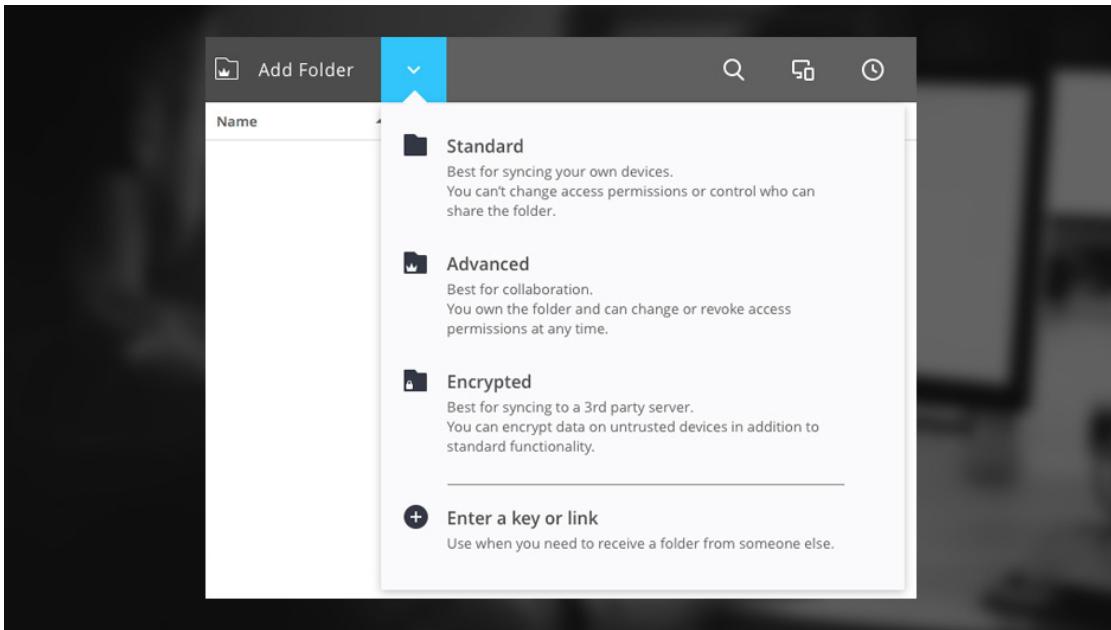
Canonical and BQ's Aquaris M10 Ubuntu Edition Tablet

Canonical's broad vision for Ubuntu Linux is to offer a single, converged personal computing experience across devices. This vision has taken a leap forward with the launch of the Aquaris M10 Ubuntu Edition tablet, "the first fully converged Ubuntu device" released by Canonical together with its European partner, BQ. As a converged device, the Aquarius M10 with dynamically adaptive user experience is capable of providing both a true tablet and the full Ubuntu desktop experiences. The former becomes the latter by simply plugging in a monitor via the HDMI port or connecting a keyboard and mouse via Bluetooth. Canonical hopes to offer customers everything they have come to expect from an Ubuntu PC, now on the tablet. Ubuntu is already the preferred desktop OS for more than 30 million users worldwide, and the first Ubuntu phones have proven successful as well. With this latest software release and the launch of the Aquaris M10 Ubuntu Edition, Canonical notes that Ubuntu is now the only platform that runs both a mobile-based full-touch interface and a true PC experience from a single smart device. Hundreds of apps and scopes (content-specific home screens) already are available in the Ubuntu App Store.
http://ubuntu.com/tablet

BitTorrent Inc.'s Sync

The Sync application from BitTorrent Inc. is simple yet powerful, offering the ability to move large amounts of data directly between devices. The new Sync 2.3 provides new features to support power users seeking to unlock Sync's full potential. BitTorrent says that Sync is simple to use in combination with a cloud provider's storage space or NAS to ensure data redundancy for backup. Of course, having data stored anywhere on third-party infrastructure is a serious concern. The new Encrypted Folder feature in Sync 2.3 solves this issue by providing the ability to encrypt data at rest on any designated location. Encrypted Folders can be shared to read-only nodes to provide an off-site snapshot of data without providing direct access, or users can deploy multiple Encrypted Folders to increase the reliability of a peer swarm. Another new feature is support for moving data to and from an SD card on Android 5+ devices. Finally, Selective Sync support is available on all flavors of Sync for Linux. Users download only the files they need, when they need them, without having to replicate entire folders on their beloved Linux boxes.

http://bittorrent.com

ACI Worldwide's UP Retail Payments

As customers of global-payments solution provider ACI Worldwide retire aging platforms, they are clamoring for Linux-based options. ACI Worldwide has responded with a Red Hat Enterprise Linux version of UP Retail Payments, a complete and customizable end-to-end enterprise payments solution. UP Retail Payments, targeted at banks and processors, combines the benefits of ACI's BASE24 and UP BASE24-eps solutions. BASE24 is ACI's retail payment platform; UP Framework orchestrates any payment type, channel, currency or network. ACI's approach with this solution is like a bridge between BASE24 customers' current systems and evolving end-user demands, enabling them to continue running some or all of their systems into the foreseeable future. This strategy lowers risk and costs, adds ACI, eliminating the need to "rip and replace" systems to address emerging payment needs. ACI emphasizes the advantages of the new RHEL version of UP Retail Payments, such as a 50% reduction in TCO while increasing performance, scalability and reliability.
http://aciworldwide.com

EnterpriseDB's EDB Postgres Advanced Server and EDB Postgres Enterprise Manager

The elegance of open source is on full display with new product releases like EnterpriseDB's (EDB's) new PostreSQL-based database solutions. On the heels of the significant PostgreSQL 9.5 update come two EnterpriseDB solutions that take Postgres further, namely EDB Postgres Advanced Server 9.5 and EDB Postgres Enterprise Manager 6.0. EDB Postgres Advanced Server integrates additional capabilities and security into Postgres that large companies and governments require in order to use it, and this new v9.5 features the following: preconfigured integration with Hadoop and MongoDB, enhanced security with password profiles, expanded compatibility with Oracle to ease and speed migrations, and dramatic performance increases through vertical scaling optimizations. Meanwhile, EDB Postgres Enterprise Manager, EDB's single console for tuning, monitoring and administering large Postgres deployments, adds enhancements as well in this v6.0 release. These include Nagios support, failover management, a streaming replication wizard, audit log alerts and an improved alert UI.
http://enterprisedb.com

LinuxFest Northwest

April 23rd & 24th
Bellingham, WA

- All things Open Source
- 40+ Exhibitors
- 80+ Presentations
- 1500+ Attendees
- Prizes and after party
- FREE admission & parking
- Bring the whole family!

Hosted By

Bellingham TECHNICAL COLLEGE

linuxfestnorthwest.org

VARNISH CACHE

Varnish Software's Hitch

Making life easier for the 2.2 million Web sites that deploy the Varnish Cache HTTP engine is the point of Hitch from Varnish Software. The recently updated Hitch is a scalable, open-source network proxy designed to handle tens of thousands of connections on multicore machines efficiently. Maker Varnish describes Hitch's benefits as easy to configure, a low memory footprint and the ideal way of terminating client-side SSL/TLS for Varnish. The deployment process for Varnish Cache is streamlined by the support for the PROXY protocol, which lets Varnish consider the original client's endpoints as if there were no TLS proxy in between. Hitch is tested on Linux, but works on other *nixes as well. Hitch's features include support for TLS1.0-1.2; SNI, with and without wild-card certificates; support for HAProxy's PROXY protocol; seamless configuration run-time reload support and performance of up to 15,000 listening sockets and 500,000 certificates.
http://hitch-tls.org,
http://varnish-software.com

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

RETURN TO CONTENTS

THE

PERL

CONFERENCE

Meet

PERL 6

June 19-24

PerlConference.Org

ROCK-SOLID ENCRYPTED VIDEO STREAMING

Using SSH Tunnels and the BeagleBone Black

Gain a deep understanding of SSH port forwarding by implementing a streaming video server on a BeagleBone Black.

RAMON CRICHLOW
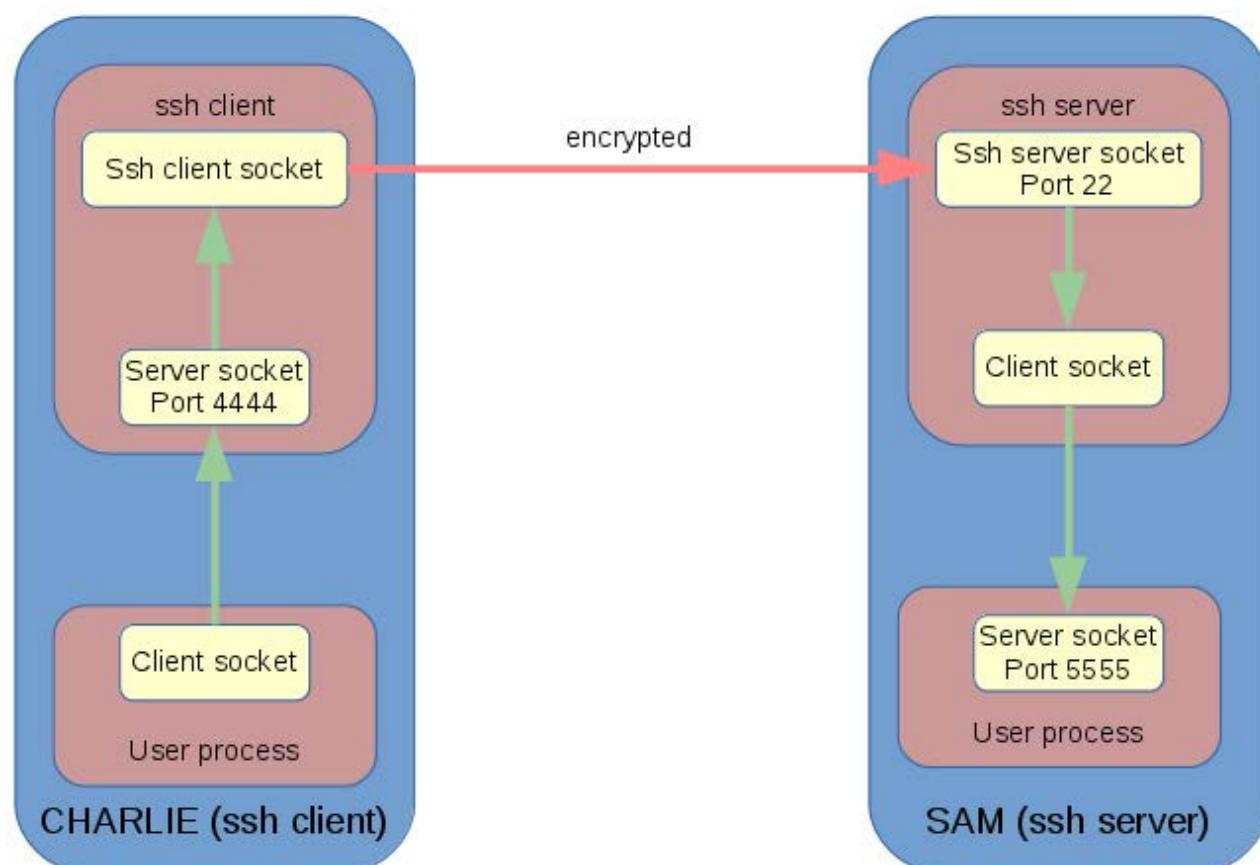
You probably have used SSH as a remote login shell, but you also can use SSH in a number of unexpected but very useful applications. One such use is tunnelling, or port forwarding, which is an effective method of accessing networked hosts located behind routers, firewalls and NAT gateways. As an added benefit, SSH encrypts the data passing through the tunnel, increasing the security of your communications. This article shows you how to set up stable and resilient SSH tunnels that will survive network outages, computer reboots and idle connection timeouts.

Understanding SSH Tunnelling

The SSH man page describes SSH as "a program for logging into a


```
$ ssh -L 4444:127.0.0.1:5555 some_user@sam
```

Figure 1. Data Path for TCP Packets Traveling between Hosts Charlie and Sam

remote machine and for executing commands on a remote machine… X11 connections and arbitrary TCP ports can also be forwarded over the secure channel."

Starting at the socket level, let's take a look at how SSH forwards TCP ports securely. It's easiest to begin with an example.

Figure 1 illustrates the data path for TCP packets traveling between hosts charlie and sam. In this example, the following command was executed on charlie:

```
$ ssh -L 4444:127.0.0.1:5555 some_user@sam
```

This command directs the SSH client on charlie to forward port 4444 to port 5555 on sam. The `-L` flag indicates that port 4444 is on the local (or client) host, which, in this case, is charlie where the `ssh` command was executed.

When this command runs, several actions take place:

■ The SSH program establishes a connection to sam using the default SSH port 22.

■ The SSH process on charlie creates a server socket bound to localhost:4444 and begins listening.

■ When an arbitrary user process creates a client socket and connects to the server socket at 4444, the SSH process encrypts the data, then transmits it over the SSH connection (established in step 1) to sam.

■ The SSH server on sam decrypts the data.

■ The SSH server creates a client socket and writes the data to the host and port specified in the SSH command, which, in this case, is 127.0.0.1:5555. 127.0.0.1 is, of course, the loopback or localhost address.

■ The arbitrary user server process on sam that created the server socket bound to 5555 receives the data.

Understanding SSH Commands

The location of the server socket and client socket is key to understanding how SSH tunnels work. As Figure 1 shows, SSH tunnels always result in the creation of a client and server socket. In a forward tunnel, the local host creates the server socket and the remote host creates the client socket. In a reverse tunnel, the local host creates the client socket and the remote host creates the server socket. The local host is the machine where the SSH command was executed.

The remote host to which the port is forwarded does not have to be the SSH server. Consider the following command executed on charlie:

```
$ ssh -L 1234:www.some_website.com:80 some_user@SAM
```

In this instance, charlie creates a server socket bound to port 1234. If you run a Web browser on charlie and point it to localhost:1234, SSH securely forwards the connection to sam, where the SSH server there creates a client socket and connects to the Web server at some_website.com. This command can be used to connect to a server from a private network on which it is blocked.

Reverse SSH tunnels work exactly the same way, except that the remote host creates the server socket and the local host creates the client socket. The -R option instructs the SSH process to create a reverse tunnel:

```
$ ssh -R 1234:www.some_website.com:80 some_user@SAM
```

This command executed on charlie creates a server socket on sam at port 1234. A browser running on sam and connecting to port 1234 on sam would have its traffic encrypted and routed to charlie, and thence to some_website.com.

One source of confusion when using 127.0.0.1 (or localhost) in port-forwarding commands is determining which host is the localhost from the point of view of the tunnel. One way to resolve where localhost refers is to remember that a client socket at the end of the tunnel is used to connect to the specified host; thus, localhost is the machine that creates the client socket. For forward tunnels, 127.0.0.1 refers to the remote SSH server, and for reverse tunnels, 127.0.0.1 refers to the client machine

where the SSH command executes.

Now, let's look at using SSH tunnelling to stream video from a BeagleBone Black. An Amazon Web server, configured with a public static IP will serve as a relay where the video can be viewed.

Amazon Web Server Configuration

Follow these steps to create and configure an EC2 instance. Amazon has an excellent set of instructions on its Web site documenting each step if you encounter problems.

1. Sign up for an AWS account at https://aws.amazon.com/free. (The AWS Free Tier is free for one year at the time of this writing; however, be aware that if you exceed the bandwidth limit, it will trigger charges to your credit card.)

2. Open the EC2 dashboard by clicking on the "EC2 virtual servers in the cloud" tab and create an EC2 instance. Select the Ubuntu Server 14.04 LTS (HVM) image. Note carefully the location of the downloaded private key file (.pem), as you will need it to `ssh` in to your EC2 instance.

3. Assign a public static IP to your instance using the Elastic IP tab under the EC2 dashboard.

4. Open port 5555 on your instance by creating a new security group.

5. Assign the newly created security group to your instance.

The final step is to configure the keep-alive time for the SSH server. Append the following lines to /etc/ssh/sshd_config (note: it is sshd_config, not ssh_config):

```
# vi /etc/ssh/sshd_config
ClientAliveCountMax 3
ClientAliveInterval 60
```

Then reboot:

```
$ sudo shutdown -r now
```

`ClientAliveInterval` is a timeout interval. If a connection has been idle for `ClientAliveInterval` seconds, the sshd server will send a message through the encrypted channel to request a response from the client.

`ClientAliveCountMax` sets the maximum number of client alive messages, which may be sent without any client response. If this threshold is reached, sshd will disconnect the client, terminating the session.

BeagleBone Black Configuration

Many intermittent BeagleBone Black problems can be resolved by using a 5V 2A DC power supply instead of powering the BeagleBone over USB. Likewise, it is safer to use a powered USB hub when connecting multiple USB devices.

I used a BeagleBone Black Rev C to complete the steps outlined in this article. The default Debian image as shipped at the time of this writing was Debian GNU/Linux 7.4 (wheezy):

```
$ lsb_release -a
No LSB modules are available.
Distributor ID: Debian
Description: Debian GNU/Linux 7.4 (wheezy)
Release: 7.4
Codename: wheezy
```

The kernel was upgraded to 3.8.13-bone79 using the built-in script:

```
$ cd /opt/scripts/tools/
$ git pull
$ sudo ./update_kernel.sh
$ sudo reboot

$ uname -r
3.8.13-bone79
```

I used a Logitech HD Webcam C270 for the USB Webcam.

Installing mjpg-streamer

To complete these steps, ensure that your BeagleBone has a live Internet connection. Log in to your BeagleBone and type:

```
$ cd
$ wget https://github.com/shrkey/mjpg-streamer/raw/master/
➡mjpg-streamer.tar.gz
$ tar -xvf ./mjpg-streamer.tar.gz
$ cd mjpg-streamer
$ make
```

Note that in the following scripts and commands you must replace `my_beagle` with its IP address.

Plug in your Webcam. You can use `lsusb` to check that it has enumerated properly:

```
$ lsusb
Bus 001 Device 003: ID 046d:0825 Logitech, Inc. Webcam C270
```

If the Webcam doesn't enumerate, leave it connected and reboot the BeagleBone.

Start mjpg_streamer:

```
$ ./mjpg_streamer -i "./input_uvc.so" -o "./output_http.so
➡-w ./www -p 8090"
```

Check whether the BeagleBone is streaming video by opening a browser in a host on the same network as the BeagleBone and browse to http://my_beagle:8090/?action=stream.

Now, create a script to start mjpg_streamer:

```
$ cd
$ mkdir scripts
$ cd scripts
```

```
$ vi mjpg_streamer.sh
#!/bin/bash
if ! pgrep -f /home/debian/mjpg-streamer/mjpg_streamer
then /home/debian/mjpg-streamer/mjpg_streamer -i "/home/debian/
➥mjpg-streamer/input_uvc.so --resolution QVGA " -o
 ➥"/home/debian/mjpg-streamer/output_http.so -w
 ➥/home/debian/mjpg-streamer/www -p 8090"
fi
$ chmod +x mjpg_streamer.sh
```

Create a cronjob that checks whether mjpg_streamer is running once per minute:

```
$ crontab -e
* * * * * /home/debian/scripts/mjpg_streamer.sh
```

At this point, the video server is running on port 8090 with QVGA resolution and will restart automatically when the BeagleBone is rebooted.

SSH Setup

To ssh in to your EC2 instance, you will need to copy the private key to the BeagleBone. The private key is the .pem file that you downloaded when configuring the EC2 instance. You can copy the private key to the BeagleBone using scp (which coincidentally also uses the SSH protocol) to transfer files between hosts. So from the computer where you downloaded the private key, type:

```
$ scp my_private_key.pem debian@my_beagle: ~/
```

Now log in to the BeagleBone and type:

```
$ chmod 0400 my_private_key.pem
```

Next, look up the public IP of my_amazon. The public IP for my_amazon can be found in the Elastic IP tabs in the AWS console.

Test that you can ssh in to your EC2 instance by typing:

```
$ ssh ubuntu@my_amazon -i my_private_key.pem
```

To make things easier, let's set up an alias to store the EC2 information. Append these lines to the end of /etc/ssh/ssh_config:

```
$ sudo vi  /etc/ssh/ssh_config
```

```
Host ec2
        HostName my_amazon
        IdentityFile  /home/debian/my_private_key.pem
        User ubuntu
        ServerAliveInterval 60
        ServerAliveCountMax 2
```

Replace my_amazon with its public IP address.

The `ServerAliveInterval` and `ServerAliveCountMax` are the client-side analogs to `ClientAliveInterval` and `ClientAliveCountMax`. They function in exactly the same way.

Now typing the following should log you in to your EC2 instance:

```
$ ssh ec2
```

autossh Configuration

Many how-tos explain how to use SSH commands for port forwarding but fail to describe how to keep the SSH tunnel alive. One common challenge in keeping network connections up is managing the router or firewall inactive connection timeout. Routers often will close a connection if it is idle for a certain period, sometimes closing a connection if it is idle for as few as five minutes. Server reboots and network outages also obviously can kill the tunnel.

Fortunately, there is a ready-made utility that monitors SSH connections and restarts them when they die: autossh. The `ServerAlive` configuration flags you added to send a packet periodically through the tunnel will defeat the inactive connection timeout.

Install autossh on my_beagle by typing:

```
$ sudo apt-get install autossh
```

Now, let's write a startup script that launches autossh at boot time using the `init.d` facilities. Create a script named tunnel in /etc/init.d:

```
$ cd /etc/init.d
$ sudo vi /etc/init.d/tunnel

#!/bin/bash
### BEGIN INIT INFO
# Provides:          tunnel
# Required-Start:    $local_fs $remote_fs $network $syslog $named
# Required-Stop:     $local_fs $remote_fs $network $syslog $named
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start/stop  ssh tunnel
### END INIT INFO

set -e #abort at first error
LOGFILE=/home/debian/my_autossh.log
export AUTOSSH_GATETIME=0  #Prevents autossh from exiting if first
                           #attempt  fails
export AUTOSSH_PIDFILE=/home/debian/my_autossh.pid

case $1 in
 start)
  su -c "date >>$LOGFILE; autossh -M 0 -N -R 5555:localhost:8090
➥ec2 &>>$LOGFILE" -s /bin/bash debian
 ;;
 stop)
  if [ -e $AUTOSSH_PIDFILE ]
   then
   kill -9 `cat $AUTOSSH_PIDFILE`
  fi
 ;;
 *)
  echo "Usage: /etc/init.d/tunnel {start|stop|}"
  exit 1
```

```
  ;;
esac
```

```
$ sudo chmod +x tunnel
$ sudo update-rc.d tunnel defaults
$ sudo shutdown -r now
```

PC Setup

Since this is a video streaming application, it is likely that you want to view the video stream on different computers. This can be achieved by allowing the sshd server on your AWS server to forward connections from any host (and by setting a password on mjpg_streamer for security). I'll explain how to do that shortly; however, this exposes mjpg_streamer to brute-force password attacks. The intruder would only be able to view the video stream, but that may not be desirable.

One alternative is to create a forward tunnel and stream the encrypted video over this tunnel. This requires installing the private key on whatever machine you are viewing from, as well as setting up the tunnel.

Secure Video Streaming

To reduce typing, create an alias for your EC2 as you did on my_beagle:

```
$ sudo apt-get install autossh
$ vi ~/.ssh/config
Host ec2
        HostName my_amazon
        IdentityFile  path_to/my_private_key.pem
        User ubuntu
        ServerAliveInterval 60
        ServerAliveCountMax 2
```

The command to create a forward tunnel to my_amazon is the following:

```
$ ssh -L 1234:localhost:5555 ec2
```

The choice of the local server port (1234) is arbitrary; it can be any port.

Port numbers under 1024 will require root access, however.

Once this command is executed, start a browser and open http://127.0.0.1:1234/?action=stream.

You should see the live video stream from the Webcam on my_beagle.

You can make this tunnel permanent by adding a startup script following the same model as before:

```
$ cd /etc/init.d
$ sudo vi /etc/init.d/forwardtunnel
#!/bin/bash
### BEGIN INIT INFO
# Provides:          forwardtunnel
# Required-Start:    $local_fs $remote_fs $network $syslog $named
# Required-Stop:     $local_fs $remote_fs $network $syslog $named
# Default-Start:     2 3 4 5
# Default-Stop:      0 1 6
# Short-Description: Start/stop  ssh forwardtunnel
### END INIT INFO

set -e #abort at first error
LOGFILE=/home/your_username/my_autossh.log  # CHANGE YOUR_USERNAME!!
export AUTOSSH_GATETIME=0 #Prevents autossh from exiting if
 ➥first attempt fails
export AUTOSSH_PIDFILE=/home/your_username/my_autossh.pid # CHANGE
 ➥YOUR_USERNAME!!

case $1 in
 start)
  su -c "date >>$LOGFILE; autossh -M 0 -N -L 1234:localhost:5555
➥ec2 &>>$LOGFILE" -s /bin/bash your_username# CHANGE YOUR_USERNAME!!
 ;;
 stop)
  if [ -e $AUTOSSH_PIDFILE ]
   then
   kill -9 `cat $AUTOSSH_PIDFILE`
  fi
```

```
  ;;
  *)
    echo "Usage: /etc/init.d/forwardtunnel {start|stop|}"
    exit 1
  ;;
esac

$ sudo chmod +x forwardtunnel
$ sudo update-rc.d forwardtunnel defaults
$ sudo shutdown -r now
```

Note that you will have to change `your_username` in the above to your user name.

Unsecured Video Streaming

Unsecured video streaming permits anyone with the correct user name and password to connect to the mjpg_streamer server and view the video stream. Although mjpg_streamer has password protection, it is vulnerable to brute-force attacks. However, the convenience of viewing the video stream from any platform without the private key and without setting up a forward tunnel may be worth it.

First, let's add some security by requiring anyone viewing the Webcam to authenticate with the correct user name and password by using the authentication option to mjpg_streamer.

On the BeagleBone:

```
$ vi /home/debian/scripts/mjpg_streamer.sh
#!/bin/bash
if ! pgrep -f /home/debian/mjpg-streamer/mjpg_streamer
then /home/debian/mjpg-streamer/mjpg_streamer -i "/home/debian/
➥mjpg-streamer/input_uvc.so" -o "/home/debian/mjpg-streamer/
➥output_http.so -w /home/debian/mjpg-streamer/www -p 8090
 ➥--resolution QVGA -c userNamE:Passw0rd"
fi
$ shutdown -r now
```

Change `userNamE:Passw0rd` to one of your own invention. The user name is unrelated to any account on your BeagleBone.

The second change is to set the `GatewayPorts` flag to yes on the Amazon Web server:

```
# vi /etc/ssh/sshd_config
ClientAliveCountMax 3
ClientAliveInterval 60
GatewayPorts yes
# service ssh restart
```

Once the SSH tunnels are re-established, browsing to my_amazon:5555/?action=stream will open the video stream after entering the correct user name and password. The `GatewayPorts` flag controls whether the sshd server forwards connections from external hosts (`GatewayPorts yes`) or only from localhost (`GatewayPorts no`).

Conclusion

In this article, I have described how to set up secure and reliable SSH tunnels, explained the underlying mechanism behind SSH tunnelling and implemented a secure video streaming server accessible from external networks without requiring any router or firewall changes. Another configuration you might explore is a tunnel to an sshd server on your BeagleBone. With this tunnel, you could log in remotely from any external network. Happy tunnelling!■

Ramon v3.0 is an embedded software AI with a keen interest in real-time video streaming. Having outgrown several Saskatchewan-based hard drives, he now resides diffusely in the Google cloud.

RETURN TO CONTENTS

Stunnel Security for Oracle

Replace database TLS for simplified best-practice compliance.

CHARLES FISHER

Oracle has integrated modern Transport Layer Security (TLS) network encryption into its eponymous database product, and TLS usage no longer requires the Advanced Security option beginning with the 10.2 database release. Legacy configurations lacking TLS exchange encrypted passwords, but the session payload is transmitted in clear text and is intercepted easily by anyone with control over the intermediate network. Databases holding sensitive content should avoid clear-text traffic configurations.

It is possible to use the stunnel utility to wrap the Oracle Transparent Network Substrate (TNS) Listener "invisibly" with TLS encryption as an isolated process, and this configuration appears to be compatible both with Oracle's sqlplus command-line utility and with database links that are used for distributed transactions between multiple database instances. There are several benefits to stunnel over the TNS Listener's native TLS implementation:

■ The stunnel utility can be far less expensive. Older Oracle database releases required the Advanced Security option to use TLS, which is licensed at $15,000 per CPU according to the latest pricing (http://www.oracle.com/us/corporate/pricing/technology-price-list-070617.pdf), but TLS is now included with Standard Edition SE2 (https://oracle-base.com/articles/misc/configure-tcpip-with-ssl-and-tls-for-database-connections).

■ The stunnel utility and the associated dependent libraries (that is, OpenSSL) are patched far more often, and updates can be applied immediately with no database "bounce" if stunnel is used in an "inetd" configuration. Oracle issued eight total patched versions of OpenSSL in 2015 for Oracle Linux 7 (https://oss.oracle.com/ol7/SRPMS-updates). Database patches are issued only four times per year at regular quarterly intervals and require instance bounces/outages. An urgent SSL/TLS update will have lengthy delays when implemented as a database patch (due in part to an overabundance

> For this reason, security-sensitive code that may require immediate updates should be kept out of the database server whenever possible. The stunnel utility meets this requirement very well.

of caution by most DBAs), but will be far easier to apply as a simple OS utility patch with no downtime. For this reason, security-sensitive code that may require immediate updates should be kept out of the database server whenever possible. The stunnel utility meets this requirement very well.

- The stunnel utility can run as a separate user and group inside a "chroot jail" that has limited visibility to the rest of the system. Oracle's server TLS implementation runs with the full privilege of the TNS Listener. A compromise of the TLS engine can be drastically less dangerous if it is confined within a chroot() jail. Privilege separation and chroot() are well-recognized security techniques, and many security-sensitive installations likely will disable listener TLS for this reason alone.

Let's proceed with adding stunnel TLS services to Oracle.

Server Configuration

I am assuming that the reader is familiar with Oracle databases and the procedures to start up an instance and the TLS Listener. For reference, let's assume that a database SID "mydb" is running, and an example listener dæmon is launched on the IP address 1.2.3.4 with the following commands:

```
export ORACLE_SID=mydb ORACLE_HOME=~oracle/Ora12c/db
```

```
$ORACLE_HOME/bin/lsnrctl start
```

The listener will generate a startup message similar to the output below:

```
LSNRCTL for Linux: Version 12.1.0.2.0 - Production on
➥19-FEB-2016 13:18:55

Copyright (c) 1991, 2014, Oracle.  All rights reserved.

Starting /home/oracle/Ora12c/db/bin/tnslsnr: please wait...

TNSLSNR for Linux: Version 12.1.0.2.0 - Production
System parameter file is /home/oracle/Ora12c/db/network/
➥admin/listener.ora
Log messages written to /home/oracle/Ora12c/diag/tnslsnr/
➥HOSTNAME/listener/alert/log.xml
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=1.2.3.4)
➥(PORT=1521)))
Listening on: (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC)))

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=TCP)(HOST=1.2.3.4)
➥(PORT=1521)))
STATUS of the LISTENER
------------------------
Alias              LISTENER
Version            TNSLSNR for Linux: Version 12.1.0.2.0 - Production
Start Date         19-FEB-2016 13:18:55
Uptime             0 days 0 hr. 0 min. 0 sec
Trace Level        off
Security           ON: Local OS Authentication
SNMP               OFF
Parameter File     /home/oracle/Ora12c/db/network/admin/listener.ora
Listener Log File  /home/oracle/Ora12c/diag/tnslsnr/HOSTNAME/listener/
➥alert/log.xml
Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=1.2.3.4)(PORT=1521)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC)))
Services Summary...
```

```
Service "mydb" has 1 instance(s).
  Instance "mydb", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully
```

It is important that the listener not engage in "port redirection" of clients to separate server ports (most commonly seen in MTS/Shared Server). Any feature causing the TNS Listener to engage in such activity must be disabled.

To configure stunnel, the root user must create a keypair for TLS. This keypair can be "signed" by a Certificate Authority (CA) if desired—this is conventionally useful for Web site encryption (HTTPS) since the lack of a recognized CA signature will trigger browser security warnings. Oracle clients can verify server keys only when signed by a recognized CA, which is addressed in the final section of this article. To obtained signed keys, follow the instructions on the stunnel Web site (https://www.stunnel.org/howto.html). Otherwise, for more informal use, a self-signed key can be generated with the following commands:

```
cd /etc/pki/tls/certs
make stunnel.pem
```

The process of key generation will ask a number of questions:

```
Generating a 2048 bit RSA private key
.......................................................................+++
...............................+++
writing new private key to '/tmp/openssl.hXP3gW'
-----
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) [XX]:US
State or Province Name (full name) []:IL
Locality Name (eg, city) [Default City]:Chicago
Organization Name (eg, company) [Default Company Ltd]:ACME Corporation
Organizational Unit Name (eg, section) []:Widget Division
Common Name (eg, your name or your server's hostname) []:darkstar
Email Address []:linus@posix.org
```

The key produced above will be set for expiration in 365 days from the day it was created. If you would like to generate a key with a longer life, you can call OpenSSL directly:

```
openssl req -new -x509 -days 3650 -nodes \
     -out stunnel.pem -keyout stunnel.pem
```

The key will look something like this:

```
# cat /etc/pki/tls/certs/stunnel.pem
-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQC23m+w0BLxI2zB
/p8/TiuFcEurTLbLCQwcO/FE+vNcJpddckuF6/VgpBAJk+d9i7NZNqrjMH711H18
3AYhewZTCbRUMQE3ndaYEIxSt4Qhbm8XbfUfx6Fmg4CnWh/XzE7B8Z7XbHpwRQ4d
kQOzICzb1nt96QKdWoAob73+hv7qdi3UjJ3/20z3Cx5LWfWoa32Y50//tvBjBtcQ
H7QpiE2tfLWHTQ5tztkqVY/MZJWVgoT5LnqQlZeZB/C4izSYNo9EGAnw4ThaFJ/y
NdvmyK6sYaO3Dq4eFO78O+zzqyfhPCtcfb8lMuRTZa8uiv7ziVf0A3eGSwKYonUf
iL7su0kJAgMBAAECggEASyeDk5EQF9ZNPjUc0XGY5VBPaOkwPqVLOtdPwt+34Glj
z93HOBTPVZZXmPgWLTyaytFyzcgChZl8sTHjuyLKaJoWaHtzWp4dsYUrhlsxjGPM
eD6SfSsYI/9rglvBtnia7Y4Vj8dfUoCu2mvcr2NLzFWLjyWSE4U8ImI6HT7xyP1y
5Ab8JYX5C0qODTjjPldovz8Fm9XE7yKVg5XQ9aA8axq1uWY7ruzxXli/+h7TJLPe
v/yCCeKLqL+gSkORjhg8ZsK/9p6InMN6uFUeSvb0Y9eLnLnJ6tCUGh4m+cgXulrD
UleXxeFzxnS3ydCNHVDngXyaJ1U+bLGVeHOLRcZHAQKBgQDY5jU+rU8VPXrbPJae
fZ8swf3Pi00QUKN4P5Zyy5cs18KMgDRHRaUdYmVCpTISR7Wi0XHEI7iHDpKlL//k
zCT0LW+fk+4A7bP5yUhLLtd175RUpR3ieBuXbAZRqDaPMDJ1xpZj+hDiShPv+OGR
k7ETnBDm/zk1T+F2Lu6qebLbcQKBgQDX1b6I7eRDkyFZITZX1v+S6PkoTMQMeCGX
cSwVvuDuiszgbf7IImWvvFxd7h+WJZEVv4jV645LOsvkY6XyFXdW7iNJGkKgThg6
```

YNE3X5f1oGvo5E3+HNXS3vGs76YVKTraDd0SKIRT98m2jiXCVCw+KWlR5GR+xp2A
8exCoTYLGQKBgQDJdcmu1brGt7wNNlGQFI5sPCNLSs/hf4TWg/lx1rgr5pvFdK8a
JA4hJOt444eGgySqfm91Bti2WUrMM7EzCoqoYitzxSsjoaWxNMv5SSDHYigcFuGT
IIxAMQ4NenhytwmnazT016gnBzdNhZW+abfnxuXMKPMyGWgJJb54iWEfgQKBgQDV
N3xgfNHwx5o8GIk8wVH86VWqMBvETbCxkMWCPeyq+kdmtoLpZsGZl7SPvjtJ8paf
K3WcDnWlb9IYLzCyM+6O2/XTs7N59WwNz7Mexrqxleb ETTWXARlilYed1aj2YqKW
4vcvhwMiiDimtUor7Uc/qV033y4/5ymVRmilceiXkQKBgEbFhAKXP9qZMdfLevWp
VXAZCc5mQ2hxQOSQRAL5VvTUXm6ZwVXVf/U42JH3YXiVXbwDbEjjxS/8MtSbQU9z
LoVQ/+bvc3xQ08u8kxdQiWzTzwRxHJM/znxpoD9astItq4uWU58hCoUNItHEKGJt
60bczdu3rZLZIB1n2zSM6soF
-----END PRIVATE KEY-----
-----BEGIN CERTIFICATE-----
MIID/TCCAuWgAwIBAgIJALT/9skCvdR5MA0GCSqGSIb3DQEBCwUAMIGUMQswCQYD
VQQGEwJVUzELMAkGA1UECAwCSUwxEDAOBgNVBAcMB0NoaWNhZ28xGTAXBgNVBAoM
EEFDTUUgQ29ycG9yYXRpb24xGDAWBgNVBAsMD1dpZGdldCBEaXZpc2lvbjERMA8G
A1UEAwwIZGFya3N0YXIxHjAcBgkqhkiG9w0BCQEWD2xpbnVzQHBvc2l4Lm9yZzZAe
Fw0xNTEwMzAwMzI2NTJaFw0yNTEwMjcwMzI2NTJaMIGUMQswCQYDVQQGEwJVUzEL
MAkGA1UECAwCSUwxEDAOBgNVBAcMB0NoaWNhZ28xGTAXBgNVBAoMEEFDTUUgQ29y
cG9yYXRpb24xGDAWBgNVBAsMD1dpZGdldCBEaXZpc2lvbjERMA8GA1UEAwwIZGFy
a3N0YXIxHjAcBgkqhkiG9w0BCQEWD2xpbnVzQHBvc2l4Lm9yZzCCASIwDQYJKoZI
hvcNAQEBBQADggEPADCCAQoCggEBALbeb7DQEvEjbMH+nz9OK4VwS6tMtssJDBw7
8UT681wml11yS4Xr9WCkEAmT532Ls1k2quMwfvXUfXzcBiF7BlMJtFQxATed1pgQ
jFK3hCFubxdt9R/HoWaDgKdaH9fMTsHxntdsenBFDh2RA7MgLNvWe33pAp1agChv
vf6G/up2LdSMnf/bTPcLHktZ9ahrfZjnT/+28GMG1xAftCmITa18tYdNDm3O2SpV
j8xklZWChPkuepCVl5kH8LiLNJg2j0QYCfDhOFoUn/I12+bIrqxho7cOrh4U7vw7
7POrJ+E8K1x9vyUy5FNlry6K/vOJV/QDd4ZLApiidR+Ivuy7SQkCAwEAAaNQME4w
HQYDVR0OBBYEFEgCdENJ+1y0STMyQtuz3uqDON3NMB8GA1UdIwQYMBaAFEgCdENJ
+1y0STMyQtuz3uqDON3NMAwGA1UdEwQFMAMBAf8wDQYJKoZIhvcNAQELBQADggEB
AITW7cBNdWBcgagsinGKfyESBlJ7JvxvsMzYVhI8myC8ht/3nFMyTHmBgtmxdNbW
mCCzDdeSigQf/iEzH02k/EK7L7I3DATGBW6w9WiYBdqrZJl98CIxoY9j+GV0AeL1
INMSb5G4R2ygnekXVNTJsICeVHTRujBJsD4psZB5dhSI888rA2MrdQ8jAFGDk7Z4
VYckA2gQ+70yXXxpFSD4n2ecq3ebNtej07zR2wAtAkt/JtuGiUjbl1m4ZFTPoTwr
xDYMcezEgopMzYMihv6CQ0CEU+qL+92CYtEDsd1hzn74SlBK9HMKjMLrbBZPhbE4
/JMRW5oa/+TFZIRcacTxgAw=
-----END CERTIFICATE-----

The PRIVATE KEY section above is the most sensitive portion of the file; ensure that it is not seen or copied by anyone that you do not trust, and any recordings on backup media should be encrypted. The BEGIN CERTIFICATE section is presented to TLS clients (that is, sqlplus) when they connect to stunnel.

It is likely wise to compute custom primes for the Diffie-Hellman key exchange algorithm, following guidance from the stunnel manual page:

```
openssl dhparam 2048 >> stunnel.pem
```

The previous command will add another section to your stunnel.pem file for high-security Diffie-Hellman primes:

```
-----BEGIN DH PARAMETERS-----
MIIBCAKCAQEAoHi5jzY5ZVwGCFFm1EhVsePXxNwCSs/eQbaC3rc+iXENL8xk21uq
6eSwYIQWUeDN/h6wBBDe6dpFoNDJQeqKCmUa8aojGHnkcqsJBdVUKVF5/7rWb1Yi
TzvbeZt8UvYnNUErJEpgBMiKPDYipE2BZ6k61WwkK6WV6svGAHpIc3o/9kU+72uf
dPFaNIygAb2HLaJYvXq9OYGvrMsmyZTh3fnpg2RiZSVJf+i4BfyeLiYkwnSZozAS
2rQ4hf2E5WY6jiAcNZBLKvqR8lUuIaXd9+VkiCSV0c2pXzb2ElxOk8sheAHliwip
SaKC694z9l63eNKQW2J4WI97wkil0qa4MwIBAg==
-----END DH PARAMETERS-----
```

The Oracle TNS Listener conventionally runs at port 1521. In this exercise, let's run Oracle TLS services at port 1522, which has the current service name:

```
# grep 1522 /etc/services
ricardo-lm    1522/tcp       # Ricardo North America License
                             # Manager
ricardo-lm    1522/udp       # Ricardo North America License
                             # Manager
```

Place the following file to control stunnel for the "ricardo" service (alter the IP address 1.2.3.4 to the location of your TNS Listener):

```
# cat /etc/stunnel/ricardo.conf
sslVersion =    TLSv1.2
     options   =    NO_SSLv3
     options   =    NO_SSLv2
     options   =    SINGLE_DH_USE
     options   =    SINGLE_ECDH_USE
     options   =    CIPHER_SERVER_PREFERENCE
     cert =    /etc/pki/tls/certs/stunnel.pem
     FIPS =    no
     debug=    6
     syslog    =    yes
     chroot    =    /var/empty
     setuid    =    nobody
     setgid    =    nobody
     connect   =    1.2.3.4:1521


; best-practice ciphers:
; https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers/
ciphers=ECDH+AESGCM:DH+AESGCM:ECDH+AES256:DH+AES256:ECDH+AES128:
➥DH+AES:ECDH+3DES:DH+3DES:RSA+AESGCM:RSA+AES:RSA+3DES:
➥!aNULL:!MD5:!DSS
```

Note above that you are configuring TLS for best-practice encryption with the highest quality protocols and ciphers (https://www.rfc-editor.org/rfc/rfc7525.txt). The Oracle clients appear compatible with these settings. Note that Michal Trojnara, the author of stunnel, does "not recommend using DH ciphersuites in the hardened set. ECDH ciphersuites are much more secure and much faster - RFC 7525 should be considered outdated after the recent attacks on DH." On the other hand, there have been recent questions of software patents on Elliptic Curve (http://security.stackexchange.com/questions/3519/can-ecc-be-used-without-infringing-on-patents), although Sun/Oracle contributed the ECC implementation in OpenSSL and used great care to avoid patented methods. Red Hat/Fedora went further in enabling only the Suite B subset of NIST ECC curves for protection from Certicom (whether this is a sufficient courtroom defense against CryptoPeak is another

matter: http://www.theregister.co.uk/2015/12/01/ cryptopeak_sues_). Beyond that, in my previous coverage of the Stribika SSH Guide [see "Cipher Security" by Charles Fisher, September 2015], I wrote that the author is "…advising against the use of NIST elliptic curves because they are notoriously hard to implement correctly. So much so, that I wonder if it's intentional. Any simple implementation will seem to work but leak secrets through side channels. Disabling them doesn't seem to cause a problem; clients either have Curve25519 too, or they have good enough DH support." Trojnara has responded that the question of "side-channel attacks on ECDHE is pure nonsense, since by definition (the last 'E' stands for 'ephemeral'), there is no persistent secret here an attacker might retrieve with [any available] side-channel attacks." In any case, Hynek Schlawack's Web site on the subject has not endorsed one over the other so far, while his silence on the growing questions behind Diffie-Hellman key exchange is somewhat unsettling (https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers). Your legal environment and encryption stance will decide your cipher string.

Use the following systemd unit files to configure stunnel for inetd-style operation (if you aren't using an OS based on systemd, see my previous articles for a discussion of [x]inetd):

```
# cat /etc/systemd/system/ricardo.socket
[Unit]
Description=oracle stunnel
[Socket]
ListenStream=1522
Accept=yes
[Install]
WantedBy=sockets.target
```

```
# cat /etc/systemd/system/ricardo@.service
[Unit]
Description=oracle stunnel service
[Service]
ExecStart=-/usr/bin/stunnel /etc/stunnel/ricardo.conf
StandardInput=socket
```

Assuming that the above unit files are in place, connections on 1522 can be enabled both at boot and for the present environment with these commands:

```
systemctl start ricardo.socket

systemctl enable ricardo.socket
```

The enable command will place systemd's startup link:

```
Created symlink from
/etc/systemd/system/sockets.target.wants/ricardo.socket to
/etc/systemd/system/ricardo.socket.
```

It might be useful to telnet to port 1522, as stunnel will print informative error messages to standard output in case of trouble. The most practical telnet client is likely BusyBox (https://busybox.net/downloads/binaries/latest).

Remote connections to port 1522 might be blocked by your Linux firewall. The root user can permit them to pass to stunnel with the following:

```
iptables -I INPUT -p tcp --dport 1522 --syn -j ACCEPT
```

The TNS Listener can be instructed to restrict the origin of sessions, and it can be used to ban clear-text traffic completely by adding your IP equivalent to the following fragment of the

> ## It might be useful to telnet to port 1522, as stunnel will print informative error messages to standard output in case of trouble.

$ORACLE_HOME/network/admin/sqlnet.ora file on the server:

```
TCP.INVITED_NODES=(127.0.0.1,1.2.3.4)
TCP.VALIDNODE_CHECKING=yes
```

Perform this modification after all testing is successful, and note that any configured clients using the TNS Listener will be shut down if and when the configuration is thus restricted.

It is likely wise to use a stunnel binary provided by Oracle Corporation, but the versions that it provides are rather old. If you can load stunnel version 5, you can omit the `NO_SSL` options shown above. However, the Oracle version 4 stunnel binaries are somewhat more likely to be tolerated in a critical support situation involving Oracle. On the other hand, commercial support from stunnel.org definitely prefers version 5 (https://www.stunnel.org/index.html). If support is an important factor, the experience and availability of the use of both versions will be helpful.

Special thanks to Michal Trojnara, the author of stunnel, for his helpful comments on this article and work in stunnel development. Commercial support, licensing and consulting for stunnel is available from his organization; please visit http://www.stunnel.org/support.html for his latest release.

Database Client

Using the sqlplus client utility that is bundled with a local database server, a TLS session can be established through the stunnel that was previously configured on the remote server. Doing so requires a new client key that is stored in a "wallet", which is created below.

Use the following commands to configure the local sqlplus:

```
export ORACLE_SID=yourdb ORACLE_HOME=/home/oracle/Ora12c/db
mkdir /home/oracle/wallet
$ORACLE_HOME/bin/orapki wallet create -wallet /home/oracle/wallet \
      -pwd SECRET123 -auto_login_local
$ORACLE_HOME/bin/orapki wallet add -wallet /home/oracle/wallet \
      -pwd SECRET123 -dn "CN=%yourdb%" -keysize 2048 \
      -self_signed -validity 3650
```

The output of both calls to the orapki utility above should be this banner:

```
Oracle PKI Tool : Version 12.1.0.2
Copyright (c) 2004, 2014, Oracle and/or its affiliates. All
rights reserved.
```

Directives also must be placed to find the new wallet repository—add the following to your sqlnet.ora file:

```
$ cat $ORACLE_HOME/network/admin/sqlnet.ora

WALLET_LOCATION =
    (SOURCE =
      (METHOD = FILE)
      (METHOD_DATA =
        (DIRECTORY = /home/oracle/wallet)
      )
    )

SSL_CLIENT_AUTHENTICATION = FALSE
```

Finally, call sqlplus with a database account and a connect descriptor that invokes the TLS port at 1522 (note that the newlines within the single quotes are optional and are included here for clarity):

```
$ORACLE_HOME/bin/sqlplus RemoteUser@'(description=
(address=
 (protocol=tcps)
```

```
  (host=1.2.3.4)
  (port=1522)
)
(connect_data=(sid=mydb)))'
```

Assuming success, enter the password for your RemoteUser account, then issue an SQL command:

```
SQL*Plus: Release 12.1.0.2.0 Production on Fri Feb 19 13:26:56 2016

Copyright (c) 1982, 2014, Oracle.  All rights reserved.

Enter password:
Last Successful login time: Fri Feb 19 2016 13:15:54 -06:00

Connected to:
Oracle Database 12c Enterprise Edition Release 12.1.0.2.0 - 64bit
Production
With the Partitioning, OLAP, Advanced Analytics and Real
Application Testing options

SQL> SELECT COUNT(*) FROM DBA_OBJECTS;

  COUNT(*)
----------
     19633
```

A few points to consider:

- Changing `protocol=tcps` to `protocol=tcp` and further modifying `port=1521` above will log in with a clear-text session (if your firewall and listener allow access).

- The `host=` clause above can reference a DNS hostname instead of an IP address if that is more convenient.

- The `TWO_TASK` environment variable can be set with the contents within the single quotation marks above. If this is done, then sqlplus will connect silently to the remote server as if it was local.

- The connect descriptor definition within the single quotation marks above would likely be moved into your TNSNAMES.ORA or network TNS resolution method (ldap, onames).

- The wallet is not required on the server—this functionality is handled by stunnel. The Oracle client needs the wallet if the client's TLS implementation will be used. It is possible to configure stunnel in client mode, then dispense with wallets on both sides.

- While the sqlplus session is active, a stunnel process will appear on the server (be cautious of NPROC or other kernel limits):

```
# ps -ef | grep stunnel

nobody    16810     1  0 13:26 ?          00:00:00 /usr/bin/
stunnel
➥/etc/stunnel/ricardo.conf
```

Database Link

With two or more Oracle database servers, sessions and transactions can be initiated between them to gather and modify data in "two-phase commits". Linkages between accounts and servers are established with the command below (if you have moved tcps hosts into your TNSNAMES.ORA, you can reference them here also):

```
SQL> CREATE DATABASE LINK MyDBLink
 CONNECT TO RemoteUser
 IDENTIFIED BY PassWord
 USING '(description=
  (address=
   (protocol=tcps)
   (host=1.2.3.4)
```

```
   (port=1522)
   )
   (connect_data=(sid=mydb)))';
```

```
Database link created.
```

Once the link is established, remote tables can be suffixed by the link name (which can be joined to other local or remote tables):

```
SQL> SELECT COUNT(*) FROM ALL_OBJECTS@MyDBLink;
```

```
  COUNT(*)
----------
      1851
```

Server Verification

It may be necessary for keys to be verified on either side of the connection to assure authorized use. The native Oracle TLS implementation requires all keys subject to verification to be signed by a recognized CA (the CA's public keys may need to be added to the certificate store used by Oracle).

Note that stunnel also can verify keys and act as a client as well as a server. The stunnel verification options are much more flexible than Oracle's, and if CA signatures are not desired but TLS verification is mandated, then Oracle's TLS should be disabled entirely.

In the examples below, let's assume that the server's public key has a CA signature. To extract that public key, the following awk pattern is useful:

```
awk '/BEGIN CERTIFICATE/,/END CERTIFICATE/' \
     /etc/pki/tls/certs/stunnel.pem > /tmp/pkey
```

Move the /tmp/pkey file to the client, then load it into the wallet:

```
$ORACLE_HOME/bin/orapki wallet add -wallet /home/oracle/wallet \
     -pwd SECRET123 -trusted_cert -cert /tmp/pkey
```

After loading the key, verify that it is now present in the wallet:

```
$ORACLE_HOME/bin/orapki wallet display -wallet /home/oracle/wallet \
     -pwd SECRET123
```

The key should appear in the Trusted Certificates section:

```
Oracle PKI Tool : Version 12.1.0.2
Copyright (c) 2004, 2014, Oracle and/or its affiliates. All
rights reserved.

Requested Certificates:
User Certificates:
Subject:        CN=%yourdb%
Trusted Certificates:
Subject: EmailAddress=linus@posix.org,CN=1.2.3.4,OU=Widget
 ➥Division,O=ACME Corporation,L=Chicago,ST=IL,C=US
Subject:        CN=%yourdb%
```

The client can verify the server keys with the `SSL_SERVER_CERT_DN` clause in the TNS descriptor:

```
$ORACLE_HOME/bin/sqlplus fishecj@'(description=
(address=
 (protocol=tcps)
 (host=1.2.3.4)
 (port=1522)
)
(connect_data=
 (sid=mydb)
 (security=(SSL_SERVER_CERT_DN="CN=1.2.3.4,OU=Widget Division,
➥O=ACME Corporation,L=Chicago,ST=IL,C=US")
)))'
```

If the CA signature is not recognized, the sqlplus login will fail with the following:

```
ERROR:
ORA-29024: Certificate validation failure
```

Additionally, stunnel will record the following in /var/log/secure:

```
LOG7: SSL alert (read): fatal: unknown CA
LOG3: SSL_accept: 14094418: error:14094418:SSL
routines:SSL3_READ_BYTES:tlsv1 alert unknown ca
```

Such errors indicate that the CA is not properly loaded into the bundle used by the database.

Conclusions

Oracle database security has received pointed criticism through the years and releases, which has slowly improved the architecture and closed exploitable weaknesses. For many, these improvements are inadequate in both speed and scope. In such cases, stunnel is a valuable tool for authentication, isolation and privacy of critical data within Oracle.∎

Charles Fisher has an electrical engineering degree from the University of Iowa and works as a systems and database administrator for a Fortune 500 mining and manufacturing corporation. He has previously published both journal articles and technical manuals on Linux for *UnixWorld* and other McGraw-Hill publications.

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

RETURN TO CONTENTS

The Forrester Wave™: Digital Experience Platforms, Q4 2015

The demand to be at every touchpoint in the customer lifecycle is no longer an option—it's a requirement. To manage and deliver experiences consistently across all touchpoints, organizations are looking to digital experience platforms as the foundation of their digital presence.

Get Forrester's evaluation of the best vendors, including:

* The ten providers that matter most.
* How each vendor stacks up to Forrester's criteria.
* Six needs a digital experience platform architecture must meet.

> **http://geekguide.linuxjournal.com/content/forrester-wave-digital-experience-platforms-q4-2015**

ACQUIA™ The Ultimate Guide to Drupal 8 by Acquia

With 200+ new features and improvements, Drupal 8 is the most advanced version of Drupal yet. Drupal 8 simplifies the development process, enabling you to do more, in less time, with proven technologies that make it easier to be a first time Drupal user. Read this eBook, written by Angie Byron (you may know her as "webchick"), to get up to speed on the new changes in Drupal 8. Drupal 8's improvements include:

* API-driven content approach.
* Rest-first native web services.
* Seamless integration with existing technologies.
* Multilingual features and capabilities.
* Responsive by nature and mobile-first.

> **http://geekguide.linuxjournal.com/content/ultimate-guide-drupal-8**

ACQUIA™ How to Choose a Great CMS by Acquia

Web Content Management Systems serve as the foundation of your digital experience strategy. Yet many organizations struggle with legacy proprietary products that can't keep pace with the new realities of digital marketing. To determine if you are in need of a new CMS, use our guide, which includes:

* An evaluation to see if your current CMS supports your digital business strategy.
* The top considerations when selecting a new CMS.
* A requirements checklist for your next CMS.
* Ten questions to ask CMS vendors.

> **http://geekguide.linuxjournal.com/content/how-choose-great-cms**

Fast/Flexible Linux OS Recovery
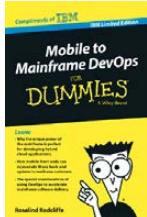
How long does it take to restore a system, whether virtual or physical, back to the exact state it was prior to a failure? Re-installing the operating system, re-applying patches, re-updating security settings takes too damn long! If this is your DR Strategy, we hope you've documented every change that's been made, on every system?!

Most companies incorporate backup procedures for critical data, which can be restored quickly if a loss occurs. However, that works only if you have an OS to restore onto and the OS supports the backup.

In this live one-hour webinar, learn how to enhance your existing backup strategies for complete disaster recovery preparedness using Storix System Backup Administrator (SBAdmin), a highly flexible full-system recovery solution for UNIX and Linux systems.

Webinar: April 26, 2016 at 1:00 PM Eastern

> **http://www.linuxjournal.com/storix-recovery**

Mobile to Mainframe DevOps for Dummies

In today's era of digital disruption empowered by cloud, mobile, and analytics, it's imperative for enterprise organizations to drive faster innovation while ensuring the stability of core business systems. While innovative systems of engagement demand speed, agility and experimentation, existing systems of record require similar attributes with additional and uncompromising requirements for governance and predictability. In this new book by Rosalind Radcliffe, IBM Distinguished Engineer, you will learn about:

- Responding to the challenges of variable speed IT.
- Why the mainframe is a unique and ideal platform for developing hybrid cloud applications.
- How mobile front ends can rejuvenate back-end systems to reach new customers.
- And, special considerations for using a DevOps approach to accelerate mainframe software delivery.

> **http://devops.linuxjournal.com/devops/mobile-mainframe-devops-dummies**

BRAND-NEW EDITION!
DevOps For Dummies – New Edition with SAFe®

In this NEW 2nd edition, learn why DevOps is essential for any business aspiring to be lean, agile, and capable of responding rapidly to changing customers and marketplace.

Download the E-book to learn about:

- The business need and value of DevOps.
- DevOps capabilities and adoption paths.
- How cloud accelerates DevOps.
- The Ten DevOps myths.
- And more.

> **http://devops.linuxjournal.com/devops/devops-dummies-new-edition-safe**

What's the Kernel Space of Democracy?

DOC SEARLS

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.

No one pretends that democracy is perfect or all-wise. Indeed, it has been said that democracy is the worst form of government except all those other forms that have been tried from time to time.—Winston Churchill

Might the same be said of operating systems and Linux? In both cases people live atop, and depend on, a deeper protected and enabling structure. In democracies, it's government. In Linux, it's the kernel. There are resemblances. For example, both need debugging, no matter how much bigger and better they get. A difference (one

of too many, I'll admit) is that Linux gets debugged by contributors and maintainers, while in democracies, users are free to mess with the government, and vice versa, by many means.

Still, although government space bears little resemblance to kernel space, the question haunts me, because I can't help thinking democracies might learn something from Linux.

It can't be easy. Politics never is. Although government runs on the form of code we call law, those that operate on it are not compelled to obey, or even to agree. Craig Butron highlights the difference when he says "all technical problems are technical and political—and you can always solve the technical ones."

The contradictions of democracy are old news. Otto Von Bismarck said "Politics is the art of the possible—the art of the next best." Yet he also said, "Not by speeches and votes of the majority, are the great questions of the time decided…but by iron and blood." While democracy might be as old as Greece, war is as old as our species, and perhaps even its evolutionary antecedents.

"Compared to war, all other forms of human endeavor shrink to insignificance", said George S. Patton. Among those forms of human endeavor are democracy and governance. Even the rule of law, which all democracies strive to maintain, is suspended gladly when battle flags are raised. In *War Is a Force That Gives Us Meaning*, Christopher Hedges writes:

> War makes the world understandable, a black and white tableau of them and us. It suspends thought, especially self-critical thought. All bow before the supreme effort. We are one. Most of us willingly accept war as long as we can fold it into a belief system that paints the ensuing suffering as necessary for a higher good, for human beings seek not only happiness but also meaning. And tragically war is sometimes the most powerful way in human society to achieve meaning.

Of course, meaning to one faction is not the same as to another. Many years ago, on a *Linux Journal* Geek Cruise (man, those were fun), I gave a talk about political differences—for example, how one side tends to see the market as a problem and government as a solution, while the

other side tends to see government as a problem and the market as a solution. My main source for that talk was George Lakoff's 1996 book, *Moral Politics: What Conservatives Know that Liberals Don't* (re-subtitled in 2002 as *How Liberals and Conservatives Think*). Even if you disagree with George's conclusions (or his politics), he does an outstanding job of laying out how liberals and conservatives talk past each other while constantly characterizing each other (sometimes correctly), and how hard it is for both sides to see truth in the others' framings. (Since the turn of the Millennium, George has written many other books, mostly to give Democrats new ways to frame debates. I believe Barack Obama would not have been elected, or re-elected, to the Presidency in 2008 and 2012, without those books. They mattered enormously to Obama's campaigns.)

After my talk, Andrew Morton said, "That's why the left thinks the right is evil, and the right thinks the left is stupid." I still can't think of a better summary statement.

I was given a similar hunk of wisdom by David Hodskins, my business partner for two decades and one of the smartest businessmen I have ever known: "Republicans are the party of wealth creation and Democrats are the party of wealth redistribution."

Consistent with that, I know Republicans who think all the good in the world is produced by business, while I know Democrats who think nature put a sum of money in the world and that it's the job of government to make sure those who have too much yield some of it to those who have too little. At one extreme, we have people who don't see how government produces any good (besides, as Ronald Reagan put it, "defending the borders"), and at the other extreme, we have people who don't respect much of what business does, besides providing jobs.

Around both poles gathers homophily: "the tendency of individuals to associate and bond with similar others". Those associations and bonds also comprise echo chambers within which even friendly and helpful voices tend to go unheard or misunderstood.

For example, a few years back, I suggested at a university meeting that we ought to define what the Internet actually is, because there is no common definitional ground between net-heads and bell-heads (those who see the Internet as a transcendent entity and those who see it as just a service offered by phone and cable companies). I said the university

would do the world a service by coming up with a canonical definition. A professor replied, "I didn't think we were going to work on policy this year." I replied, "Whoa! This isn't about policy. It's about language. Linguistics. Dictionary stuff." But the discussion drifted toward policy anyway, and the desire by nearly all present for net neutrality regulation, which made sense. After all, this was an echo chamber for netheads, including me (example: Cluetrain's New Clues). But I'm also a business guy, and at the time had at least *some* hope that a university could think in more, um, universal ways. But universities tend to be castles, and this was an example of the one on the left in Figure 1.

On the right were phone and cable companies who talk about "saving the marketplace" and "keeping government out of business", even though they are regulatory zoo animals holding their keepers so captive that they have successfully lobbied legislation restricting business to themselves in all but a handful of states (http://www.muninetworks.org/communitymap) and wouldn't know what to do in a truly open marketplace.

Speaking of netheads and bellheads, it's interesting to see how net neutrality has gone since I wrote about it in the July 2006 issue of *Linux Journal* (http://www.linuxjournal.com/article/8979). In it, I
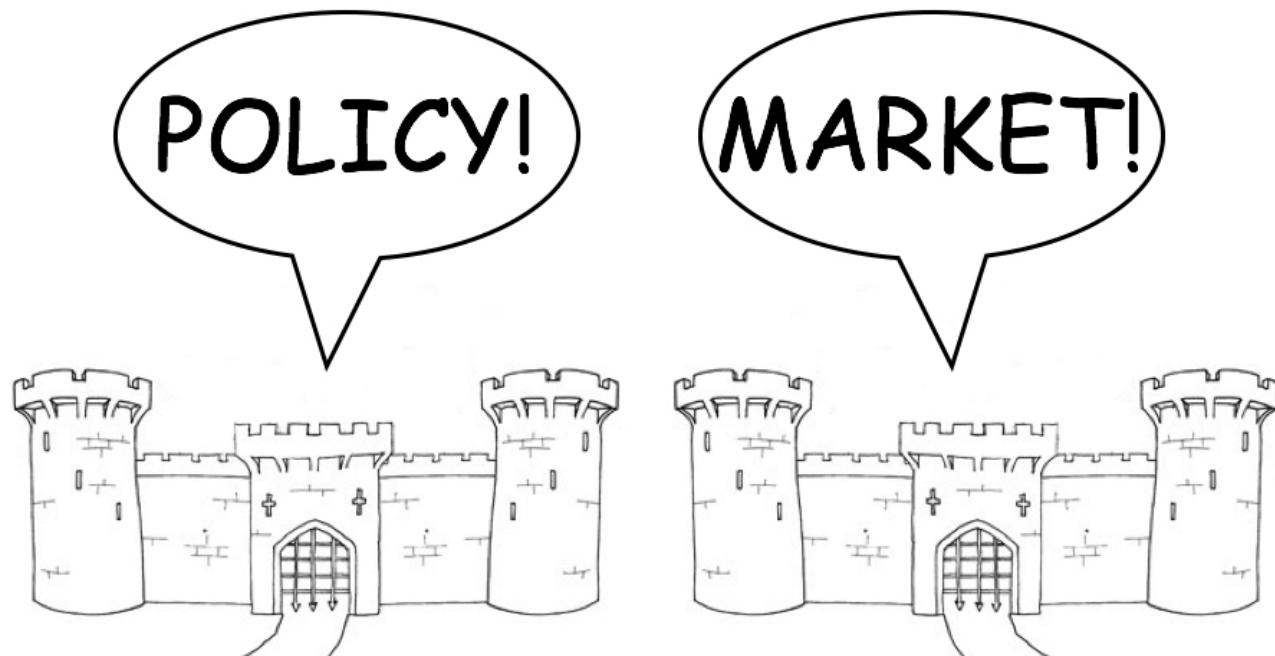

Figure 1. Two Castles

sourced Larry Lessig, Vint Cerf and Michael Powell, who was then a recently retired FCC chairman and is now cable's top lobbyist. Here are a few one-liners from a talk he gave at Freedom to Connect that year:

> …be careful of inviting the legislative process when they have a very bad understanding of the technical underpinnings. Because the secondary consequences of their errors can be enormous.

> I would rather try constantly to position my industry where I succeed if government does nothing, versus positioning it in a way where I need them to do something or I'm dead.

> I can tell you right now, very few people in Washington (understand net neutrality). This means you're going to get a potentially very ambiguous, subject to massive variations in interpretation, pile of law.

> …government has a way of turning on people. Ask Bill Gates. It may be about networks today, but those same principles can be used against innovative business models and applications in other contexts. And, I submit to you they would be.

> Be careful because you're playing their game.…The average one of these incumbents, whether a cable or a phone company, have 40 lawyers in Washington dedicated to this work. Resources. Ability. One hundred years of skill.…Then, let me add the judicial process. Every decision you get from the Congress and the FCC will spend the next three and a half to four years in court.

Since then, no net neutrality legislation proposed in Congress has moved past the committee stage. But on the regulatory side, the FCC in 2010 issued the pro-neutrality Open Internet Order. In September 2011, the FCC added a final rule titled Preserving the Open Internet. Naturally, Verizon sued, eventually winning Verizon v. FCC in the DC Circuit Court of Appeals, which vacated two of the three main portions of the FCC's Order. That happened in May 2014—right in line with Powell's estimate for time in court.

But the battle is hardly over. In February 2015, the FCC issued "strong, sustainable rules to protect the open Internet" by re-classifying "broadband Internet access service" under Title II of the Telecommunications Act. Like everything Federal, it's complicated, which means it met mostly with approval by netheads and disapproval with the bellheads. Expect some of this to be worked out (and not) over the next few years.

Meanwhile, Linux won without going to war, even though it had lots of opposition from the start. Microsoft hated it, as did competing UNIX and UNIX-ish operating system camps.

I showed up when Linux itself was to the right of the decimal point, in the early 1990s. I was invited by Phil Hughes, who included me in an e-mail conversation about starting a free software magazine. This became *Linux Journal* in April 1994, when Linux arrived at v.1.0. (Little known item: the first editor was Bob Young, who left to start Red Hat.) In those days, and to a large degree ever since, the Linux homophily has been largely a libertarian one. Perhaps this is a manifestation of the distinction between kernel space and userspace. The perspective of the kernel on userspace is like that of the Earth's core toward what happens on its surface. As Linus often says about userspace, "I don't care."

Yet all of us live in userspace, and we have politics here, especially between now and November, when the US holds its presidential election. While that's happening, I suggest we technical folks look at the process as

ADVERTISER INDEX

one that requires debugging. Here are my suggested few:

1. Get money as far as possible out of politics. Larry Lessig has required reading on this (http://republic.lessig.org).

2. Amend the Constitution to eliminate the Electoral College and allow direct election of the President.

3. Eliminate gerrymandering, as far as possible.

4. Expand voting rights to all citizens.

All those are simply for restoring democracy and nothing more. Like many in our community, I not only want better government, but less of it, especially as we work out what it means to be digital as well as analog creatures. For particulars on that, check out what I said about Ralph Nader here in 2000 (http://www.linuxjournal.com/article/4316). A sample:

Consumers and workers are rhetorical relics. The Net is uniting both, and they're throwing off their chains. Industrial communism and capitalism are both terminal. They can't survive in a networked marketplace, where "We the People" means exactly what it says.

We have real challenges in this marketplace, not the least of which is understanding how it's going to work, now that "the People" are working both sides of the supply-and-demand handshake. What are the new social contracts? What laws do we really need, and why are we keeping the ones we don't? How can we truly help those who can't help themselves? What are our agreements about privacy and anonymity, and how do we organize them? How do we build the needed infrastructure around our new Commons, and how do we keep those stuck with dead industrial market models from wasting our time?

Those questions still stand. So does the one in the headline above.■

RESOURCES

Otto Von Bismarck: https://en.wikipedia.org/wiki/Otto_von_Bismarck

George S. Patton: https://en.wikipedia.org/wiki/George_S._Patton

Christopher Hedges: https://en.wikipedia.org/wiki/Chris_Hedges

George Lakoff: https://en.wikipedia.org/wiki/George_Lakoff

George Lakoff's Books: http://georgelakoff.com

Andrew Morton: https://en.wikipedia.org/wiki/Andrew_Morton_%28computer_programmer%29

Homophily: https://en.wikipedia.org/wiki/Homophily

"Netheads vs Bellheads" by Steve G. Steinberg, *Wired*: http://www.wired.com/1996/10/atm-3

The Cluetrain Manifesto: http://cluetrain.com

New Clues: http://newclues.cluetrain.com

Lawrence Lessig: https://en.wikipedia.org/wiki/Lawrence_Lessig

Vint Cerf: https://en.wikipedia.org/wiki/Vint_Cerf

Michael Powell: https://en.wikipedia.org/wiki/Michael_Powell_%28lobbyist%29

FCC Open Internet Order 2010: https://en.wikipedia.org/wiki/FCC_Open_Internet_Order_2010

Federal Communications Commission, Preserving the Open Internet; Final Rule:
https://www.gpo.gov/fdsys/pkg/FR-2011-09-23/pdf/2011-24259.pdf

Verizon Communications Inc. v. FCC (2014):
https://en.wikipedia.org/wiki/Verizon_Communications_Inc._v._FCC_%282014%29

"FCC Adopts Strong, Sustainable Rules to Protect the Open Internet":
http://transition.fcc.gov/Daily_Releases/Daily_Business/2015/db0226/DOC-332260A1.pdf

Bob Young: https://en.wikipedia.org/wiki/Bob_Young_%28businessman%29

"Cruise Report 3: New Species Discovered at Sea" by Doc Searls, *Linux Journal*:
http://www.linuxjournal.com/article/8664

Gerrymandering: https://en.wikipedia.org/wiki/Gerrymandering

ACLU: Voting Rights: https://www.aclu.org/issues/voting-rights

"Let Freedom Ping" by Doc Searls, November 2000, *Linux Journal*:
http://www.linuxjournal.com/article/4316

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

RETURN TO CONTENTS

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!