LINUX JOURNAL™
Since 1994: The Original Magazine of the Linux Community
SEPTEMBER 2015 | ISSUE 257 | www.linuxjournal.com

CONTROL YOUR 3D PRINTER WITH A RASPBERRY PI

HOW-TOs: SECURITY

Harden TLS and SSH with CIPHERS

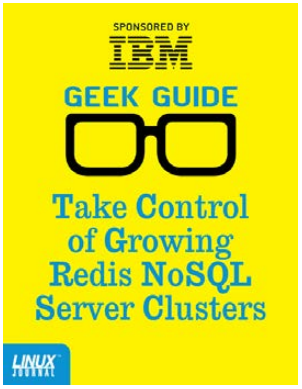
Lock Down Your System Using UEFI SECURE BOOT

+ CONSIDERING THE INTERNET OF THINGS

OPTIMIZE YOUR WEB APPLICATIONS

STARGAZING WITH LINUX

WATCH: ISSUE OVERVIEW



Take Control of Growing Redis NoSQL Server Clusters

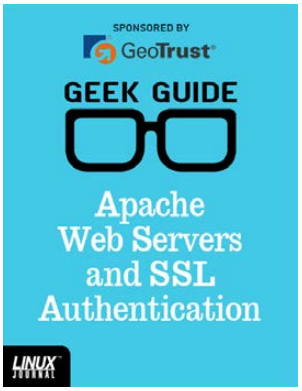
Author: Reuven M. Lerner

Sponsor: IBM

Linux in the Time of Malware

Author: Federico Kereki

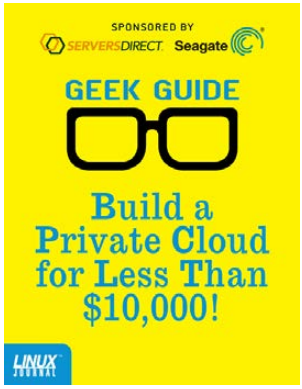
Sponsor: Bit9 + Carbon Black



Apache Web Servers and SSL Encryption

Author: Reuven M. Lerner

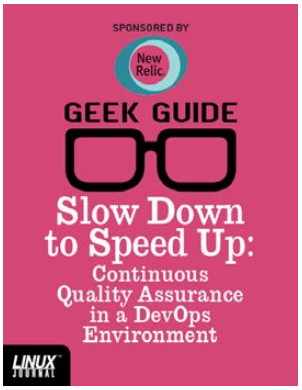
Sponsor: GeoTrust



Build a Private Cloud for Less Than $10,000!

Author: Mike Diehl

Sponsor: Servers Direct and Seagate



Slow Down to Speed Up: Continuous Quality Assurance in a DevOps Environment

Author: Bill Childers

Sponsor: New Relic

Enterprise Monitoring Success

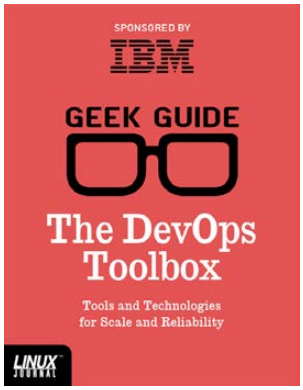
Author: Mike Diehl

Sponsor: InterMapper

Beyond Cron: How to Know When You've Outgrown Cron Scheduling—and What to Do Next

Author: Mike Diehl

Sponsor: Skybot



The DevOps Toolbox: Tools and Technologies for Scale and Reliability

Author: Bill Childers

Sponsor: IBM

CONTENTS

SEPTEMBER 2015
ISSUE 257

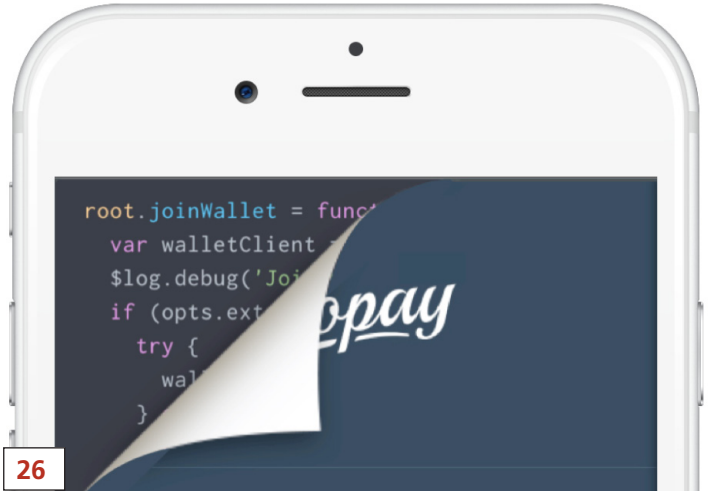
HOW-TOs: SECURITY

FEATURES

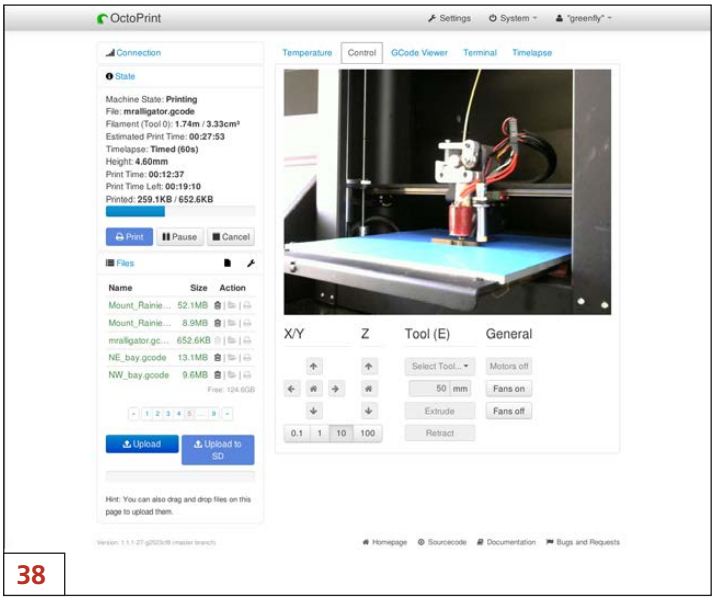
ON THE COVER

COLUMNS

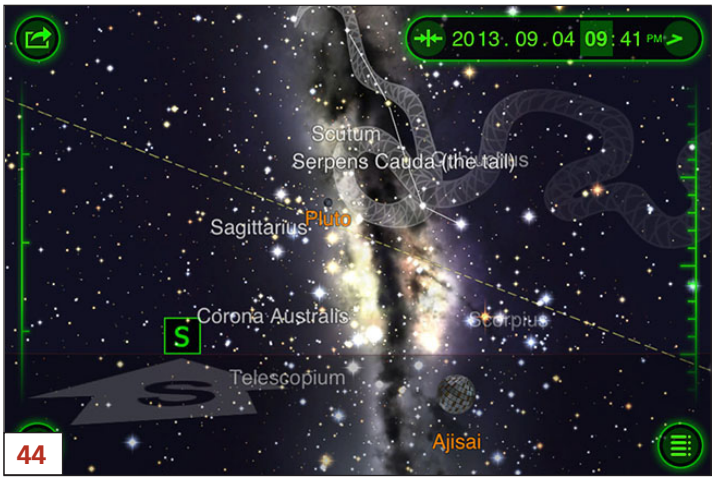
IN EVERY ISSUE



26



38



44

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

LINUX JOURNAL

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

Contributing Editors

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

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

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

Take your Android development skills to the next level!

AnDevCon

The Android Developer Conference

Dec. 1-3, 2015

Hyatt Regency Santa Clara

Register Early! AnDevCon Santa Clara will sell out!

Get the best Android developer training anywhere!

- Choose from more than 75 classes and in-depth tutorials
- Meet Google and Google Development Experts
- Network with speakers and other Android developers
- Check out more than 50 third-party vendors
- Women in Android Luncheon
- Panels and keynotes
- Receptions, ice cream, prizes and more (plus lots of coffee!)

Whether you're an enterprise developer, work for a commercial software company, or are driving your own startup, if you want to build Android apps, you need to attend AnDevCon!

Check out the program online at www.AnDevCon.com

Android is everywhere! But AnDevCon is where you should be!

A BZ Media Event #AnDevCon

SHAWN POWERS

How to Do That Thing You Do

I love to learn. I've always been a learner, so grade school and college were both extremely enjoyable for me. That doesn't mean I always got the best grades, because I never really cared about how successful I was; I just cared that I absorbed more knowledge and skills. When I was in college (early 1990s), the Internet came into its own, and my insatiable desire for knowledge found its Holy Grail. This issue of *Linux Journal* is dedicated to learning how to do things. If you're like me, that's an awesome prospect, so I won't delay with lots of pontificating.

Reuven M. Lerner starts the issue by discussing what it means to be "fast" in the world of Web applications. It seems like a simple topic, but since speed can refer to many things, and those things can be affected by even more things, it can become complicated fairly quickly. Reuven helps you figure out how to go about speeding up your applications by identifying what you actually should be trying to do in the first place.

In Dave Taylor's scripting column, you'll learn how to play with blocks. No, not blocks of data—children's letter blocks. Whether you want to spell out silly messages to your kids or just want to know how many various words you could spell with a limited number of blocks, Dave's column this month aims to please. Kyle Rankin follows Dave with his final article in his four-part series on 3D printing. He's covered a lot during the past few months, and this final article brings it all together with a how-to on using OctoPrint. It's a great Web-based program for controlling your printer, and if you're thinking about delving

VIDEO:
Shawn Powers runs through the latest issue.

into the world of 3D printing, you'll want to catch Kyle's latest installment.

It gives me a great amount of joy seeing space exploration once again gaining popularity in our society. I worried the end of the Space Shuttle program would start a dry spell for NASA, and although NASA's budget is continually slashed, it hasn't stopped interest in space. Last month's Perseid meteor shower, the recent flyby of Pluto and even the upcoming movie *The Martian* inspired me to write about how to use your computer to help enjoy the cosmos. Being Linux users doesn't hinder you from using software to view the stars, and in this month's column, I discuss the various applications available to aid in your nighttime up-looking or your ability to explore space from your monitor. By the time you read this, the software likely will have updated images from Pluto, so be sure to check it out if you're a space nut like me!

Anyone who has ever been frustrated with UEFI Secure Boot will want to read Greig Paul and James Irvine's article on how to bend UEFI to your will. If you take control of its security features, you can protect your entire system by encrypting the entire disk. I love turning problems into opportunities, and this month, Greig and James provide a perfect opportunity to do just that.

Charles Fisher finishes out this how-to issue with a very important look at hardening TLS and SSH encryption. Thanks to recent exploits, it's painfully obvious that the old reliable SSL encryption is no longer secure. At all. Even SSLv3 has been proven insecure, and so you need to know what to do to keep your data safe. Charles covers the nitty-gritty information on not only what versions of TLS encryption you should be using, but what ciphers are currently reliable. If you're interested in encryption, or are responsible for securing data on your network, be sure to read his article, as it gives a helpful, if sobering, look at the current state of encryption software.

This issue, like every issue of *Linux Journal*, has many other articles, tips, reviews and community announcements. If you're a lifelong learner like me (and if you're reading *Linux Journal*, the chances are fairly good you are), you'll find this issue particularly enjoyable. At the very least, hopefully my article will give you a reason to go outside and look to the stars. Looking into the heavens always reminds me just how much there is to learn and discover.■

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

letters

Size of Print

I was going to praise your magazine with printing code that I could read in the article on containers. (Okay, I am a little behind in my reading.) But, then I go to At the Forge, and I have code in a font so small that even when I enlarge it, it's too small. Please help an old man's eyes.

—Doug Broadie

Trying to fit code into a readable, not word-wrapped format that is flexible between devices is surprisingly difficult. It's a challenge every month for our layout person. Although I'm sure there will be some articles that are frustrating with particularly long lines of code, we'll keep trying to make it all readable—or as readable as possible!—Shawn Powers

OpenSSL vs. Libre SSL

Considering the vulnerabilities that have become known recently in OpenSSL, please publish an article on OpenSSL vs. LibreSSL, and when LibreSSL may be ready for prime time. I did find this article that is about a year old: https://blog.hboeck.de/archives/851-LibreSSL-on-Gentoo.html.

—Richard

Thanks for the tip. Hopefully a contributor will grab your idea and give it a go.—Shawn Powers

General Relativity in Python

I just wanted to thank Joey Bernard for the great introduction to SymPy [see Joey's article in the July 2015 Upfront section]. It always give me a kick to read (and sometimes replicate) on my cobbled-together Linux box what once was the demesne of secret government research facilities. As more and more emphasis in placed on reproducible research rather than on splashy headlines, these types of articles will be invaluable.

—Kwan L. Lowe

Joey Bernard replies: *It is always great to hear from people who get some use from the articles I write. In my day job, I am constantly pushing on researchers the importance of reproducible research, so consider me a kindred spirit.*

Find and Xargs

Here's a follow-up comment regarding Gary Artim's letter and Shawn Powers' reply in the July 2015 issue.

Although `find`'s `exec` option can be handy, I would respectfully disagree that it's "probably even more efficient" than `xargs`. As I understand it, the user-supplied command is invoked once for every file that matches `find`'s criteria.

Each command invocation involves overhead: program initialization, option processing, cleanup and so on.

For commands that support multiple files (such as `rm`, `tar`, `zip`, `grep` and so on), the `xargs` approach is better. The filenames are collected into a list, and the user-supplied command is invoked a minimal number of times. The overhead is minimized.

For small numbers of matching files, the `exec` option won't cause any noticeable delay. But, if the user-supplied command is non-trivial, or if there are large numbers of matching files, the `xargs` approach can avoid a lot of wasted processing time.

As a followup article, I would suggest discussing how to use a `find-xargs` pipeline to execute commands where the file list is not the last item on the command line (for example, copying or moving a set of files to a destination directory). The `xargs` command is supposed to support this, but in my experience, it's quite fragile. Maybe I'm missing something.

—Chris

Hmm, I'll have to look into this and see if I can make an interesting article out of it. Thanks for the tip!—Shawn Powers

System Status as SMS Text Message

Regarding Dave Taylor's article "When Is a Script Not a Script" in the May 2015 issue: here's just an alternate way to text message, not requiring

mail setup, using curl:

```
#!/usr/bin/bash
while [ 1 ]
do
        pingout=$(ping -c 1 textbelt.com)
        STATUS=$?
        if [ "$STATUS" -eq "0" ]; then
                break
        else
                sleep 5
        fi
done
ipaddress="$(/usr/bin/hostname -i)"
/usr/bin/curl http://textbelt.com/text
 ➥-d number=9999999999 -d "message=$ipaddress"  >
/var/log/rc.local.log 2>&1 &
exit 0
```

Cheers! I enjoy your column.
—Gary Artim

July 2015 *Linux Journal*

As always, it's a pleasure to read and so much to learn. The only typo I spotted in the July 2015 issue was on page 38 with Dave Taylor's "Working with Functions: Tower of Hanoi", as I am sure everyone will mention. The solution is (2**n)-1 and not (2**n)+1 as printed.
—John

Dave Taylor *replies: By George, I think you're right!*

July 2015 Work the Shell

In the July 2015 issue, Dave Taylor writes, "The biggest limitation with shell functions is that they can return an integer value only of 0–127, where a typical script actually utilizes the 0 = false or failure, 1 = true or success", and the above is even repeated as a big banner across the top of the page 37.

These claims are quite easy to verify. Let's try the following shell script:

```
check () {
    return $1
}

for x in 254 255 256 257 258 ; do
    check $x ; echo $?
done
if check 0 ; then echo "A" ; fi
if check 1 ; then echo "B" ; fi
```

If you take the quoted statement at face value, you will likely be quite surprised by results of running this script with the help of a few different shells (say bash, dash, zsh, ksh) accepting the syntax. What you will see may be somewhat different in places but not in a material way.
—Michal Jaegermann

Dave Taylor replies: *Thanks for*

writing in, Michal. Of course, a function can return a numeric value in the shell, and that value can be interpreted as the calling script desires. My point was more that function return values are constrained by the simple integer range—whether it's 0–127 or 0–65536—and so I generally just ignore return values from functions entirely and use global variables instead. If we were in any more sophisticated programming language, of course, functions could return any number of complex data types and sidestep the limitation.

Thanks for the Docker Articles

Thanks very much for the Docker articles in the June 2015 issue of *LJ*! They were quite a helpful primer. [See Shawn Powers' "Doing Stuff with Docker" and Federico Kereki's "Concerning Containers' Connections: on Docker Networking".]

—Erik

You're very welcome! I'm a big fan of breaking down complicated or scary topics into the basics. It's mainly because I know that awkward, embarrassed feeling of being in a conversation with others about a new technology (such as Docker) and not having any clue what they're talking about. Information wants to be free! (Okay, I'll get off my soapbox. Thanks again for the kind words.)—Shawn Powers

Shared Libraries and Kernel Mode

In the July issue of *LJ*, Zack Brown's "diff -u" article discusses the efforts to extract part of the kernel code into libraries. He likens this to the debate about micro versus monolithic kernels. I thought that the current debate was about making libraries that could be used elsewhere (and/or substituted) but still running in kernel space. I understand that Linus Torvalds is enthusiastic to push stuff to user space, but I don't believe that was directly related to the current discussion of the "library-ification" of kernel code.

—Ray Foulkes

Zack Brown replies: *Thanks for the letter! You're right that current efforts to librarify the kernel don't make it more of a microkernel and are not part of that debate.*

But, I wanted to link the two, because I feel they attempt to achieve similar things. In a microkernel, as you say, the user can swap out chunks of the system that inhabit user space, replacing them with new or experimental implementations.

The library-ification effort is similar; although in the case of libraries, the control is given at compile time rather than at runtime. The user may replace one subsystem with another in a relatively orderly way, using a known library interface.

It's this orderliness that for me has a similar feel to a microkernel approach. Both cases rely on clear interfaces that provide the ability to replace whole subsystems, without requiring a deep rewrite of adjoining kernel code.

So I don't mean to say that the library-ification effort is part of an actual switch to a microkernel design. But—and now giving vent to pure speculation—I do think that library-ification could someday make it easier for Linux to support "hotswapping" arbitrary subsystems in a running kernel (at which point Linux truly might start to resemble a microkernel in practice).

WRITE *LJ* A LETTER

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

PHOTO OF THE MONTH

Remember, send your Linux-related photos to ljeditor@linuxjournal.com!

LINUX JOURNAL

At Your Service

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

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

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

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

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

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

Percona Live Europe

Data Performance Conference 2015

September 21–23
Amsterdam

3 days of tutorials & breakout sessions by MySQL & NoSQL pros

www.percona.com/live

diff -u
WHAT'S NEW IN KERNEL DEVELOPMENT

Boot times can become slow on systems with many CPUs, partly because of the time it takes to crank up all the **RAM chips**. **Mel Gorman** recently submitted some patches to start up RAM chips in parallel instead of one after the other. One of the main problems with trying to implement such a feature—and one of the main reasons such patches haven't made it into the kernel before—is the need to avoid slowing things down for smaller systems.

Mel's patches modified the **kswapd** code to give each CPU its own RAM initialization thread. On smaller systems this theoretically would amount to no change at all, while larger systems could see dramatically reduced boot times.

An initial test by **Waiman Long** reported a 25% reduction in boot time on his 12 terabyte system—from 404 seconds to 298. And when **Peter Zijlstra** and Mel asked if this made a worthwhile difference to him, Waiman replied:

> Booting 100s faster is certainly something that is nice to have. Right now, more time is spent in the firmware POST portion of the bootup process than in the OS boot. So I would say this patch isn't really critical right now as machines with that much memory are relatively rare. However, if we look forward to the near future, some new memory technology like persistent memory is coming, and machines with large amount of memory (whether persistent or not) will become more common. This patch will certainly be useful if we look forward into the future.

And **Scott J. Norton** also added, "100 seconds really does matter and is a big deal. When businesses have one of these large machines go down, their business is stopped (unless they have a fast failover solution in place). Every minute and second the machine is down is crucial to these businesses."

There was a bit of a push by **Andrew Morton** for Mel to simplify his code, but Mel felt that Andrew's

suggestions could make things worse, such as forcing the kernel to rely on user-space code. And so long as systems keep getting bigger, patches like these seemed destined for eventual acceptance.

Intel has invited Linux kernel engineers to assist in development for chips that are so new that their in-house developers must code on software simulations of the eventual hardware.

The patches, released by **Dave Hansen**, wouldn't run for anyone outside Intel—since no one else has those chips—but he was hoping for feedback on their implementation of **Memory Protection Keys** for user space.

The underlying idea involves utilizing previously unused bits from existing registers and introducing new registers and associated assembler instructions to secure system memory on a page-by-page basis. Essentially, this gives users the ability to enable a particular set of actions on a given set of pages, while prohibiting others.

When **Ingo Molnar** asked Dave to list some potential use cases for this chip feature, Dave replied that there were various things that a user might want to protect, such as the following: "Data structures like logs or journals that are only written to in very limited code paths, but that you want to protect from 'stray' writes." Or: "a database where a query operation will never need to write to memory, but an insert would. You could keep the data R/O during the entire operation except when an insert is actually in progress."

And, **Alan Cox** also suggested:

> You also can use it for certain types of emulator trickery, and I suspect even for things like interpreters and controlling access to "tainted" values.
>
> Other obvious uses are making it a shade harder for SSL or ssh type errors to leak things like key data by reducing the damage done by out of bound accesses.

Ingo asked if there could be any issues surrounding this feature existing on some CPUs but not others. And Dave replied, "It's always a problem with new CPU features." He then went on to say:

> I've thought a bit about trying to "emulate" the feature on older

> CPUs using good ol' mprotect() so that we could have an API that folks can use *today*, but that would get magically fast on future CPUs. But, the problem with that is the thread-local aspect.
>
> mprotect() is fundamentally process-wide and protection keys are fundamentally thread-local. Those things are going to be hard to reconcile unless we do something slightly extreme like having per-thread page tables.

The discussion got technical, but clearly the main question is how to support the new chip features, rather than whether to support them at all.

Luis R. Rodriguez has extended **module signing** to support signing firmware as well. Eventually, he figures it should be possible to sign user data too. This seemed to be a natural extension of existing features and not very controversial. But, there are certain differences between the firmware signing code and the module signing code; for example, Luis' code introduces separate files to contain the firmware signatures as a means to better handle licensing issues.

Luis' patches also "do not taint the kernel in the permissive [firmware] signing mode due to restrictions on the firmware_class API; extensions to enable this are expected, however, in the future."—**ZACK BROWN**

They Said It

The highest reward for man's toil is not what he gets for it, but what he becomes by it.
—John Ruskin

The greatest conflicts are not between two people but between one person and himself.
—Garth Brooks

I can't give you a sure-fire formula for success, but I can give you a formula for failure: try to please everybody all the time.
—Herbert Bayard Swope

Who begins too much accomplishes little.
—German Proverb

It is impossible to live without failing at something, unless you live so cautiously that you might as well not lived at all. In which case, you've failed by default.
—J. K. Rowling

LINUX JOURNAL
on your
Android device

Download the app now on the **Google Play Store**

www.linuxjournal.com/android

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

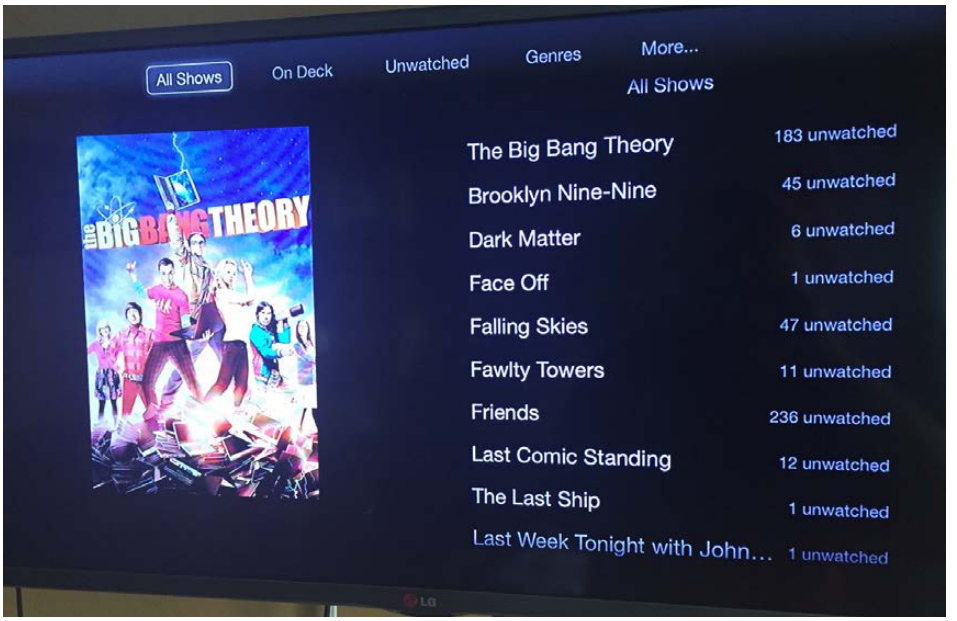
BitTorrentSync: Dropbox for Nerds

It's not really fair to compare Dropbox directly with BitTorrentSync. First of all, my title implies Dropbox is somehow inferior. To be honest, I haven't found anything that works as smoothly as Dropbox when it comes to sync reliability and ease of installation. That said, although it has incredible strengths, it also has a few shortcomings. One, it's not free, and if you have a lot of data, it can be expensive. Two, your data is stored on the Dropbox servers. The second one is a showstopper for many folks.

BitTorrentSync uses the BitTorrent protocol to keep multiple locations in sync. Everything is stored on your own machines, and you're limited only by your own storage limitations. The basic features of BitTorrentSync are free, but like Dropbox, there is a paid version. For me, quantity of storage is the most important feature with a syncing app, and with BitTorrentSync, that's unlimited. If you've been in the market for a Dropbox replacement, I urge you to check out BitTorrentSync (or more properly just "Sync" as it's been rebranded, probably due to the bad press "torrents" get). It's matured to the point that I trust it with my data, and its cross-platform nature makes it easy to distribute. Grab a copy at http://www.getsync.com.

—SHAWN POWERS

Non-Linux FOSS: PlexConnect



It's no secret that I'm a huge fan of Plex. It might be a secret, however, that I live in a house with quite a few Apple products. That said, I find the Apple TV to be one of the most limiting, frustrating set-top boxes to work with. (I'm sure most readers would agree.) I prefer to be a lover, not a hater, so I searched long and hard to find a way to make the Apple TV suck less. Thanks to PlexConnect, I succeeded.

The Apple TV is still not rootable (if you see claims that it is, you're likely being bamboozled). PlexConnect works around the walled garden of iOS by hijacking an official Apple app (the Trailers app specifically) and allowing access to a Plex server.

The open-source PlexConnect is really just a brilliant translation layer that hijacks DNS (pointing http://trailers.apple.com to the PlexConnect server IP) and feeds the Apple TV data formatted like it expects. Rather than showing a listing of recent movie trailers, however, PlexConnect shows a direct interface with your Plex media server. And to be honest, the interface is actually surprisingly pleasant to use.

If you're stuck using an Apple TV for your living-room media playing, or if you'd simply like to hop over that walled garden just because you can, check out PlexConnect today. It's open source and available on GitHub: https://github.com/iBaa/PlexConnect.

—SHAWN POWERS

Finite-Element Methods for PDEs

One of the common classes of equations that is encountered in several branches of science is partial differential equations. So in this article, I look at a software package called FreeFem++ that is designed to help you calculate these partial differential equations (http://www.freefem.org).

One popular method of solving these types of equations, and the one FreeFem++ uses, is the finite-element method (https://en.wikipedia.org/wiki/Finite_element_method). The basic idea with this method is to take the entire problem domain and subdivide it into a mesh of smaller regions. You then apply a simplified version of the partial differential equations that still is locally valid. This makes the problem a tractable one that actually can be solved in a reasonable amount of time.

FreeFem++ is available in several different flavors. The earlier versions were named freefem, freefem3D and freefem+. The latest version is called FreeFem++ and is written in C++. It can be compiled and runs on all three major operating systems, Windows, Mac OS X and Linux. Since this is *Linux Journal*, I focus on Linux here. On Debian-based distributions, installation is as easy as:

```
sudo apt-get install freefem++
```

The other versions also are available as the freefem and freefem3d packages.

Although I am specifically discussing solving electromagnetic fields in this article, FreeFem++ is a general finite-element method solver. This means it should be able to deal with most partial differential equations.

Once it is installed, you will want to start using it. The first thing to be aware of is that FreeFem++ is designed to be used for "production-level" work—meaning active, high-level research work. As such, it does not have a pretty front end to help new users through their first attempts at using it. It is best to think of FreeFem++ as a programming language that you use to write a program to solve your problem.

The first step is to define the geometry of your problem. This

step is usually called meshing, or building a mesh. FreeFem++ does not include any CAD functionality to build geometries directly. You can, however, generate meshes automatically from a set of border descriptions. FreeFem++ can take a set of equations describing your geometry and generate a mesh based on the Delaunay-Voronoi algorithm. As a simple example, let's say you wanted to build a square object. You could create the related mesh with these commands:

```
int sides = 8;
mesh Th = square(sides, sides);
```

From this first example, you should notice that the language FreeFem++ uses is very similar to C/C++. Variables are typed and can store values only of that type. The language is also polymorphic, so commands and operations will do different things based on the types of the parameters or operands.

You can define more complex shapes with border functions. An example looks like this:

```
border aa(t=0, 2*pi) {x=cos(t); y=sin(t);}
```

You then can hand these types of objects in to the `buildmesh()` function to generate the same type of mesh you would have received from the higher-level `square()` function.

If you want to see what these equations actually look like (to verify that you have it correct), you can use the `plot()` function to visualize them. You can hand in the border functions you may have created or even the mesh object you get from `buildmesh()`.

Once the mesh is created, you need to create a set of two-dimensional spaces from this mesh in order to solve your problem. This is done with the `fespace` function:

```
fespace Vh(Th, P1);
Vh u,v;
```

The `P1` parameter tells `fespace` what type of finite element you want, whether it is continuous or discontinuous, smooth, linear or with a bubble. (A rather large set of possibilities is available that will be left as an exercise for the dear reader.)

Next, you need to define the finite-element functions within this newly generated space—in this case `u` and `v`.

Now that you have all of the background scaffolding built, you need to define your problem and solve it within your problem geometry. You can use the `problem` type to

define a more complex problem. As an example, you might want to look at the cooling of a hot plate. The following would set up the problem for you:

```
mesh Th=square(30,5,[6*x,y]);
fespace Vh(Th, P1);
Vh u=u0, v, uold;
problem thermic(u,v)=int2d(Th)(u*v/dt + k*(dx(u)
➥* dx(v) + dy(u) * dy(v)))
             + int1d(Th,1,3)(alpha*u*v)
             - int1d(Th,1,3)(alpha*ue*v)
             - int2d(Th)(uold*v/dt) + on(2,4,u=u0);
```

The variable name `thermic` is now a function call. When you issue the command `thermic`, FreeFem++ will go ahead and solve this problem that you defined. The purpose for this method is to be able to define your problem and make alterations and adjustments before actually solving it.

If the problem is simpler to define, you can use the `solve` command to define your problem and do the solving step immediately. For example, if you wanted to model motion on a membrane, you could use something like this:

```
solve Laplace(phi,w)=int2d(Th)(dx(phi)*dx(w)
➥+ dy(phi)*dy(w))
         - int2d(Th)(f*w) + on(Gamma1, phi=z);
```

where the appropriate finite-element space and functions have been defined. Once FreeFem++ has solved your problem, you can use `plot` with the finite-element functions to visualize the actual results of this numerical solution.

Although being able to visualize the results of your work immediately is important, you need to have a way of saving this work so you don't need to repeat any calculations unnecessarily. You can save meshes that are generated with the `savemesh` function. You simply need to hand in the mesh to save and a filename to use:

```
savemesh(Th, "my_mesh.msh");
```

You can reload this mesh at a later time with the `readmesh` command, for example:

```
mesh Sh = readmesh("my_mesh.msh");
```

Outputting results is a bit more of a hassle. You have access to the standard C++ input/output streams, specifically cin and cout, so you can dump out the numerical results that way. You also can create a new output stream with `ofstream` to write things out to a specific file, rather than to what standard output

is redirected to. In this way, you have full control over what data gets saved, and what format this file and its data uses.

Now that you've read this introduction to FreeFem++, you should take a look at other tutorials and documentation on the Web. Several good examples are available that should give you at least a starting point to solve the specific problem you are investigating. If the problem you are trying to solve is especially large, FreeFem++ also has MPI support available. In this way, you can spread your calculations over potentially hundreds of CPUs and hopefully get even more work done.

—JOEY BERNARD

iTVMediaCenter: Scam or Brilliance?

The folks at iTVMediaCenter recently contacted me about their one-stop-shop solution for cord-cutters. For $14.99, they sell a program that consolidates tons of on-line media into a central location so you can watch it on demand. The problem is, it looks like the application does little more than open the same Web sites you can open with a browser. Also, the "one-time fee" is rumored to be an annual fee. I can't verify whether the fee is recurring, but honestly, I don't recommend it either way. What I can recommend, however, is to visit their Web site!

Although the application is seemingly a bust, the Web site is free, and it does a decent job of linking to dozens and dozens of on-line streaming sites. If you're looking for some free on-line media, but don't know where to begin, go to http://itvmediacenter.com and check out the on-line offerings. I urge you to be cautious about the application, but I found the free on-line catalog to be very helpful.

—SHAWN POWERS

Android Candy: Copay—the Next-Generation Bitcoin Wallet

When I hear the word "copay", I think of the doctor's office. Thankfully, the Copay app from the folks at Bitpay doesn't cost you anything, and it keeps your Bitcoin healthy and secure. I've mentioned many Bitcoin wallet applications and cloud solutions during the past few years, but Copay truly is different. It has features other wallets can't touch, such as:

- Truly cross-platform with availability for Android, iOS, Windows, OS X, Linux and Chrome.
- Fast Bitcoin communication with Bitcoin network, no blockchain download.
- Payment verification (BIP-0070-0073).
- Shared wallets (multi-party transfer approval).
- Wallet backups, multi-device access of same wallet.

(Photo Credit: https://copay.io)

- 100% open, downloadable source code hosted on GitHub.

I'll admit the first time I tried Copay, I didn't quite understand the hype. It feels like every other app accessing a cloud-based Bitcoin wallet (for example, Coinbase). But Copay is an actual wallet, with private keys stored only where you back them up. Thankfully, it allows simple backup of your wallet keys, so you can access your Bitcoin from multiple locations. In fact, you really really *really* need to have your wallet backed up and/or accessed from multiple locations. If you lose your phone and don't have a backup of your wallet, there's no way to recover your Bitcoin.

Thanks to its devotion to multi-platform open-source development and attention to security while never compromising flexibility, Copay gets this month's Editors' Choice award. If you want to manage your own Bitcoin without trusting an on-line cloud provider, Copay puts the control in your hands. Check it out today at https://copay.io.—**SHAWN POWERS**

REUVEN M. LERNER

What Does "Fast" Mean?

Want to optimize your Web application? First, decide what you mean by "fast".

Good news! One of my clients is launching a new marketing campaign, which we expect will make the business even more successful than before.

Bad news! This means our Web application, which has existed for some time on a fairly simple infrastructure, and which has handled a steadily growing number of users, now (we hope) will need to deal with a massive spike in users.

The big question is this: can our servers handle the load we expect? Indeed, what load can we expect? And, what happens if we need to crank up even more capacity?

So in this article, I walk through some of the basic points having to do with Web scalability, describing a few of the key things to keep in mind. Next month, I'll take a deeper dive into these ideas and discuss some of the techniques you can use to improve the speed—or apparent speed—of your applications.

Background

Many of my clients are companies that need a Web application, but aren't familiar with the ways in which the Web works. A common question for them to ask me is "We have many thousands of users every month. Can the server handle that many people?" When I explain that users consume server resources only when they actively are making an HTTP request, their understanding begins to improve. A company with 10,000 visitors a month doesn't need to worry about 10,000 simultaneous visitors; they likely will have some periods of time with a few dozen and other periods of time with absolutely none. Thus, scaling up their infrastructure to handle 10,000 simultaneous users would be foolish.

At the same time, there are times—such as after launching an advertising campaign or being mentioned on a TV show—that you indeed will have a

But of course, life is more complicated than that. First and foremost, every system has bottlenecks that can't just be wished away by auto-scaling.

huge spike in traffic. Companies that advertise during the Super Bowl not only expect to get millions of viewers, they also expect to have many of those people visit their Web sites after (or during) watching the ads. This means normal assumptions for scaling no longer are applicable.

This is one of the reasons why Amazon's EC2 has become so popular. If you can treat a server as a commodity, paying for it by the hour and spinning servers up and down as necessary, you can solve this scaling problem. As traffic rises, you add more servers. As it falls, you remove them.

But of course, life is more complicated than that. First and foremost, every system has bottlenecks that can't just be wished away by auto-scaling. For example, if it turns out that your database can't handle a large load and you have only a single database server, auto-scaling your Web servers may exacerbate the problem, rather than solve it.

Second, although it's nice to imagine infinite budgets for auto-scaling servers, it's probably a bit more realistic to think not just about increasing the number of servers, but also about making each individual server more efficient. If there are ways to improve the efficiency of your code, that's often a good place to work on scaling, before throwing (virtual) hardware at the problem.

Third, if you're in charge of a site's technical infrastructure, your answer to the "how many people can we serve simultaneously" question probably should not be "it's infinite, assuming an infinite budget". The technical staff might like that answer, but the company's CFO might have a bit of an issue with giving the IT department a blank check.

What Is Speed?

Many non-technical people will say "I want to have a fast Web site." From a technical perspective, however, that's not a very useful statement, because it neither differentiates between the different types of speed, nor does it consider the multiple layers involved in a modern Web application, nor does it take into consideration

multiple people and the crunch that comes from a sudden surge of interest in the site.

So, let's consider the many different parts of a Web application and how each of them might affect the speed.

Speed

It's true that networks can have different speeds. In general, people describe this in terms of bandwidth, which doesn't really mean that the electrons (or photons) are moving through the wires (or fibers or air) any faster, but rather that more of them are pushing through, in parallel, at a time. You can think of bandwidth as a straw through which you're trying to drink your favorite cold beverage. Two straws will allow you to drink twice as much at the same time and, thus, drink more quickly, even if the speed with which liquid flows through each straw is the same.

One of the potential problems with using shared servers, and with using a virtual machine on shared hardware, is that the network capacity is being divided among the many users. Think of what would happen if several people were to share your drinking straw from the previous example. Sure, the overall straw might be the same size, but each individual gets less than the full bandwidth. You don't need a virtual machine to see such effects either—just try to run several network-intensive applications on the same computer, and you'll quickly find that they are competing for network resources.

The upshot here is that you want to maximize the bandwidth available to your server. This means having your own server—even if it's a VM, you probably don't want it sharing resources with other VMs—and putting different services on different computers.

Latency

This term also has to do with speed, but in a different way from pure bandwidth. Let's say you want to transfer data between two huge servers, so you put a huge, high-speed wire between those networks. You would say that such a network has high bandwidth but also low latency, since the signals would go between the two via a high-speed wire.

Let's now replace that high-speed wire with a satellite link. Suddenly, because it takes time to send the data from one network to another, you have decreased your latency while keeping your bandwidth identical. The network speed hasn't changed, but loading each page now will take significantly longer. One of the major

Many people, even those who have been using the Web for years, don't understand that a single Web page is usually the result of dozens, and sometimes even hundreds, of different files—often from different servers.

considerations of a Web application is latency—of the networks on which the server is running, but also of the application itself. If it takes several seconds for a server to reply, you can say that the application has high latency. This not only frustrates users (who have to wait for a response from the server), but it also means that a larger number of processes are running on the server at the same time, consuming resources. Thus, reducing latency in a Web application is in the best interest of users and of the company.

Client-Side Wait Time

Many people, even those who have been using the Web for years, don't understand that a single Web page is usually the result of dozens, and sometimes even hundreds, of different files—often from different servers. Of course, there's the HTTP response from the Web server, but that might (will) then refer to JavaScript, CSS and static files that might reside in a variety of places. JavaScript is a particularly well known culprit in this arena, in that sites increasingly are downloading JavaScript from such sites as Google Analytics, Optimizely, Facebook and the like.

The problem is that in order to display the complete Web page, your browser needs to retrieve all of those individual pieces. Thus, one delayed image or one delayed CSS file can cause the wait on the user's side to be frustratingly long. Note that this has only partly to do with the bandwidth and latency on the server. If your Web app responds lightning-fast, but tells the user's browser to download a JavaScript file from a very slow server, then from the user's perspective, things might take a very long time.

This means you need to think about performance in new and different ways from what you might have before. It's not enough to push all of the files to the user's browser or to indicate

from which sites the user's browser can retrieve them. You also need to think about where they are loaded. A <script> tag at the top of the page can have very different performance characteristics than at the bottom of the page, since browsers interpret and render tags from top to bottom.

Client-Side Performance

As if all of that weren't enough, it is increasingly becoming an era of rich client-side Web applications. Regardless of whether you're using something as simple as Backbone or as complex as Ember.js, you are writing software that will be executing inside the user's browser. For its first two decades, the Web was highly biased in favor of the server, which also made it easier to scale, profile, debug and improve programs. But, now that programs are running inside different browsers, on users' computers, there is much more to think and worry about.

A very small JavaScript program might allocate lots of memory and/or take a long time to run. Or, a very large JavaScript program might be straightforward, affecting the in-browser performance very little. I've increasingly found my browser to be consuming a huge proportion of my computer's CPU—not because I'm doing so much, but rather because lots of JavaScript is executing there.

What Does This All Mean?

Web development used to seem so simple. You get a domain, set up a server, slap together some software, and you're in business. And indeed, you often still can do that today. But, if you're expecting to get lots of visitors at once, you need to understand the different types of "fast" that you'll need to consider, measure and then optimize.

Next month, I'll dig deeper into each of these types of speed, looking at specific parts of your software that can affect each of them. I'll give some specific suggestions for how you can identify such issues, as well as solve them, particularly if you have some low-hanging fruit.■

Reuven M. Lerner trains companies around the world in Python, PostgreSQL, Git and Ruby. His ebook, "Practice Makes Python", contains 50 of his favorite exercises to sharpen your Python skills. Reuven blogs regularly at http://blog.lerner.co.il and tweets as @reuvenmlerner. Reuven has a PhD in Learning Sciences from Northwestern University, and he lives in Modi'in, Israel, with his wife and three children.

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

12th Annual

HPC for Wall Street - Cloud & Big Data Show and Conference

September 21, 2015 (Monday) **Roosevelt Hotel, NYC**
Madison Ave and 45th St, next to Grand Central Station

Register for the Free Show

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

Free VIP Show Ticket

Capital Markets and Wall Street IT professionals will assemble at this 2015 HPC Show and Conference, Sept. 21.

This 12th Annual HPC networking opportunity will assemble 500 Capital Markets IT professionals at one time and one place in New York on September 21.

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

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

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

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

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

September 2015 Sponsors

IBM, HP, redhat, Micron, Juniper Networks, lenovo, ARISTA, ORACLE, cisco, NYC NO SQL NewSQL

Show Hours: Mon, Sept 21	8:00-4:00
Conference Hours:	8:30-4:50

Media Sponsors

Linux Journal, Linux Magazine, Securities Technology Monitor, datanami, information management, Linux New Media The Pulse of Open Source, Money management executive, Traders Magazine, Integration developer news, HPCwire, WSTA Wall Street Technology Association

Visit: www.flaggmgmt.com/hpc

HPC for Wall Street – Cloud & Big Data September 21, 2015

Register your whole team. Make multiple copies of this form and fax, email or mail today. A free show badge will be mailed to you.

☐ **Free** Show Registration please print

Name _____

Title _____

Company _____

Street _____

City _____ State____ Zip____

Phone (____) _____

FAX (____) _____

Email _____

☐ **Free** Show Registration please print

Name _____

Title _____

Company _____

Street _____

City _____ State____ Zip____

Phone (____) _____

FAX (____) _____

Email _____

3 Ways to Register

Before Sept. 11, Save time and register:

ONLINE: flaggmgmt.com/hpc

FAX: 212-286-0086

After Sept. 11

BRING TO SHOW:
Roosevelt Hotel
Madison Ave and 45th St.
Mezzanine Level

Flagg Management Inc
353 Lexington Ave,
New York, NY 10016
Phone (212) 286-0333
FAX: (212) 286-0086

DAVE TAYLOR

Words from Letter Cubes

Dave looks at the math behind permutations of children's letter blocks to make different words—and it's a lot of possibilities.

I got a great letter from a reader with a puzzle to solve, so let's dig in, shall we? Here's what he wrote:

> Love your column in *Linux Journal*. I've read it for years and learned a lot about shell scripting, but not quite enough to solve a puzzle on my own.
>
> My parents have a set of 12 or so wooden blocks on a shelf in their guest room, and each block's face has a letter or symbol printed on it. For years I've arranged the blocks to spell a new message whenever I visit, and I got to thinking that I should be able to program a script to give me all the possible words that can be spelled concurrently with this set of blocks.
>
> I started with making an array for each block where each position holds a letter. My thought was to go through the set of 12 blocks, and for each block, print the six letters (or a space for non-letters) in about the same way I'd loop through a row/column grid, and then compare the results to a dictionary to find valid words. But my solution would handle only one combination of blocks. How do I account for all possible block combinations?

This is an interesting puzzle, partly because of the number of combinations involved. If we start with the math, each block is going to have six sides, which means that there are 6^{12} possible combinations of letters you can create—a total of 2,176,782,336 different possibilities.

Now, let's assume that you say "Yikes!" and decide you want to constrain yourself to only six-letter words. That's just as crazy, because the first block can be any of the 12, then the second one of the remaining

11, and so on. So, 12*11*10*9*8*7 gives us the number of block combinations (which adds up to 665,280 possibilities). But each block can show any of six sides, so it's many times more—really, not much of a reduction at all.

If we could check 1,000 words/second, it'd still take years to go through every combination. So we need to do something smarter than just a brute-force lookup in the dictionary.

Where we really can reduce the number of possibilities is by deciding that each block has an identical set of letters, so that for each slot in the word, there are only six possible options. That means that we have 6*6*6*6*6*6 possibilities for a six-letter word, or 46,656.

That's better! Let's assume that each block is identical and has the letters e, t, a, o, l and n, which turn out to be the six most common letters in the English language, in order, according to Wikipedia.

Note: making all the blocks have identical letters dramatically reduces the complexity of our search, because now order doesn't count, but real wooden-letter blocks vary, so although this may be a good assumption for this column, it's unlikely to match the actual blocks at your parents' house.

To check for four-letter words that could be constructed from these blocks, we now could do a regular-expression search across a fully enumerated English language dictionary:

```
$ grep -E '^[etaoln][etaoln][etaoln][etaoln]$' dictionary
```

For testing purposes, the dictionary I'm going to use is the "linuxwords" dictionary available on SourceForge.net: http://sourceforge.net/projects/souptonuts/files/souptonuts/dictionary.

The results are:

```
alee
aloe
anal
anon
ante
lane
late
lean
lent
loan
lone
loon
loot
neat
neon
none
noon
note
onto
```

```
tale
tall
teen
tell
tent
toll
tone
tool
```

That's easy enough.

Even better, we can reduce the regular-expression notation and make it much easier to check for longer words by using this regex:

```
^[etaoln]{4}$
```

The {4} notation indicates that the set has to match four times in a row for it to be valid, the ^ matches the beginning of the line, and $ is the end of the line.

If we want to check six-letter words, it's easy enough:

```
$ grep -E '^[etaoln]{6}$'
dictionary
allele
atonal
latent
nettle
talent
tattoo
tenant
$
```

Let's go for broke and try longer words too:

```
$ grep -E '^[etaoln]{7}$'
dictionary
antenna
$ grep -E '^[etaoln]{8}$'
dictionary
annotate
antennae
neonatal
```

We didn't find any matches for 9–12 occurrences, so we probably won't be able to use all the blocks in a single, *Scrabble*-killer word.

The next logical step is to have a map of all letters on all blocks, but then we can see that if "A" is the set of all letters on the first block, "B" on the second, "C" on the third, and so on, that means we need to test for five-letter words against: ABCDE, ABCDF, ABCDG, ABCDH, ABCDI, ..., ABCDL, ABCEDF and so on.

Remember that "A" might be [abcde], B might be [cdefgh], and so on, depending on how the blocks actually are labeled. In short, that's a crazy lot of possibilities, so we're back into something where brute-force simply won't work.

To look at this differently, what if each block had only one

letter on every face. Now we're talking *Scrabble* tiles, where your rack might be the seven letters AEFRGHM. How many words can you make from that?

It turns out that that's a much easier question. One way to figure it out is simply to generate all possible combinations of those letters and test them against the dictionary (a brute-force solution), which would be 7*6*5*4*3*2*1 (7!) possibilities, or about 5000 seven-letter combinations.

For this, we could do a first filter of the dictionary by using the same regular expression:

```
grep -E '^[aefrghm]{7}$' dictionary
```

This reveals that there are in fact two words that match these criteria: grammar and referee. But, that's not right! Why?

Because the regular expression doesn't take into account that once a letter is used, it cannot be used a second time. There's no way to spell "referee" with one e, however much we'd like to have it so.

So now what? Well, I did say that the regular expression was just a starting point, right? So the second step is to check the possibilities identified against the actual pool of available letters.

To do that, let's do something that's possibly non-intuitive. Let's sort the letters in each word so that they're in alphabetical order, then sort the pool of available letters in alphabetical order too. For example, "grammar" becomes "aagmmrr".

How? It turns out the code is easy:

```
$ echo grammar | grep -o . | sort
|tr -d "\n"
```

But, I've run out of space, so stay tuned. We'll come back to this topic next month.

Acknowledgement

Special thanks to my friend Jesse Stay for his help with regular expressions and that last letter-sorting trick.■

Dave Taylor has been hacking shell scripts for more than 30 years—really. He's the author of the popular *Wicked Cool Shell Scripts* (10th anniversary update coming very soon from O'Reilly and NoStarch Press). He can be found on Twitter as @DaveTaylor and more generally at his tech site http://www.AskDaveTaylor.com.

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

KYLE RANKIN

What's New in 3D Printing, Part IV: OctoPrint

Control your 3D printer from anywhere in your house with a Raspberry Pi.

This is the last article in a four-part series on the current state of 3D printing. In the first part, I gave an overall introduction to differences in 3D printing since I wrote my original articles on 3D printing three years ago. The second piece focused on advances in 3D printing hardware, and the third column covered 3D printing software. In that last article, I mentioned one particular piece of 3D printing software, OctoPrint, that I felt warranted its own article, so I am devoting this final article to how to set up and use OctoPrint.

In the past, the process to print a 3D object involved creating or downloading a 3D model in STL format, using slicing software to convert that STL file to the GCODE language your printer understood, and then using other host software that knew how to communicate to the printer (like Pronterface) to load that GCODE and send it to the printer. More recently, there has been software that combines the slicing and host software into one interface (like Cura), and while that's certainly convenient, it also means that the computer you have connected to the 3D printer to control it must stay connected throughout the entire print.

Although many printers these days support loading GCODE onto an SD card for "headless" printing, in my opinion, OctoPrint is an even better approach. OctoPrint combines 3D printer control software with a Web interface so you can control your printer and monitor its progress over the network. Even better, although OctoPrint can run on any Linux machine, the OctoPi distribution is a

customized SD card image designed to run OctoPrint off a Raspberry Pi. Since most geeks tend to have a Raspberry Pi lying around (and if you don't, it's relatively inexpensive compared to a full-fledged computer), this makes for an easy way to control your 3D printer from anywhere in the house.

The Installation

To get started with the installation, you need to download OctoPrint. The official downloads page is at http://octoprint.org/download, and you can find links to the source package and the GitHub repository there. I'm going to assume that the majority of the people who want to set up OctoPrint will do so on a Raspberry Pi though, so my instructions are geared to installing OctoPi. On the download page, you will find links to a few OctoPi mirrors, so choose one and download the most recent OctoPi image in compressed zip form.

OctoPi is based on Raspbian, so once you have the .zip file, installation works like most other Raspberry Pi images you may have worked with in the past. Unzip the file, insert an SD card in your computer, and then use `dd` to write the image to your SD card device:

```
$ sudo dd if=2015-07-02_2015-05-05-octopi-wheezy-0.12.0.img
➥of=/dev/mmcblk0 bs=1M
```

After you write the image to the SD card, check to see if it automatically mounted the drive. If not, eject and re-insert it, and mount it. If you plan to network the Raspberry Pi via Wi-Fi, OctoPi has added a new mechanism as of version 0.12. Edit the octopi-network.txt file on the root of the SD card with your wireless network settings. Save your changes and unmount the SD card. Now you are ready to insert the SD card in your Raspberry Pi and boot it.

When the Raspberry Pi boots and is on the network, you will be able to log in to it over SSH with the same default credentials that Raspbian uses (login "pi", password "raspberry"), but of course, you'll want to use the `passwd` command to change that to something else. As with Raspbian, you can type `sudo raspi-config` to change system settings, and OctoPrint recommends that you use this tool to expand the SD card's filesystem to fill the device.

Web Interface

Once your Raspberry Pi is on the network, you can access the OctoPrint Web interface through http://octopi.local (if your system supports bonjour), or otherwise, just type the host's IP address in your Web browser. You should see an interface

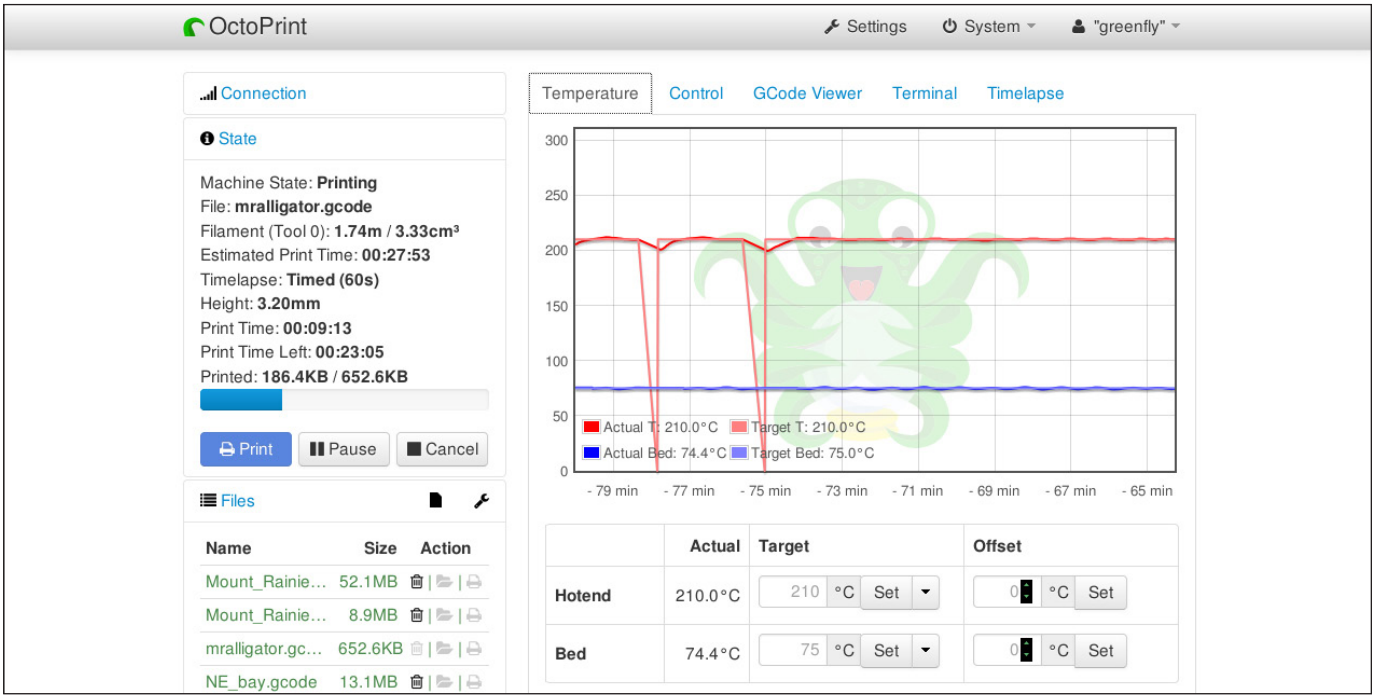


Figure 1. Default OctoPrint Interface

like the one shown in Figure 1. If you haven't yet connected your 3D printer to the Raspberry Pi, this is a good time. On the left side of the Web page is a Connection section that should list reasonable default serial devices it can use to connect to your printer, so you should be able to press the Connect button successfully. If for some reason that doesn't work, you may have to log in to your Raspberry Pi and check the output of `dmesg` to see what sort of device your 3D printer showed up as when you plugged it in.

The default OctoPrint screen shows a graph of your printer's extruder temperature and (if it has one) the heated bed temperature, and it will allow you to set the temperature of either one manually. On the left-hand side of that main page, you can upload GCODE files you have created on Cura or some other slicing tool, and then you can click the printer icon to start printing. This main page also lets you pause and cancel jobs. The pause feature is useful if you need to swap filament in the middle of a print, either because you ran out or because you want to blend multiple colors.

The second tab on OctoPrint provides a Web-based control panel that you can use for manual control of the X, Y and Z axes of your printer as well as extrude plastic. If OctoPrint detects a USB Webcam or the

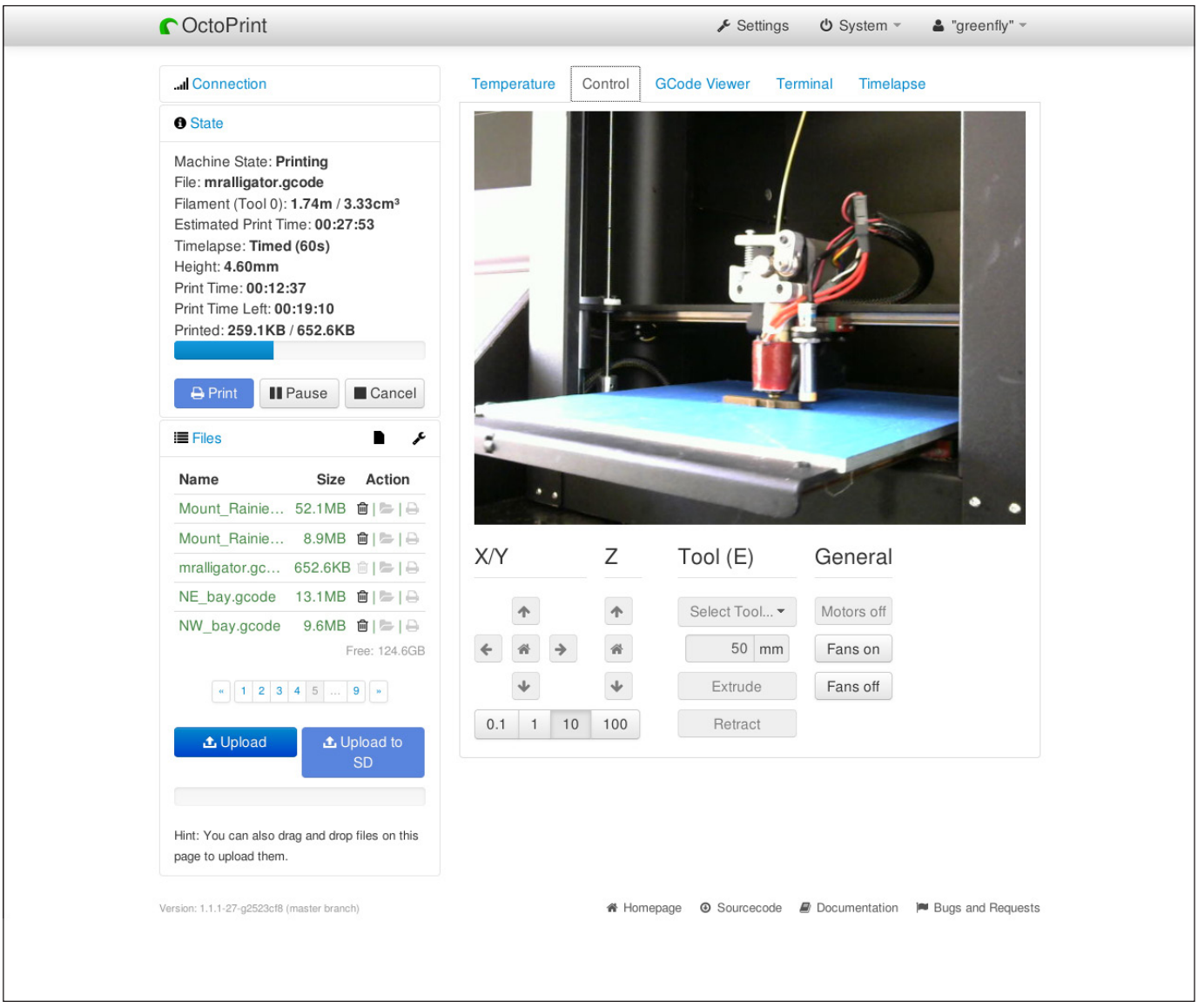


Figure 2. OctoPrint Control Interface

Raspberry Pi camera, it automatically will start MJPG-streamer, and you should be able to see live video from your Webcam (Figure 2). This is pretty useful when you want to start a print from some other room of the house or if you want to monitor a print in progress. If you do have a camera aimed at your printer, you also can click on the Timelapse tab and configure OctoPrint to create a time-lapse video of your print. It can either take a snapshot every few seconds or it can take a shot every time the Z axis changes (that is, when the printer moves to the next layer).

The GCode Viewer tab lets you view the current GCODE being sent

All in all, even though Cura is great for 3D printer control, I find I typically use OctoPrint for all of my 3D printing.

to the printer in a 2D representation of the print. You either can watch it command by command, or you can use a slider on this screen to inspect the tool path on each layer of your print. I've used this feature in the past when I wanted to create a multi-color print of a topographic map in order to identify the specific layer where the GCODE changed from sea level to ground level, so I could pause it and swap blue plastic to another color.

Finally, the Terminal tab is handy, as most 3D printers use GCODE commands to tweak calibration settings. For instance, my Printrbot has a Z axis probe it uses to level the print bed automatically in software; however, you still need to fine-tune the Z axis via a GCODE command. The terminal tab lets you type in those raw GCODE commands and see their output.

All in all, even though Cura is great for 3D printer control, I find I typically use OctoPrint for all of my 3D printing. It's just too nice to be able to start a long print and check up on it from the couch. Plus, the ability to generate time-lapse videos of your prints is hard to beat.

As you can see, an awful lot has changed in 3D printing since my first series of columns three years ago. Although in many cases the changes are negative (like the shift toward proprietary software and hardware), there still are quite a few positive changes from advances in the remaining 3D printers that have open hardware, advances in 3D printing software and more focus on ease of use. I hope you've enjoyed this series on the current state of 3D printing from a Linux perspective. If not, the good news is that next month I'll move on to more traditional topics for Hack and /.■

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

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

Attend

InterDrone

The International Drone Conference and Exposition

InterDrone is Three Awesome Conferences:

Drone TECHCON

For Builders

More than 35 classes, tutorials and panels for hardware and embedded engineers, designers and software developers building commercial drones and the software that controls them.

Drone FLYER

For Flyers and Buyers

More than 35 tutorials and classes on drone operations, flying tips and tricks, range, navigation, payloads, stability, avoiding crashes, power, environmental considerations, which drone is for you, and more!

Drone BUSINESS

For Business Owners, Entrepreneurs & Dealers

Classes will focus on running a drone business, the latest FAA requirements and restrictions, supporting and educating drone buyers, marketing drone services, and where the next hot opportunities are likely to be!

The Largest Commercial Drone Show in North America

Register Today!
Meet with 80+ exhibitors!
Demos! Panels! Keynotes!
The Zipline!

September 9-10-11, 2015
Rio, Las Vegas
www.InterDrone.com

A BZ Media Event

SHAWN POWERS

Pluto and Linux, the Underdog Superheroes

Explore the cosmos without ever leaving your office.

I'm a space nerd. That's probably not a surprise, but just how deep my nerdery goes might be. I have just about every space photo NASA has ever released. I schedule NASA.tv mission briefings on my Google Calendar as if they were specifically for me. I used to make my kids watch Shuttle launches on our TV, even if they were doing homework! (They didn't mind that so much.) Heck, growing up, I wanted to be an astronaut, but since I've worn Coke-bottle thick glasses since I was four years old, I realized I'd never be able to go to space. In fact, I'm a writer today because if I couldn't actually go to space, I wanted to write fiction about space.

I'm not a science fiction writer, although I hope someday that might change. (As a side note, if you haven't read *The Martian* by Andy Weir, you should do so right now. It's possibly my favorite book of all time!) I am, however, still an avid space fan. Thankfully, being a Linux user doesn't hinder me from using technology in order to augment my stargazing. In fact, Linux has an incredible set of tools available for the space nerd, and with the New Horizons probe recently passing Pluto, there's an increase in the number of people looking up at night. So first, I'm going to talk about how Linux can help you appreciate the heavens—then I plan to yammer on about Pluto for a bit. Don't worry, I'll warn you before this article jumps the shark!

The Software

There are four main astronomy packages for the Linux user. Only one of them is Linux-specific, but all work well with every distribution I've tried. There is a lot of overlap between the four programs, but each has its own unique twist as well.

Google Sky: I want to love Google Sky—really. The idea of searching for objects in space using Google is really appealing. Unfortunately, it just never seems to work for me very well. It's perfectly possible that I've never seen how amazing it can be, but since other options exist that are Linux-native, I'm not willing to put too much effort into it.

My experience has been that Google Sky appears to have lots of graphical data, including Hubble imagery. Unfortunately, it lets you zoom in to only a few celestial objects, and in my experience, none of the planets, save Earth. I've had similar luck with Linux using Firefox and Chrome, and I even tried with OS X to similar result. If you're a Google fan and want to search the cosmos like you search for a local pizza joint, perhaps you'll have better luck than me. Check it out at http://sky.google.com.

Kstars: Kstars is the Linux-specific program. And quite honestly, that's not entirely true. Although historically Kstars (part of the KDE suite) has been a Linux application, it does in fact run on multiple platforms. I'd be surprised if it runs as well on Windows as it does on Linux, but it seems only fair to mention the possibility.

And, although I'd like to say the Linux-specific application is my favorite, it's just not true. Kstars does have a boatload of data and more than half a million objects in its database, but the interface is more like a dynamic star map than a stargazing experience itself. That's not a bad thing, and if you're looking for research information or want to see exactly what the sky looks like from your yard, Kstars is great. To re-use my pizza metaphor from earlier though, Kstars is like my least favorite pizza. It's still pizza, and pizza is delicious!

Kstars is available in most distribution's software repositories. It's as easy as an `apt-get` or `yum` away, and there's no reason not to give it a try. The initial setup wizard will ask you for your specific location (Figure 1), and unless you live in a big city, it likely won't be able to find you. In order to get an accurate view of your night sky, you'll need to know your precise latitude and longitude coordinates. (Google should be able to help you find that information.)

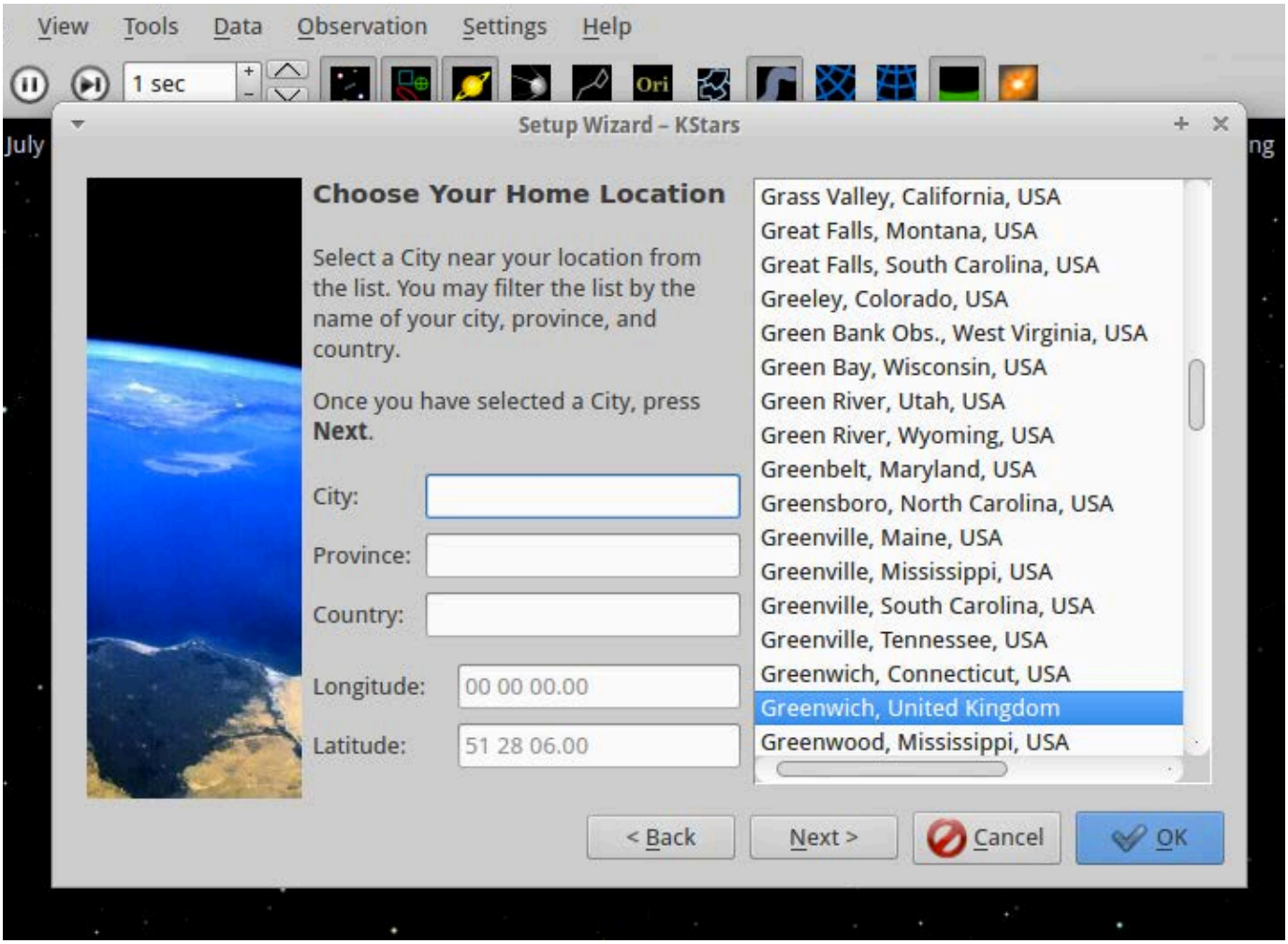


Figure 1. Kstars is a powerful program with loads of data and the ability to import even more.

Celestia: My favorite feature of Celestia (a GTK program) is the ability to "fly" to other planets (Figure 2). Perhaps it's my lingering dreams of being an astronaut, but there's something about flying through the cosmos to a distant planet or galaxy that really appeals to me. Celestia also does time shifting, which allows you to see what the stars and planets will look like, or did look like, on any given day. Although it's not exactly powerful enough to predict a collision with a Near Earth Asteroid, it does make planning for a future evening of stargazing much more fun. Traveling to a Dark Skies park with a Dobsonian light bucket only to discover Saturn has dipped below the horizon can make for a horrible evening. Time shifting is incredibly useful. You can get Celestia for your Linux (or other)

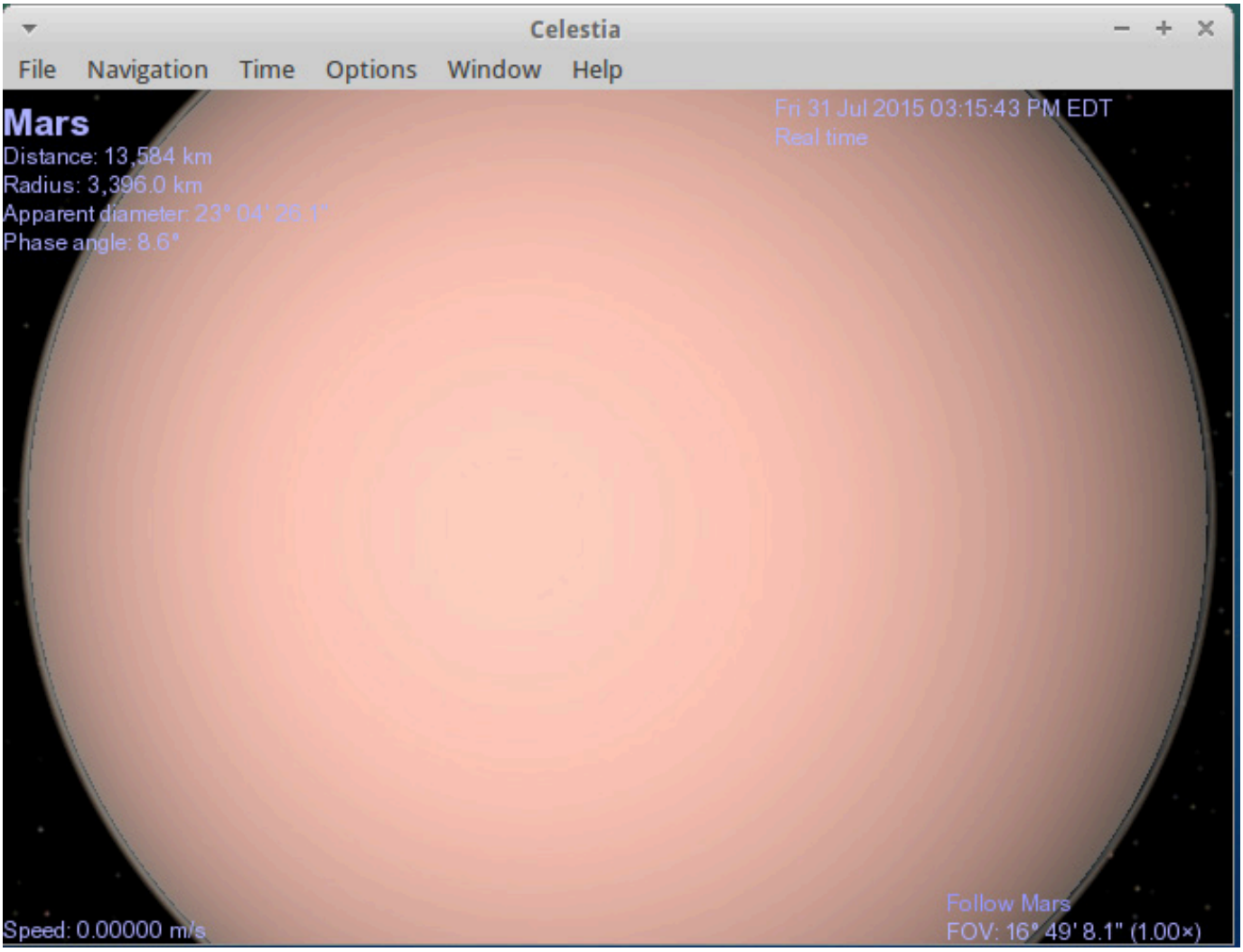


Figure 2. A still photo can't do the "flying" feature justice.

computer in the software repository or at http://www.shatters.net/celestia.

Stellarium: I won't try to hide it, Stellarium is my favorite piece of astronomy software. Celestia wins the wow factor with its "cruise through space" feature, but Stellarium wins for me in every other way—except when it comes to quitting the program. Honestly, I couldn't figure it out for a long time. There's no quit button from what I can tell, and I just couldn't figure out how to exit the program (which starts in full-screen mode, making quitting even more challenging). Let me save you some time: Ctrl-Q. You're welcome.

Some of my favorite features include:

- Graphical Horizon: not only does Stellarium give you the view of the sky from your latitude and

Have you ever wondered what the stars look like from Mars? With Stellarium, you can adjust your point of reference and see the universe from other places *in* the universe.

longitude coordinates, but it also shows you what to expect on the horizon. This includes what it might look like with trees, buildings and so on (Figure 3). Although it's a generic trees/buildings skyline, it does help approximate what you can expect to see. If Jupiter is just over the horizon, you might not see it if there are houses in the way. It's not perfect, but it's a great visualization tool. That said, I think it would be amazing to integrate a feed from Google Street View to see what the actual skyline might look like, but I'm sure there are practical reasons that hasn't been implemented.

- Night Mode: the other programs I mention in this article might have a night mode, but I've never seen the option. With Stellarium, a simple click into night mode will change all the graphics to a red color. If you're a stargazer, you probably know that red lights don't ruin night vision like other colors of light. That means you can take your laptop out with you and use it in the field to identify objects without ruining your adjusted night vision.

- Interplanetary Visitation: okay, I made that term up, but as a wannabe science fiction writer, it's a pretty significant feature. Have you ever wondered what the stars look like from Mars? With Stellarium, you can adjust your point of reference and see the universe from other places *in* the universe. I can't wait for the data to be updated enough to stand on Pluto and look at Charon with high-resolution images from New Horizons.

- Time Shifting: this isn't unique to Stellarium, but it's a feature I really like. Not only can you travel to a different time in the past or future, but you actually can "fast-forward" time to see the changes that will occur in a pseudo-video of events. This helps young kids see the stars

Figure 3. The horizon feature is incredible and helps identify where you're looking.

move when in real life they appear to be static. (Okay, the stars aren't actually moving around, it's us that is moving, but that's part of the discussion to have with your kids. More learning!)

Stellarium is available in most distributions, or you can head over to http://stellarium.org and download the version for your operating system.

Some nights, especially in the winter, I like to play with Stellarium and stare at the night sky from the confines of my cozy house. A large monitor and a dark room can make Stellarium almost as good as the real thing. Almost.

Mobile Magic

As much as I like computer software for finding and planning stargazing sessions, they're honestly not my

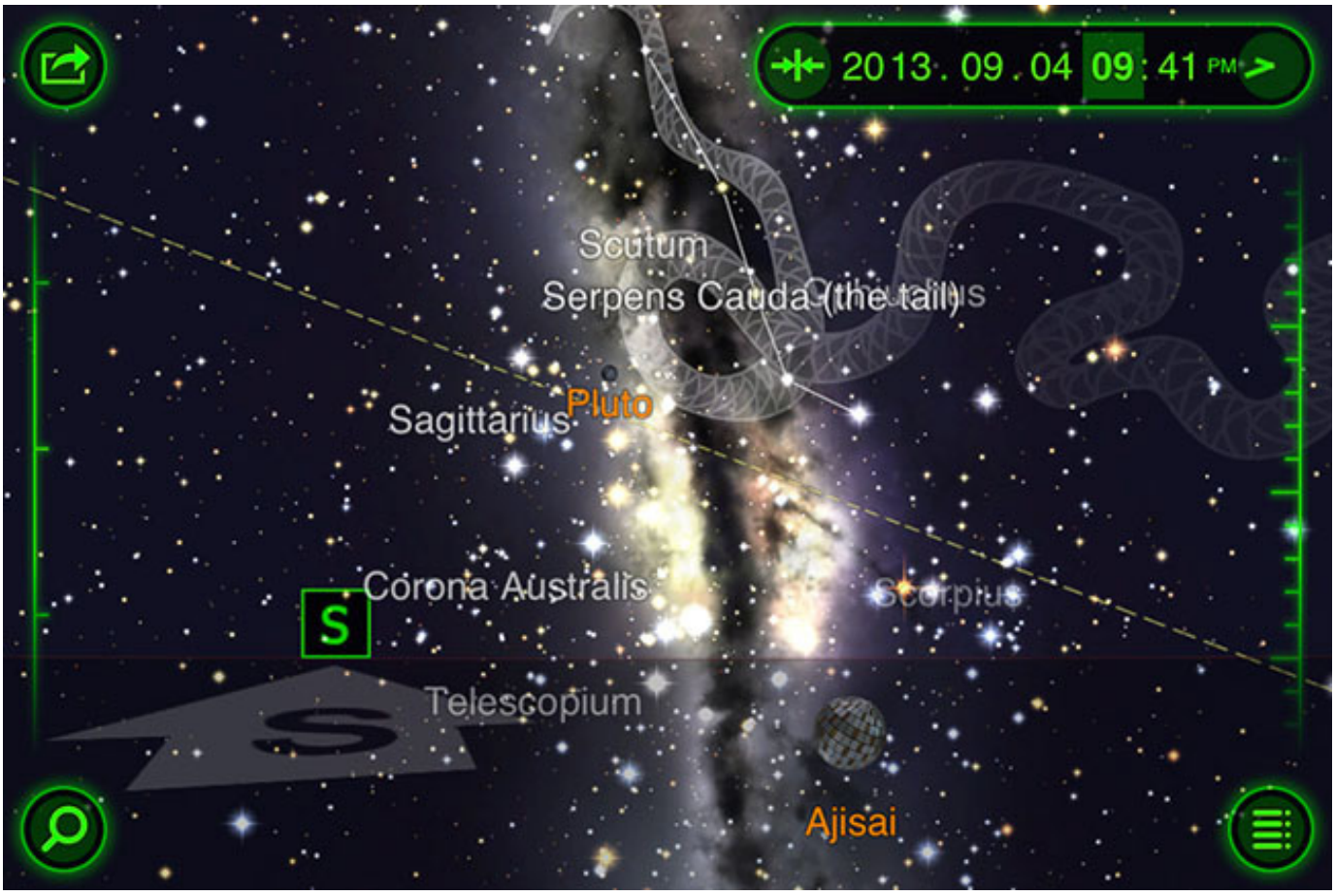


Figure 4. Smartphones have changed the way we stargaze.

go-to way to technologize the cosmos. For that, I tend to whip out my cell phone and point it at the sky. No, I don't try to call E.T.; rather I employ apps like Star Walk (my favorite) to use "augmented reality", which is the coolest thing since telescopes.

With Star Walk (or Sky Map, or Star Chart), you simply hold your phone toward the sky, and using GPS coordinates, motion sensors and digital compass, the app shows you an overlay of what you're pointing at. Want to know what that super-bright star is on the horizon? (It's probably Venus.) Just point your phone at it, and you'll see a nice clear label. Not sure if you're seeing a cluster of stars in the distance or just a smudge on your binoculars? Pinch out to zoom in and look (Figure 4). Truly, smartphones have changed the way nerds like me share excitement about stars and such with others. I've personally never seen Pluto in a telescope, but I've

pointed my phone at the sky knowing I'm looking in the right direction. It's a little like having a GPS device for stargazing. It's incredibly awesome.

You can find Star Walk and the other Android apps I mentioned in the Google Play store. I could give you URLs, but really, I've come to the conclusion that it's easier just to search by name. Star Walk isn't free, but it's only a few dollars and well worth the price of admission. And with that, I must warn you, I'm going to talk about our formerly-a-planet friend, Pluto.

Shawn's Excitement, and Soapbox

Almost exactly in time for my 40th birthday (literally, three days before), the New Horizons probe zoomed past Pluto and its moons at more than 50,000km/hr. There was no way for it to slow down, but the flyby was the goal of the mission, and it succeeded! In a matter of a week, we learned more about Pluto than we've ever

Figure 5. I <3 you too, Pluto!

known in the history of mankind—and that information wasn't what we were expecting at all.

Pluto is so small and cold, we expected to see an ancient, cratered, frozen rock. Instead, we discovered a geologically active world with young mountain ranges and atmospheric attributes never imagined possible. Although most of the world looked at the charming heart-shaped area on the photo (Figure 5), scientists were wondering where the craters were and how on earth mountains formed. See, a lack of craters means the surface is changing and reforming, covering up impact craters. And the mountains mean there has been some sort of geological action like we see on Earth.

We have no idea why—and that's awesome. Jupiter's moon Io is geologically active, but we understand why. With Jupiter's gravity squeezing it, Io has no choice but to spew lava out of volcanoes. If you've ever bent a metal coat hanger back and forth, you know firsthand how friction can cause heat. Jupiter's squeezing literally causes enough friction on Io to make it geologically active. Our own planet has an active, molten core that causes our geologic activity.

Pluto shouldn't have either. Charon isn't large enough to squeeze Pluto, and the dwarf planet is so small it should have cooled off long ago. So why is Pluto geologically active? We don't know! Yay for science!

I won't get too high on my soapbox about how sad I am over the budget slashing our space program has received during the past 50 years. Space exploration is a passion of mine, and until it's a passion for the rest of the world, its budgetary importance will remain low. Although I won't urge you one way or another politically, I will urge you to introduce your spouses, kids, nephews, nieces and grandkids to the cosmos. You can use Linux to make it technological and relevant, and with smartphones, you even can make it relevant to teenagers!

And really, you should read that Andy Weir book. It's incredible!■

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

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

drupalize.me

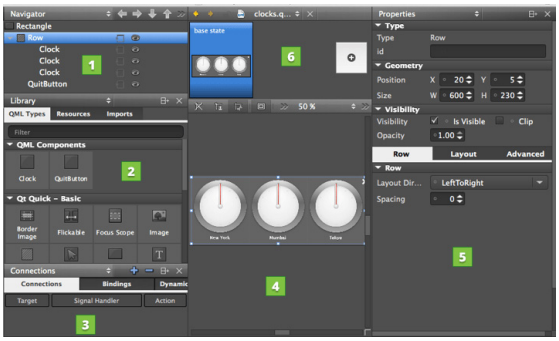
Instant Access to Premium Online Drupal Training

- *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)!

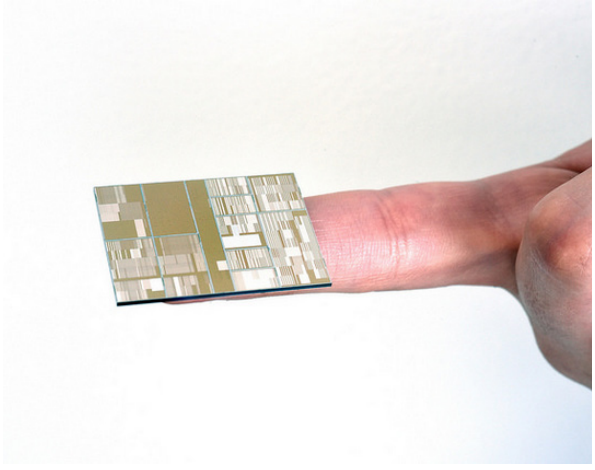
Go to http://drupalize.me and
get Drupalized today!

The Qt Company's Qt



The motto for the Qt Company is simple: "Code less. Create more. Deploy everywhere." It's a sensible leitmotif given that the company made the new Qt 5.5, the upgraded C++-based framework of libraries and tools for developing powerful, interactive and cross-platform applications and devices. Qt's support for multiple desktop, embedded and mobile operating systems allows developers to save significant time on application and device development simply by reusing one code. The most notable innovations in Qt 5.5 are the following: full Bluetooth Low Energy for Internet of Things deployments, a pre-built version of Qt for RHEL 6.6 and preliminary support for upcoming Windows 10 (full subsequent support to follow with a patch release). Other new features include extended support for multimedia and graphics creation with 3D capabilities as well as new multi-screen and IoT development features that strengthen overall performance across applications and devices.

http://www.qt.io/qt5-5



IBM Research Alliance's 7nm Node Chips

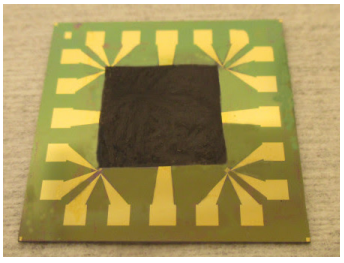
The secret to packing a whopping 20 billion transistors onto a fingernail-sized chip involves a combination of Silicon Germanium (SiGe) channel transistors and Extreme Ultraviolet (EUV) lithography integration. This formula, championed by an alliance led by IBM Research, is billed as the semiconductor industry's first 7nm node chip with functioning transistors. Today, microprocessors leverage 22nm and 14nm technologies, and 10nm is on its way to maturity. The new 7nm technology in the IBM consortium's test chips is considered critical to meeting the anticipated demands of future cloud computing and Big Data systems, cognitive computing, mobile products and other emerging technologies. Other partners in the public-private consortium include GLOBALFOUNDRIES, Samsung and the SUNY Polytechnic Institute's Colleges of Nanoscale Science and Engineering.

http://www.research.ibm.com

Astragon Software & kunst-stoff's *TruckSim*

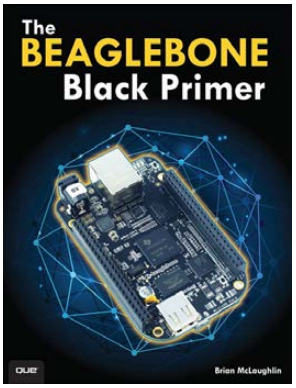
Who knows which side of you will like *TruckSim* best—the truck geek, the sim lover, the Europhile or maybe all three equally! *TruckSim* is a new truck simulation game for Android and iOS devices from the team of game publisher and sim specialist Astragon Software and game developer kunst-stoff. In *TruckSim*, players play the role of a small freight forwarder seeking to build a company and vehicle fleet into a dominating agency by fulfilling steadily bigger delivery orders from customers throughout Europe. Varied missions will lead "simmers" across miles of highway to metropolises all over the continent, each easily recognizable by faithfully re-created landmarks. The biggest attention to detail, however, is of given to the faithfully modeled trucks and trailers in the TGX and TGS model range from popular German maker MAN.
http://www.astragon.de

Chalmers University of Technology's Graphene-Based Cooling Film



Linux Journal's "Future New Products" desk is reporting on a new development out of Sweden's Chalmers University of Technology: a graphene-based film for efficiently cooling electronics that is attachable to components made of silicon. The graphene film has a thermal conductivity capacity that is four times that of copper. Until recently, the methods in place for utilizing graphene for cooling have proven problematic, such as with adhesiveness, when presented with high amounts of heat. This advancement at Chalmers involves the creation of strong covalent bonds between the graphene film and the silicon surface through the addition of property-altering (3-Aminopropyl) triethoxysilane (APTES) molecules. Moreover, functionalization using silane coupling doubles the thermal conductivity of the graphene. While copper has a thermal conductivity value of 401 W/mK, the Chalmers graphene-based solution boasts 1600 W/mK. The results were recently published in the journal *Advanced Functional Materials*.
http://www.chalmers.se/en

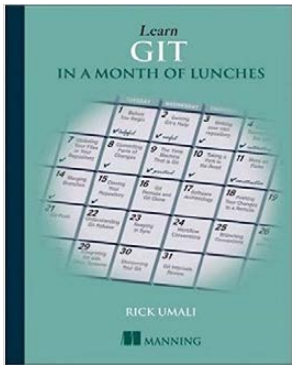
Brian McLaughlin's *The BeagleBone Black Primer* (Que)



If you're looking for a comprehensive, hands-on guide to creating projects on the slick BeagleBone Black embedded development platform, your moment has arrived. Brian McLaughlin, a developer at NASA, has penned said guide: *The BeagleBone Black Primer* from publisher Que. Millions of DIYers, makers, hobbyists and engineers are discovering the fun and utility of BeagleBone Black. Users of the platform can boot a full Linux operating system in less than ten seconds and start developing in less than five minutes with just a single USB cable. Author McLaughlin first reviews the basic embedded programming concepts and techniques that every hardware developer needs to know and then introduces the BeagleBone Black hardware. In the course of the book, McLaughlin guides readers step by step to mastery of increasingly advanced BeagleBone Black programming techniques using the Cloud9 IDE and BoneScript. Throughout, McLaughlin offers both "starter" and "advanced" projects, as well as "jumping-off points" carefully conceived to encourage the reader's own creativity.
http://informit.com

Arduino Srl's Arduino Studio

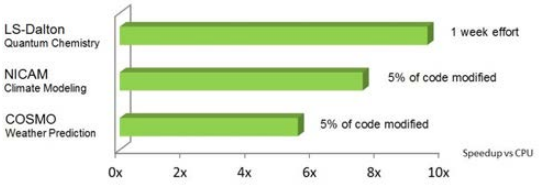
In the dynamic open-source development platform space there is also a "birth" to announce. Arduino Srl (aka Arduino.org), the maker of the popular Arduino microcontroller-based open-source board kits, recently announced the release of a new, bouncing-baby integrated development environment called Arduino Studio. Arduino Studio is a development environment that is completely open source and dedicated to the Arduino programming language. The company describes the IDE as "simple, practical and versatile, enabling [users] to take advantage of the characteristics and potential of Adobe Brackets Editor". Arduino Studio is an IDE for all environments, and it is available for Linux, Windows, Mac OS or as a Bracket extension. Furthermore, the IDE can be utilized not just on standalone PCs, but also in Web/cloud mode, via browser or embedded on Arduino boards. Arduino Studio is open to contributions from the community for new functions, libraries and themes.
http://Arduino.org



Rick Umali's *Learn Git in a Month of Lunches* (Manning Publications Co.)

Make the hour spent stuffing your face with food productive time with Rick Umali's new book *Learn Git in a Month of Lunches*. In this novel "study while dining" format, Umali helps those hungry for knowledge of the diverse and sprawling beast that is Git, the source code control system preferred by distributed development teams. Each easy-to-follow lesson on the discipline of source code control using Git is designed to take an hour or less. The author's goal is to distill Git down to the essential concepts and techniques that users need to get the most out of it with a focus on stuff they'll likely use every day. Rather than cover a shallow introduction to Git's massive surface area, readers will find a road map to the commands and processes that they need to be instantly productive.
http://manning.com/umali

NVIDIA OpenACC Toolkit



It is not that computing cores aren't getting faster. Instead, processors are getting more parallel, which is a trend that is likely to continue. To harness advances in parallel computing, NVIDIA and its partners developed the OpenACC standard, which NVIDIA says "simplifies parallel programming for modern processors, like GPUs". In order to simplify access to OpenACC for researchers, NVIDIA has released the new NVIDIA OpenACC Toolkit, a free, all-in-one suite of OpenACC parallel programming tools. NVIDIA claims that scientists can do "more science, less programming" from the solution, which features "the industry-leading" PGI Accelerator Fortran/C Workstation Compiler Suite for Linux. The compiler is free to academic developers and researchers. The toolkit also includes the NVProf Profiler, which gives guidance on where to add OpenACC "directives"—that is, simple compiler hints to accelerate code, as well as simple, real-world code samples.
http://developer.nvidia.com/openacc

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.

Take Control of Your PC with UEFI Secure Boot

UEFI secure boot often is regarded as a nuisance for Linux users, but you can use it to protect your system by taking control of it. Read on to learn how to do this, sign your own bootloader and protect your whole system with full disk encryption (including the kernel).

GREIG PAUL and JAMES IRVINE

UEFI (Unified Extensible Firmware Interface) is the open, multi-vendor replacement for the aging BIOS standard, which first appeared in IBM computers in 1976. The UEFI standard is extensive, covering the full boot architecture. This article focuses on a single useful but typically overlooked feature of UEFI: secure boot.

Often maligned, you've probably encountered UEFI secure boot only when you disabled it during initial setup of your computer. Indeed, the introduction of secure boot was mired with controversy over Microsoft being in charge of signing third-party operating system code that would boot under a secure boot environment.

In this article, we explore the basics of secure boot and how to take control of it. We describe how to install your own keys and sign your own binaries with those keys. We also show how you can build a single standalone GRUB EFI binary, which will protect your system from tampering, such as cold-boot attacks. Finally, we show how full disk encryption can be used to protect the entire hard disk, including the kernel image (which ordinarily needs to be stored unencrypted).

UEFI Secure Boot

Secure boot is designed to protect a system against malicious code being loaded and executed early in the boot process, before the operating system has been loaded. This is to prevent malicious software from installing a "bootkit" and maintaining control over a computer to mask its presence. If an invalid binary is loaded while secure boot is enabled, the user is alerted, and the system will refuse to boot the tampered binary.

On each boot-up, the UEFI firmware inspects each EFI binary that is loaded and ensures that it has either a valid signature (backed by a locally trusted certificate) or that the binary's checksum is present on an allowed list. It also verifies that the signature or checksum does not appear in the deny list. Lists of trusted certificates or checksums are stored as EFI variables within the non-volatile memory used by the UEFI firmware environment to store settings and configuration data.

Secure boot is designed to allow someone with physical control over a computer to take control of the installed keys. A pre-installed manufacturer PK can be programmatically replaced only by signing it with the existing PK. With physical access to the computer, and access to the UEFI firmware

UEFI Key Overview

The four main EFI variables used for secure boot are shown in Figure a. The Platform Key (often abbreviated to PK) offers full control of the secure boot key hierarchy. The holder of the PK can install a new PK and update the KEK (Key Exchange Key). This is a second key, which either can sign executable EFI binaries directly or be used to sign the db and dbx databases. The db (signature database) variable contains a list of allowed signing certificates or the cryptographic hashes of allowed binaries. The dbx is the inverse of db, and it is used as a blacklist of specific certificates or hashes, which otherwise would have been accepted, but which should not be able to run. Only the KEK and db (shown in green) keys can sign binaries that may boot the system.

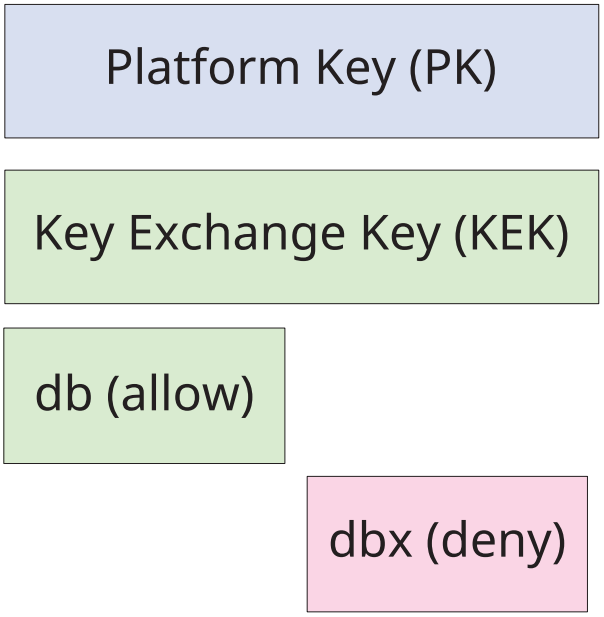


Figure a. Secure Boot Keys

The PK on most systems is issued by the manufacturer of the hardware, while a KEK is held by the operating system vendor (such as Microsoft). Hardware vendors also commonly have their own KEK installed (since multiple KEKs can be present). To take full ownership of a computer using secure boot, you need to replace (at a minimum) the PK and KEK, in order to prevent new keys being installed without your consent. You also should replace the signature database (db) if you want to prevent commercially signed EFI binaries from running on your system.

environment, this key can be removed and a new one installed. Requiring physical access to the system to override the default keys is an important security requirement of secure boot to prevent malicious software from completing this process. Note that some locked-down

ARM-based devices implement UEFI secure boot without the ability to change the pre-installed keys.

Testing Procedure

You can follow these procedures on a physical computer, or alternatively in a virtualized instance of the Intel Tianocore reference UEFI implementation. The `ovmf` package available in most Linux distributions includes this. The QEMU virtualization tool can launch an instance of `ovmf` for experimentation. Note that the `fat` argument specifies that a directory, `storage`, will be presented to the virtualized firmware as a persistent storage volume. Create this directory in the current working directory, and launch QEMU:

```
qemu-system-x86_64 -enable-kvm -net none \
-m 1024 -pflash /usr/share/ovmf/ovmf_x64.bin \
-hda fat:storage/
```

Files present in this folder when starting QEMU will appear as a volume to the virtualized UEFI firmware. Note that files added to it after starting QEMU will not appear in the system—restart QEMU and they will appear. This directory can be used to hold the public keys you want to install to the UEFI firmware, as well as UEFI images to be booted later in the process.

Generating Your Own Keys

Secure boot keys are self-signed 2048-bit RSA keys, in X.509 certificate format. Note that most implementations do not support key lengths greater than 2048 bits at present. You can generate a 2048-bit keypair (with a validity period of 3650 days, or ten years) with the following `openssl` command:

```
openssl req -new -x509 -newkey rsa:2048 -keyout PK.key \
-out PK.crt -days 3650 -subj "/CN=My Secure PK/"
```

The CN subject can be customized as you wish, and its value is not important. The resulting PK.key is a private key, and PK.crt is the corresponding certificate (containing the public key), which you will install into the UEFI firmware shortly. You should store the private key securely on an encrypted storage device in a safe place.

Now you can carry out the same process for both the KEK and for the db key. Note that the db and KEK EFI variables can contain multiple keys (and in the case of db, SHA256 hashes of bootable binaries), although for simplicity, this article considers only storing a single certificate in each. This is more than adequate for taking control of your own computer. Once again, the .key

files are private keys, which should be stored securely, and the .crt files are public certificates to be installed into your UEFI system variables.

Taking Ownership and Installing Keys

Every UEFI firmware interface differs, and it is therefore not possible to provide step-by-step instructions on how to install your own keys. Refer to your motherboard or laptop's instruction manual, or search on-line for the maker of the UEFI firmware. Enter the UEFI firmware interface, usually by holding a key down at boot time, and locate the security menu. Here there should be a section or submenu for secure boot. Change the mode control to "custom" mode. This should allow you to access the key management menus.

At this point, you should make a backup of the UEFI platform keys

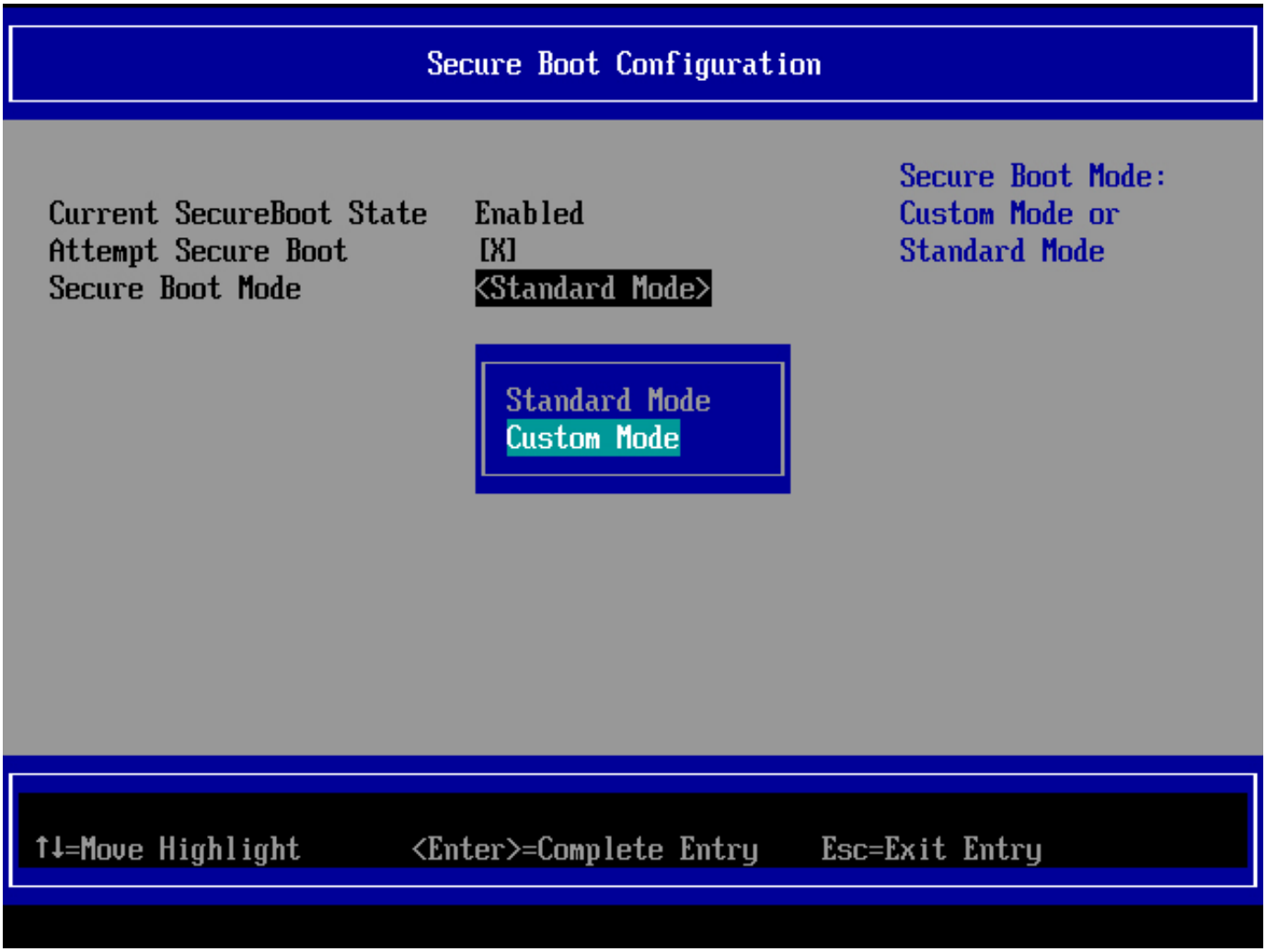


Figure 1. Enabling Secure Boot and Entering Custom Mode

currently installed. You should not need this, since there should be an option within your UEFI firmware interface to restore the default keys, but it does no harm to be cautious. There should be an option to export or save the current keys to a USB Flash drive. It is best to format this with the FAT filesystem if you have any issues with it being detected.

After you have copied the backup keys somewhere safe, load the public certificate (.crt) files you created previously onto the USB Flash drive. Take care not to mix them up with the backup certificates from earlier. Enter the UEFI firmware interface, and use the option to reset or clear all existing secure boot keys.

This also might be referred to as "taking ownership" of secure boot. Your system is now in secure boot "setup" mode, which will remain until a new PK is installed.

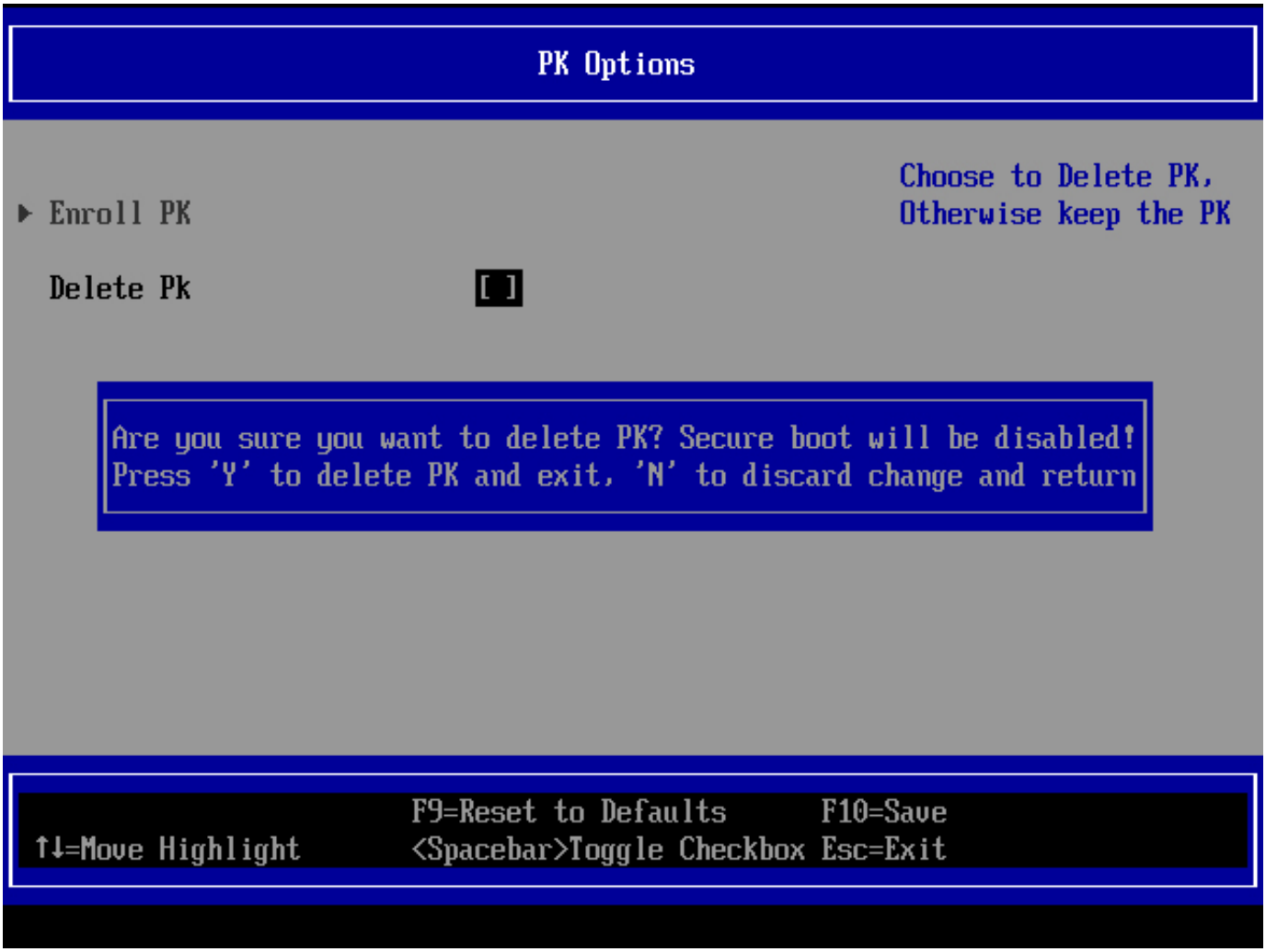


Figure 2. Erasing the Existing Platform Key

Figure 3. Loading a New Key from a Storage Device

At this point, the EFI PK variable is unprotected by the system, and a new value can be loaded in from the UEFI firmware interface or from software running on the computer (such as an operating system).

At this point, you should disable secure boot temporarily, in order to continue following this article. Your newly installed keys will remain in place for when secure boot is enabled.

Signing Binaries

After you have installed your custom UEFI signing keys, you need to sign your own EFI binaries. There are a variety of different ways to build (or obtain) these. Most modern Linux bootloaders are EFI-compatible (for example, GRUB 2, rEFInd or gummiboot), and the Linux kernel itself can be built as a bootable EFI binary since version 3.3. It's possible to sign and boot any valid EFI binary,

although the approach you take here depends on your preference.

One option is to sign the kernel image directly. If your distribution uses a binary kernel, you would need to sign each new kernel update before rebooting your system. If you use a self-compiled kernel, you would need to sign each kernel after building it. This approach, however, requires you to keep on top of kernel updates and sign each image. This can become arduous, especially if you use a rolling-release distribution or test mainline release candidates. An alternative, and the approach we used in this article, is to sign a locked-down UEFI-compatible bootloader (GRUB 2 in the case of this article), and use this to boot various kernels from your system.

Some distributions configure GRUB to validate kernel image signatures against a distribution-specified public key (with which they sign all kernel binaries) and disable editing of the kernel `cmdline` variable when secure boot is in use. You therefore should refer to the documentation for your distribution, as the section on ensuring your boot images are encrypted would not be essential in this case.

The Linux `sbsigntools` package is available from the repositories of most Linux distributions and is a good first port of call when signing UEFI binaries. UEFI secure boot binaries should be signed with an Authenticode-format signature. The command of interest is `sbsign`, which is invoked as follows:

```
sbsign --key DB.key --cert DB.crt unsigned.efi \
--output signed.efi
```

Due to subtle variations in the implementation of the UEFI standards, some systems may reject a correctly signed binary from `sbsign`. The best alternative we found was to use the `osslsigncode` utility, which also generates Authenticode signatures. Although this tool was not specifically intended for use with secure boot, it produces signatures that match the required specification. Since `osslsigncode` does not appear to be commonly included in distribution repositories, you should build it from its source code. The process is relatively straightforward and simply requires running `make`, which will produce the executable binary. If you encounter any issues, ensure you have installed `openssl` and `curl`, which are dependencies of the package. (See Resources for a link to the source code repository.)

Binaries are signed with

`osslsigntool` in a similar manner to `sbsign` (note that the hash is defined as sha256 per the UEFI specification; this should not be altered):

```
osslsigncode -certs DB.crt -key DB.key \
-h sha256 -in unsigned.efi -out signed.efi
```

Booting with UEFI

After you have signed an EFI binary (such as the GRUB bootloader binary), the obvious next step is to test it. Computers using the legacy BIOS boot technology load the initial operating system bootloader from the MBR (master boot record) of the selected boot device. The MBR contains code to load a further (and larger) bootloader held within the disk, which loads the operating system. In contrast, UEFI is designed to allow for more than one bootloader to exist on one drive, without the need for those bootloaders to cooperate or even know the others exist.

Bootable UEFI binaries are located on a storage device (such as a hard disk) within a standard path. The partition containing these binaries is referred to as the EFI System Partition. It has a partition ID of 0xEF00 in gdisk, the GPT-compatible equivalent to fdisk. This partition is conventionally located at the beginning of the filesystem and formatted with a FAT32 filesystem. UEFI-bootable binaries are then stored as files in the EFI/BOOT/ directory.

This signed binary should now boot if it's placed at EFI/BOOT/BOOTX64.EFI within the EFI system partition or an external drive, which is set as the boot device. It is possible to have multiple EFI binaries available on one EFI system partition, which makes it easier to create a multi-boot setup. For that to work however, the UEFI firmware needs a boot entry created in its non-volatile memory. Otherwise, the default filename (BOOTX64.EFI) will be used, if it exists.

To add a new EFI binary to your firmware's list of available binaries, you should use the `efibootmgr` utility. This tool can be found in distribution repositories and often is used automatically by the installers for popular bootloaders, such as GRUB.

At this point, you should re-enable secure boot within your UEFI firmware. To ensure that secure boot is operating correctly, you should attempt to boot an unsigned EFI binary. To do so, you can place a binary (such as an unsigned GRUB EFI binary) at EFI/BOOT/BOOTX64.EFI on a FAT32-formatted USB Flash drive. Use the UEFI firmware interface to set this drive as the current boot drive, and ensure that a security warning

appears, which halts the boot process. You also should verify that an image signed with the default UEFI secure boot keys does not boot—an Ubuntu 12.04 (or newer) CD or bootable USB stick should allow you to verify this. Finally, you should ensure that your self-signed binary boots correctly and without error.

Installing Standalone GRUB

By default, the GRUB bootloader uses a configuration file stored at /boot/grub/grub.cfg. Ordinarily, this file could be edited by anyone able to modify the contents of your /boot partition, either by booting to another OS or by placing your drive in another computer.

Therefore, there would be no real security advantage in signing the GRUB bootloader, since the signed (and verified) bootloader would then load unsigned modules from

Bootloader Security

Prior to the advent of secure boot and UEFI, someone with physical access to a computer was presumed to have full access to it. User passwords could be bypassed by simply adding `init=/bin/bash` to the kernel `cmdline` parameter, and the computer would boot straight up into a root shell, with full access to all files on the system.

Setting up full disk encryption is one way to protect your data from physical attack—if the contents of the hard disk is encrypted, the disk must be decrypted before the system can boot. It is not possible to mount the disk's partitions without the decryption key, so the data is protected.

Another approach is to prevent an attacker from altering the kernel cmdline parameter. This approach is easily bypassed on most computers, however, by installing a new bootloader. This bootloader need not respect the restrictions imposed by the original bootloader. In many cases, replacing the bootloader may prove unnecessary—GRUB and other bootloaders are fully configurable by means of a separate configuration file, which could be edited to bypass security restrictions, such as passwords.

By having GRUB create a single, bootable EFI binary, containing all the necessary modules and configuration files, you no longer need to trust the modules and configuration file of your GRUB binary.

the hard disk and use an unsigned configuration file. By having GRUB create a single, bootable EFI binary, containing all the necessary modules and configuration files, you no longer need to trust the modules and configuration file of your GRUB binary. After signing the GRUB binary, it cannot be modified without secure boot rejecting it and refusing to load. This failure would alert you to someone attempting to compromise your computer by modifying the bootloader.

As mentioned earlier, this step may not be necessary on some distributions, as their GRUB bootloader automatically will enforce similar restrictions and checks on kernels when booted with secure boot enabled. So, this section is intended for those who are not using such a distribution or who wish to implement something similar themselves for learning purposes.

To create a standalone GRUB binary, the `grub-mkstandalone` tool is needed. This tool should be included as part of recent GRUB2 distribution packages:

```
grub-mkstandalone -d /usr/lib/grub/x86_64-efi/ \
-O x86_64-efi --modules="part_gpt part_msdos" \
--fonts="unicode" --locales="en@quot" \
--themes=""  -o "/home/user/grub-standalone.efi" \
"boot/grub/grub.cfg=/boot/grub/grub.cfg"
```

A more detailed explanation of the arguments used here is available on the man page for `grub-mkstandalone`. The significant arguments are `-o`, which specifies the output file to be used, and the final string argument, specifying the path to the current GRUB configuration file. The resulting standalone GRUB binary is directly bootable and contains a memdisk, which holds the configuration file and modules, as well as the configuration file. This

A Licensing Warning

As GRUB 2 is licensed under the GPLv3 (or later), this raises one consideration to be aware of. Although not a consideration for individual users (who simply can install new secure boot keys and boot a modified bootloader), if the GRUB 2 bootloader (or indeed any other GPL-v3-licensed bootloader) was signed with a private signing key, and the distributed computer system was designed to prevent the use of unsigned bootloaders, use of the GPL-v3-licensed software would not be in compliance with the licence. This is a result of the so-called anti-tivo'ization clause of GPLv3, which requires that users be able to install and execute their own modified version of GPLv3 software on a system, without being technically restricted from doing so.

GRUB binary now can be signed and used to boot the system. Note that this process should be repeated when the GRUB configuration file is re-generated, such as after adding a new kernel, changing boot parameters or after adding a new operating system to the list, since the embedded configuration file will be out of date with the regular system one.

Locking Down GRUB

To prevent a malicious user from modifying the kernel cmdline of your system (for example, to point to a different init binary), a GRUB password should be set. GRUB passwords are stored within the configuration file, after being hashed with a cryptographic hashing function. Generate a password hash with the `grub-mkpasswd-pbkdf2` command, which will prompt you to enter a password.

The PBKDF2 function is a slow hash, designed to be computationally intensive and prevent brute-force attacks against the password. Its performance is adjusted using the `-c` parameter, if desired, to slow the process further on a fast computer by carrying out more rounds of PBKDF2. The default is for 10,000 rounds. After copying this password hash, it should be added to your GRUB configuration files (which normally are

located in /etc/grub.d or similar). In the file 40_custom, add the following:

```
set superusers="root"
password_pbkdf2 root <generated password hash>
```

This will create a GRUB superuser account named root, which is able to boot any GRUB entry, edit existing boot items and enter a GRUB console. Without further configuration, this password also will be required to boot the system. If you prefer to have yet another password on boot-up, you can skip the next step. With full disk encryption in use though, there is little need in requiring a password on each boot-up.

To remove the requirement for the superuser password to be entered on a normal boot-up, edit the standard boot menu template (normally /etc/grub.d/10-linux), and locate the line creating a regular menu entry. It should look somewhat similar to this:

```
echo "menuentry '$(echo "$title" | grub_quote)'
➥${CLASS} \$menuentry_id_option
➥'gnulinux-$version-$type-$boot_device_id' {" | sed
➥"s/^/$submenu_indentation/"
```

Change this line by adding the argument `--unrestricted`, before the opening curly bracket. This change tells GRUB that booting this entry does not require a password prompt. Depending on your distribution and GRUB version, the exact contents of the line may differ. The resulting line should be similar to this:

```
echo "menuentry '$(echo "$title" | grub_quote)' ${CLASS}
➥\$menuentry_id_option
➥'gnulinux-$version-$type-$boot_device_id'
➥--unrestricted {" | sed "s/^/$submenu_indentation/"
```

After adding a superuser account and configuring the need (or otherwise) for boot-up passwords, the main GRUB configuration file should be re-generated. The command for this is distribution-specific, but is often `update-grub` or `grub-mkconfig`. The standalone GRUB binary also should be re-generated and tested.

Protecting the Kernel

At this point, you should have a system capable of booting a signed (and password-protected) GRUB bootloader. An adversary without access to your keys would not be able to modify the bootloader or its configuration or modules. Likewise, attackers would not be able to change the parameters passed by the bootloader to the kernel. They could, however, modify your kernel image (by swapping the hard disk

into another computer). This would then be booted by GRUB. Although it is possible for GRUB to verify kernel image signatures, this requires you to re-sign each kernel update.

An alternative approach is to use full disk encryption to protect the full system, including kernel images, the root filesystem and your home directory. This prevents someone from removing your computer's drive and accessing your data or modifying it—without knowing your encryption password, the drive contents will be unreadable (and thus unmodifiable).

Most on-line guides will show full disk encryption but leave a separate, unencrypted /boot partition (which holds the kernel and initrd images) for ease of booting. By only creating a single, encrypted root partition, there won't be an unencrypted kernel or initrd stored on the disk. You can, of course, create a separate boot partition and encrypt it using `dm-crypt` as normal, if you prefer.

The full process of carrying out full disk encryption including the boot partition is worthy of an article in itself, given the various distribution-specific changes necessary. A good starting point, however, is the ArchLinux Wiki (see Resources). The main difference from a conventional encryption setup is the use of the GRUB `GRUB_ENABLE_CRYPTODISK=y` configuration parameter, which tells GRUB to attempt to decrypt an encrypted volume prior to loading the main GRUB menu.

To avoid having to enter the encryption password twice per boot-up, the system's `/etc/crypttab` can be used to decrypt the filesystem with a keyfile automatically. This keyfile then can be included in the (encrypted) initrd of the filesystem (refer to your distribution's documentation to find out how to add this to the initrd, so it will be included each time it is regenerated for a kernel update).

This keyfile should be owned by the root user and does not require any user or group to have read access to it. Likewise, you should give the initrd image (in the boot partition) the same protection to prevent it from being accessed while the system is powered up and the keyfile is being extracted.

Final Considerations

UEFI secure boot allows you to take control over what code can run on your computer. Installing your own keys allows you to prevent malicious people from easily booting their own code on your computer. Combining this with full disk encryption will keep your data protected against

unauthorized access and theft, and prevent an attacker from tricking you into booting a malicious kernel.

As a final step, you should apply a password to your UEFI setup interface, in order to prevent a physical attacker from gaining access to your computer's setup interface and installing their own PK, KEK and db key, as these instructions did. You should be aware, however, that a weakness in your motherboard or laptop's implementation of UEFI could potentially allow this password to be bypassed or removed, and that the ability to re-flash the UEFI firmware through a "rescue mode" on your system could potentially clear NVRAM variables. Nonetheless, by taking control of secure boot and using it to protect your system, you should be better protected against malicious software or those with temporary physical access to your computer.■

Greig Paul is a PhD researcher in the mobile communications group at the University of Strathclyde, Glasgow (UK), where he works on mobile device security and secure data storage.

James Irvine is a Reader in the mobile communications group at the University of Strathclyde, Glasgow (UK), focusing on wireless communication resource management and security.

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

Resources

Information about third-party secure boot keys:
http://mjg59.dreamwidth.org/23400.html

More information about the keys and inner workings of secure boot:
http://blog.hansenpartnership.com/the-meaning-of-all-the-uefi-keys

osslsigncode repository: http://sourceforge.net/projects/osslsigncode

ArchLinux Wiki instructions for fully encrypted systems:
https://wiki.archlinux.org/index.php/Dm-crypt/Encrypting_an_entire_system#Encrypted_boot_partition_.28GRUB.29

Guide for full-disk encryption including kernel image:
http://www.pavelkogan.com/2014/05/23/luks-full-disk-encryption

Fedora Wiki on its secure boot implementation:
https://fedoraproject.org/wiki/Features/SecureBoot

peer1 hosting

Where every interaction matters.

break down
your innovation barriers

power your business to its full potential

When you're presented with new opportunities, you want to focus on turning them into successes, not whether your IT solution can support them.

Peer 1 Hosting powers your business with our wholly owned FastFiber Network™, global footprint, and offers professionally managed public and private cloud solutions that are secure, scalable, and customized for your business.

Unsurpassed performance and reliability help build your business foundation to be rock-solid, ready for high growth, and deliver the fast user experience your customers expect.

Want more on cloud?
Call: 844.855.6655 | go.peer1.com/linux | Vew Cloud Webinar:

Public and Private Cloud | Managed Hosting | Dedicated Hosting | Colocation

CIPHER SECURITY

How to harden TLS and SSH.

CHARLES FISHER

Encryption and secure communications are critical to our life on the Internet. Without the ability to authenticate and preserve secrecy, we cannot engage in commerce, nor can we trust the words of our friends and colleagues.

It comes as some surprise then that insufficient attention has been paid in recent years to strong encryption, and many of our "secure" protocols have been easily broken. The recent Heartbleed, POODLE, CRIME and BEAST exploits put at risk our trust in our networks and in one another.

Gathered here are best-practice approaches to close known exploits and strengthen communication security. These recommendations are by no means the final word on the subject—the goal here is to draw focus upon continuing best practice.

Please note that many governments and jurisdictions have declared encryption illegal, and even where allowed, law enforcement has become increasingly desperate with growing opaque content (see the Resources section for articles on these topics). Ensure that both these techniques and the content that they protect are not overtly illegal.

This article focuses on Oracle Linux versions 5, 6 and 7 and close brethren (Red Hat, CentOS and Scientific Linux). From here forward, I refer to these platforms simply as V5, V6 and V7. Oracle's V7 runs only on the x86_64 platform, so that's this article's primary focus.

These products rightly can be considered defective, in spite of constant vendor patches. The library designers would likely argue that their place is to implement mechanism, not policy, but the resulting products are nonetheless critically flawed. Here is how to fix them.

Strong Ciphers in TLS

The Transport Layer Security (TLS) protocols emerged from the older Secure Sockets Layer (SSL) that originated in the Netscape browser and server software.

It should come as no surprise that SSL must not be used in any context for secure communications. The last version, SSLv3, was rendered completely insecure by the recent POODLE exploit. No version of SSL is safe for secure communications of any kind—the design of the protocol is fatally flawed, and no implementation of it can be secure.

TLS version 1.0 is also no longer safe. The immediate preference for secure communication is the modern TLS version 1.2 protocol, which,

unfortunately, is not (yet) widely used. Despite the lack of popularity, prefer 1.2 if you value security.

Yet, even with TLS version 1.2, there still are a number of important weaknesses that must be addressed to meet current best practice as specified in RFC 7525:

- "Implementations MUST NOT negotiate RC4 cipher suites." The RC4 cipher is enabled by default in many versions of TLS, and it must be disabled explicitly. This specific issue was previously addressed in RFC 7465.

- "Implementations MUST NOT negotiate cipher suites offering less than 112 bits of security, including so-called 'export-level' encryption (which provide 40 or 56 bits of security)." In the days of SSL, the US government forced weak ciphers to be used in encryption products sold or given to foreign nationals. These weak "export" ciphers were created to be easily broken (with sufficient resources). They should have been removed long ago, and they recently have been used in new exploits against TLS.

- "Implementations MUST NOT negotiate SSL version 3." This formalizes our distaste for the entire SSL suite.

- "Implementations SHOULD NOT negotiate TLS version 1.0 (or) 1.1." Prefer TLS 1.2 whenever possible.

There are several implementations of the TLS protocols, and three competing libraries are installed on Oracle Linux systems by default: OpenSSL, NSS and GnuTLS. All of these libraries can provide Apache with TLS for HTTPS. It has been asserted that GnuTLS is of low code quality and unsafe for binary data, so exercise special care with this particular library in critical applications. This article focuses only on OpenSSL, as it is the most widely used.

For TLS cipher hardening under OpenSSL, I turn to Hynek Schlawack's Web site on the subject. He lists the following options for the SSL configuration of the Apache Web server:

```
SSLProtocol ALL -SSLv2 -SSLv3
SSLHonorCipherOrder On
SSLCipherSuite
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
```

This configuration focuses upon the Advanced Encryption Standard (AES)—also known as the Rijndael cipher (as named by the cipher's originators), with 3DES as a fallback for old browsers. Note that 3DES generally is agreed to provide 80 bits of security, and it also is quite slow. These characteristics do not meet the above criteria, but we allow the legacy Data Encryption Standard (Triple-DES) cipher to provide continued access to older browsers.

On an older V5 system (which does not implement TLS 1.1 or 1.2 in OpenSSL), the list of acceptable ciphers is relatively short:

```
$ cat /etc/oracle-release /etc/redhat-release
Oracle Linux Server release 5.11
Red Hat Enterprise Linux Server release 5.11 (Tikanga)

$ openssl ciphers -v
'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'
DHE-RSA-AES256-SHA   SSLv3 Kx=DH  Au=RSA Enc=AES(256)  Mac=SHA1
DHE-RSA-AES128-SHA   SSLv3 Kx=DH  Au=RSA Enc=AES(128)  Mac=SHA1
EDH-RSA-DES-CBC3-SHA SSLv3 Kx=DH  Au=RSA Enc=3DES(168) Mac=SHA1
AES256-SHA           SSLv3 Kx=RSA Au=RSA Enc=AES(256)  Mac=SHA1
AES128-SHA           SSLv3 Kx=RSA Au=RSA Enc=AES(128)  Mac=SHA1
DES-CBC3-SHA         SSLv3 Kx=RSA Au=RSA Enc=3DES(168) Mac=SHA1
```

Note that TLS version 1.1 introduced new defenses against CBC exploits. CBC is used above only with the 3DES cipher, which calls into question the use of 3DES with TLS version 1.0. Removing 3DES and/or enforcing a minimal protocol of TLS version 1.1 might be required if your security concerns are very grave, but this will adversely impact compatibility with older browsers. Banishing CBC on OpenSSL 0.9.8e will leave you with few working ciphers indeed.

On V7, the list of allowed ciphers is considerably longer:

```
$ cat /etc/oracle-release /etc/redhat-release
Oracle Linux Server release 7.1
Red Hat Enterprise Linux Server release 7.1 (Maipo)

$ openssl ciphers -v
'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'
ECDHE-RSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH Au=RSA
 ➥Enc=AESGCM(256) Mac=AEAD
ECDHE-ECDSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH Au=ECDSA
 ➥Enc=AESGCM(256) Mac=AEAD
ECDH-RSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH/RSA Au=ECDH
 ➥Enc=AESGCM(256) Mac=AEAD
ECDH-ECDSA-AES256-GCM-SHA384 TLSv1.2 Kx=ECDH/ECDSA
 ➥Au=ECDH Enc=AESGCM(256) Mac=AEAD
ECDHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH Au=RSA
 ➥Enc=AESGCM(128) Mac=AEAD
ECDHE-ECDSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH Au=ECDSA
 ➥Enc=AESGCM(128) Mac=AEAD
ECDH-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH/RSA Au=ECDH
 ➥Enc=AESGCM(128) Mac=AEAD
```

```
ECDH-ECDSA-AES128-GCM-SHA256 TLSv1.2 Kx=ECDH/ECDSA
 ➥Au=ECDH Enc=AESGCM(128) Mac=AEAD
DHE-RSA-AES256-GCM-SHA384 TLSv1.2 Kx=DH Au=RSA
 ➥Enc=AESGCM(256) Mac=AEAD
DHE-RSA-AES128-GCM-SHA256 TLSv1.2 Kx=DH Au=RSA
 ➥Enc=AESGCM(128) Mac=AEAD
ECDHE-RSA-AES256-SHA384 TLSv1.2 Kx=ECDH Au=RSA
 ➥Enc=AES(256) Mac=SHA384
ECDHE-ECDSA-AES256-SHA384 TLSv1.2 Kx=ECDH Au=ECDSA
 ➥Enc=AES(256) Mac=SHA384
ECDHE-RSA-AES256-SHA     SSLv3 Kx=ECDH  Au=RSA
 ➥Enc=AES(256) Mac=SHA1
ECDHE-ECDSA-AES256-SHA  SSLv3 Kx=ECDH Au=ECDSA
 ➥Enc=AES(256) Mac=SHA1
ECDH-RSA-AES256-SHA384  TLSv1.2 Kx=ECDH/RSA Au=ECDH
 ➥Enc=AES(256) Mac=SHA384
ECDH-ECDSA-AES256-SHA384 TLSv1.2 Kx=ECDH/ECDSA
 ➥Au=ECDH Enc=AES(256) Mac=SHA384
ECDH-RSA-AES256-SHA      SSLv3 Kx=ECDH/RSA Au=ECDH
 ➥Enc=AES(256) Mac=SHA1
ECDH-ECDSA-AES256-SHA   SSLv3 Kx=ECDH/ECDSA Au=ECDH
 ➥Enc=AES(256) Mac=SHA1
DHE-RSA-AES256-SHA256   TLSv1.2 Kx=DH  Au=RSA
 ➥Enc=AES(256) Mac=SHA256
DHE-RSA-AES256-SHA       SSLv3 Kx=DH    Au=RSA
 ➥Enc=AES(256) Mac=SHA1
ECDHE-RSA-AES128-SHA256 TLSv1.2 Kx=ECDH Au=RSA
 ➥Enc=AES(128) Mac=SHA256
ECDHE-ECDSA-AES128-SHA256 TLSv1.2 Kx=ECDH
 ➥Au=ECDSA Enc=AES(128) Mac=SHA256
ECDHE-RSA-AES128-SHA     SSLv3 Kx=ECDH  Au=RSA
 ➥Enc=AES(128) Mac=SHA1
ECDHE-ECDSA-AES128-SHA  SSLv3 Kx=ECDH Au=ECDSA
 ➥Enc=AES(128) Mac=SHA1
ECDH-RSA-AES128-SHA256  TLSv1.2 Kx=ECDH/RSA
 ➥Au=ECDH Enc=AES(128) Mac=SHA256
ECDH-ECDSA-AES128-SHA256 TLSv1.2 Kx=ECDH/ECDSA
 ➥Au=ECDH Enc=AES(128) Mac=SHA256
ECDH-RSA-AES128-SHA      SSLv3 Kx=ECDH/RSA Au=ECDH
 ➥Enc=AES(128) Mac=SHA1
ECDH-ECDSA-AES128-SHA   SSLv3 Kx=ECDH/ECDSA
 ➥Au=ECDH Enc=AES(128) Mac=SHA1
DHE-RSA-AES128-SHA256   TLSv1.2 Kx=DH  Au=RSA
 ➥Enc=AES(128) Mac=SHA256
DHE-RSA-AES128-SHA       SSLv3 Kx=DH    Au=RSA
 ➥Enc=AES(128) Mac=SHA1
ECDHE-RSA-DES-CBC3-SHA  SSLv3 Kx=ECDH  Au=RSA
 ➥Enc=3DES(168) Mac=SHA1
ECDHE-ECDSA-DES-CBC3-SHA SSLv3 Kx=ECDH Au=ECDSA
 ➥Enc=3DES(168) Mac=SHA1
ECDH-RSA-DES-CBC3-SHA   SSLv3 Kx=ECDH/RSA Au=ECDH
 ➥Enc=3DES(168) Mac=SHA1
ECDH-ECDSA-DES-CBC3-SHA SSLv3 Kx=ECDH/ECDSA
 ➥Au=ECDH Enc=3DES(168) Mac=SHA1
EDH-RSA-DES-CBC3-SHA    SSLv3 Kx=DH    Au=RSA
 ➥Enc=3DES(168) Mac=SHA1
AES256-GCM-SHA384       TLSv1.2 Kx=RSA Au=RSA
 ➥Enc=AESGCM(256) Mac=AEAD
AES128-GCM-SHA256       TLSv1.2 Kx=RSA Au=RSA
 ➥Enc=AESGCM(128) Mac=AEAD
AES256-SHA256           TLSv1.2 Kx=RSA Au=RSA
 ➥Enc=AES(256) Mac=SHA256
AES256-SHA              SSLv3 Kx=RSA   Au=RSA
 ➥Enc=AES(256) Mac=SHA1
AES128-SHA256           TLSv1.2 Kx=RSA Au=RSA
 ➥Enc=AES(128) Mac=SHA256
AES128-SHA              SSLv3 Kx=RSA   Au=RSA
 ➥Enc=AES(128) Mac=SHA1
DES-CBC3-SHA            SSLv3 Kx=RSA   Au=RSA
 ➥Enc=3DES(168) Mac=SHA1
```

If possible under your release of Apache, also issue an `SSLCompression Off` directive. Compression should not be used with TLS because of the CRIME attack.

If you have connectivity problems with Web clients, try disabling the Cipher Order directive first. Custom HTTP clients may not fully implement the TLS negotiation, which might be solved by allowing the client to pick the cipher.

The cipher selector above also prevents any exploit of the "Logjam" (weak Diffie-Hellman primes) security flaw that recently has surfaced. If your version of Apache supports an alternate dh-prime configuration, it is recommended that you follow this procedure:

```
openssl dhparam -out /home/httpd/conf/dhparams.pem 2048
```

Then add the following line to your Apache SSL configuration:

```
SSLOpenSSLConfCmd DHParameters "/home/httpd/conf/dhparams.pem"
```

Ensure that you have appropriate permissions on your dhparams.pem file, and note that V5 does not support this configuration.

When you have applied these configuration changes to your Apache Web server, use the SSLlabs.com scan tool to rate your server (see Resources). If you are on an older V5 platform that uses the OpenSSL 0.9.8e release, the grade assigned to your server should be a "B"—your final security grade will be higher if you are on a later release.

It is also important to restart your TLS Web server for key regeneration every day, as is mentioned in the Apache changelog:

> Session ticket creation uses a random key created during web server startup and recreated during restarts. No other key recreation mechanism is available currently. Therefore using session tickets without restarting the web server with an appropriate frequency (e.g. daily) compromises perfect forward secrecy.

This information is not well known, and has been met with some surprise and dismay in the security community: "You see, it turns out that generating fresh RSA keys is a bit costly. So modern web servers don't do it for every single connection. In fact, Apache mod_ssl by default will generate a single export-grade RSA key when the server starts up, and

FOR A PUBLIC MAILSERVER, IT IS IMPORTANT TO BE MORE PERMISSIVE WITH THE ALLOWED CIPHERS TO PREVENT SMTP SESSIONS FROM GOING CLEAR TEXT.

will simply re-use that key for the lifetime of that server" (from http://blog.cryptographyengineering.com/2015/03/attack-of-week-freak-or-factoring-nsa.html).

Note that Hynek Schlawack's site provides configuration instructions for nginx and HAProxy in addition to Apache. Several other applications allow a custom cipher specification—two that I mention here are stunnel and sendmail.

The stunnel "TLS shim" allows clear-text socket applications to be wrapped in TLS encryption transparently. In your stunnel configuration, specify the `cipher=` directive with the above string to force stunnel to best practice. Also, on the V7 platform, supply the `fips=no` directive; otherwise, you will be locked to the TLS version 1 protocol with the message `'sslVersion = TLSv1' is required in FIPS mode.`

The sendmail transport agent has received recent patches to specify ciphers fully. You can add the following options to your sendmail.cf to force best practice ciphers:

```
O CipherList=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
O ServerSSLOptions=+SSL_OP_NO_SSLv2 +SSL_OP_NO_SSLv3
 ➥+SSL_OP_CIPHER_SERVER_PREFERENCE
O ClientSSLOptions=+SSL_OP_NO_SSLv2 +SSL_OP_NO_SSLv3
 ➥+SSL_OP_CIPHER_SERVER_PREFERENCE
```

With these settings, you will see encryption information in your mail logs:

```
May 12 10:17:58 myhost sendmail[1234]: STARTTLS=client,
 ➥relay=mymail.linuxjournal.com., version=TLSv1/SSLv3,
 ➥verify=FAIL, cipher=AES128-SHA, bits=128/128
May 12 10:38:28 myhost sendmail[5678]: STARTTLS=client,
 ➥relay=mymail.linuxjournal.com., version=TLSv1/SSLv3,
 ➥verify=FAIL, cipher=AES128-SHA, bits=128/128
```

The `verify=FAIL` indicates that your keys are not signed by a certificate authority (which is not as important for an SMTP server). The encryption is listed as AES128-SHA.

For a public mailserver, it is important to be more permissive with the allowed ciphers to prevent

SMTP sessions from going clear text. Behind a corporate firewall, however, it is likely better to force strong TLS ciphers more rigorously.

It is also important to apply vendor patches promptly for TLS. It recently was discovered that later TLS versions were using SSLv3 padding functions directly in violation of the standards, rendering the latest versions vulnerable (this was more a concern for NSS than OpenSSL). Prompt patching is a requirement for a secure TLS configuration.

I would like to thank Hynek Schlawack for his contribution to and thoughtful commentary on TLS security.

Strong Ciphers in SSH

It is now well-known that (some) SSH sessions can be decrypted (potentially in real time) by an adversary with sufficient resources. SSH best practice has changed in the years since the protocols were developed, and what was reasonably secure in the past is now entirely unsafe.

The first concern for an SSH administrator is to disable protocol 1 as it is thoroughly broken. Despite a stream of vendor updates, older Linux releases maintain this flawed configuration, requiring the system manager to remove it by hand. Do so by ensuring "Protocol 2" appears in your sshd_config, and all reference to "Protocol 2,1" is deleted. Encouragement also is offered to remove it from client SSH applications as well, in case a server is inaccessible or otherwise overlooked.

For further hardening of Protocol 2 ciphers, I turn to the Stribika SSH Guide. These specifications are for the very latest versions of SSH and directly apply only to Oracle Linux 7.1.

For older versions of SSH, I turn to the Stribika Legacy SSH Guide, which contains relevant configuration details for Oracle Linux 5, 6 and 7.

There are only two recommended sshd_config changes for Oracle Linux 5:

```
Ciphers aes256-ctr,aes192-ctr,aes128-ctr
MACs hmac-ripemd160
```

Unfortunately, the PuTTY suite of SSH client programs for Win32 are incompatible with the `MACs hmac-ripemd160` setting and will not connect to a V5 server when this configuration is implemented. As PuTTY quietly has become a corporate standard, this likely is an insurmountable incompatibility, so most enterprise deployments will implement only the Cipher directive.

Version 0.58 of PuTTY also does not implement the strong AES-CTR ciphers (these appear to have been introduced in the 0.60 release) and likewise will

not connect to an SSH server where they are used exclusively. It is strongly recommended that you implement the Cipher directive, as it removes RC4 (arcfour), which is totally inappropriate for modern SSH. It is not unreasonable to expect corporate clients to run the latest versions of PuTTY, as new releases are trivially easy to install.

Oracle Linux 5 has a role of special importance as it is the underlying OS for the Linux version of the Oracle Exadata architecture (the alternate OS being Solaris). If you are an Exadata customer, confirm with Oracle that you will retain vendor support if you change cipher and protocol settings on a supported Exadata appliance.

V5's default SSH ciphers will be pruned especially hard:

```
$ man sshd_config | col -b | awk "/Ciphers/,/ClientAlive/"

Ciphers

Specifies the ciphers allowed for protocol version 2.
Multiple ciphers must be comma-separated. The
supported ciphers are 3des-cbc, aes128-cbc, aes192-cbc,
aes256-cbc, aes128-ctr, aes192-ctr, aes256-ctr, arcfour128,
arcfour256, arcfour, blowfish-cbc, and cast128-cbc. The
default is

    aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,
    aes128-cbc,3des-cbc,blowfish-cbc,cast128-cbc,aes192-cbc,
    aes256-cbc,arcfour
```

It is possible to install a newer version of OpenSSH on V5, but it is not easy. Attempting to compile the latest release results in the following error:

```
error: OpenSSL >= 0.9.8f required (have "0090802f
➥(OpenSSL 0.9.8e-fips-rhel5 01 Jul 2008)")
```

It is possible to compile OpenSSH without OpenSSL dependencies with the following:

```
--without-openssl Disable use of OpenSSL; use only
➥limited internal crypto **EXPERIMENTAL**
```

Enterprise deployments are likely unwilling to use experimental code, so I won't go into further details. If you obtain binary RPMs for upgrade, ensure that you know how they were produced.

Oracle Linux 7 lacks a few ciphers from the latest releases of SSH and differs only slightly from the recommended settings:

```
HostKey /etc/ssh/ssh_host_rsa_key
Ciphers aes256-gcm@openssh.com,aes128-gcm@openssh.com,
➥aes256-ctr,aes192-ctr,aes128-ctr
KexAlgorithms diffie-hellman-group-exchange-sha256
MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-
➥etm@openssh.com,hmac-ripemd160-etm@openssh.com,
➥umac-128-etm@openssh.com,hmac-sha2-512,hmac-sha2-256,
➥hmac-ripemd160,umac-128@openssh.com
```

DESPITE UNFORTUNATE RECENT EVENTS, MODERN SECURE COMMUNICATION HAS MUCH TO OWE TO THE DATA ENCRYPTION STANDARD AND THOSE WHO WERE INVOLVED IN ITS INTRODUCTION.

Oracle Linux 7.1 can be configured exactly as recommended, including the new ed25519 hostkey:

```
HostKey /etc/ssh/ssh_host_ed25519_key
HostKey /etc/ssh/ssh_host_rsa_key
Ciphers chacha20-poly1305@openssh.com,aes256-
➥gcm@openssh.com,aes128-gcm@openssh.com,
➥aes256-ctr,aes192-ctr,aes128-ctr
KexAlgorithms curve25519-sha256@libssh.org,diffie-
➥hellman-group-exchange-sha256
MACs hmac-sha2-512-etm@openssh.com,hmac-sha2-256-
➥etm@openssh.com,hmac-ripemd160-etm@openssh.com,
➥umac-128-etm@openssh.com,hmac-sha2-512,hmac-sha2-
➥256,hmac-ripemd160,umac-128@openssh.com
```

The Stribika Guide immediately dismisses the 3DES cipher, which is likely reasonable as it is slow and relatively weak, but also goes to some length to criticize the influence of NIST and the NSA. In the long view, this is not entirely fair, as the US government's influence over the field of cryptography has been largely productive. To quote cryptographer Bruce Schneier, "It took the academic community two decades to figure out that the NSA 'tweaks' actually improved the security of DES....DES did more to galvanize the field of cryptanalysis than anything else." Despite unfortunate recent events, modern secure communication has much to owe to the Data Encryption Standard and those who were involved in its introduction.

Stribika levels specific criticism:

> ...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. And ECDSA (or regular DSA for that matter) is just not a

good algorithm, no matter what curve you use.

In any case, there is technical justification for leaving 3DES in TLS, but removing it from SSH—there is a greater financial cost when browsers and customers cannot reach you than when your administrators are inconvenienced by a software standards upgrade.

If you are using ssh-agent with a private key, you can strengthen the encryption of the password on the key using this method documented by Martin Kleppmann with PKCS#8. Here is the procedure summarized from the author:

```
cd ~/.ssh

mv ~/.ssh/id_rsa ~/.ssh/id_rsa.old

openssl pkcs8 -topk8 -v2 des3 -in ~/.ssh/id_rsa.old
➥-out ~/.ssh/id_rsa

chmod 600 ~/.ssh/id_rsa
```

The author estimates that this procedure provides the equivalent benefit of adding two extra characters to the password. It is important to note, however, that the PuTTY agent is not able to read the new format produced here. If you use pagent with PuTTY (or expect to), convert your OpenSSH key to pagent first, then run this procedure, assuming that retention of your key in both formats is allowed. It is likely wise to retain a copy of the original private key on offline media. It is also important to note that this procedure does not add any extra protection from a keylogger.

User SSH keypairs are likely superior to passwords for many aspects of security. SSH servers cannot enforce password standards on remote keys (minimum password length, change frequency, reuse prevention and so on), and there are definite risks in forwarding the ssh-agent that would compromise server security. If you allow your users to authenticate with SSH keypairs that they generate, you should understand how they can be (ab)used.

Finally, be aware that keystroke delay duration can be used as a side channel exploit in SSH via the application of the Viterbi Algorithm. Interactive SSH sessions are more revealing of content than most expect and should be avoided for those with high security requirements. Send batches of `ssh` commands, or implement a bandwidth "fuzzer" in a secondary session on the same channel if an

NEW!

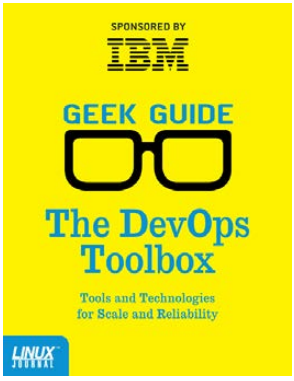
Linux Journal eBook Series

GEEK GUIDES

FREE Download NOW!

The DevOps Toolbox:

Tools and Technologies for Scale and Reliability



By Bill Childers

Introducing *The DevOps Toolbox: Tools and Technologies for Scale and Reliability* by Linux Journal Virtual Editor Bill Childers.

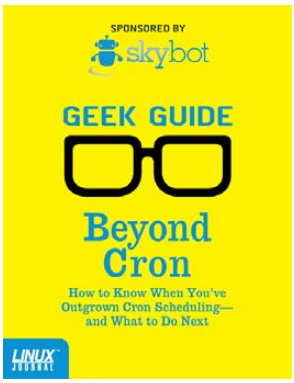
When I was growing up, my father always said, "Work smarter, not harder." Now that I'm an adult, I've found that to be a core concept in my career as a DevOps engineer and manager. In order to work smarter, you've got to have good tools and technology in your corner doing a lot of the repetitive work, so you and your team can handle any exceptions that occur. More important, your tools need to have the ability to evolve and grow over time according to the changing needs of your business and organization.

In this eBook, I discuss a few of the most important tools in the DevOps toolbox, the benefits of using them and some examples of each tool. It's important to not consider this a review of each tool, but rather a guide to foster thinking about what's appropriate for your own organization's needs.

Register today to receive your complimentary copy of The DevOps Toolbox:
http://linuxjournal.com/devops-toolbox-guide

Beyond Cron

How to Know When You've Outgrown Cron Scheduling—and What to Do Next



By Mike Diehl

If you've spent any time around UNIX, you've no doubt learned to use and appreciate cron, the ubiquitous job scheduler that comes with almost every version of UNIX that exists. Cron is simple and easy to use, and most important, it just works. It sure beats having to remember to run your backups by hand, for example.

But cron does have its limits. Today's enterprises are larger, more interdependent, and more interconnected than ever before, and cron just hasn't kept up. These days, virtual servers can spring into existence on demand. There are accounting jobs that have to run after billing jobs have completed, but before the backups run. And, there are enterprises that connect Web servers, databases, and file servers. These enterprises may be in one server room, or they may span several data centers.

Register today to receive your complimentary copy of Beyond Cron:
http://linuxjournal.com/beyond-cron-guide

---

http://linuxjournal.com/geekguides

interactive session is required but security is critical. Of particular note:

- The "superuser" command (that is, `su -`) creates a distinct traffic signature in the encrypted data stream that reveals the exact length of the target password, plus keystroke timing. It should not be used over an SSH session.

- If a user logs in to a remote SSH host, then uses the remote to log in to yet another host in a three-host configuration, this creates an even more distinct traffic signature in the encrypted data stream that essentially advertises the exact length of any passwords used. Avoid this practice.

- Depending upon the cipher used, a short password (less than seven characters) can be detected at login. Enforce a minimum password length larger than seven characters, especially for SSH sessions.

I would like to thank Stribika for his contribution to and thoughtful commentary on SSH security.

Unbreakable Encryption

While the best practices above are helpful, these protocols have been entirely inadequate in assuring private communication channels, and they have been broken many times.

If your needs for secure communication are so dire that any risk of interception is too great, you likely should consider encryption tools that do not appear to have been broken as of yet.

A careful parse of recent evidence indicates that the Gnu Privacy Guard implementation of Pretty Good Privacy (PGP) continues to present insurmountable difficulty to eavesdropping and unauthorized decryption.

This utility is installed in all recent versions of Oracle Linux by default. It should be your first thought for secure communications, and you should realize that all the techniques described above are compromises for the sake of expedience.■

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.

Resources

The Heartbleed Bug: http://heartbleed.com

“Meaner POODLE bug that bypasses TLS crypto bites 10 percent of websites” by Dan Goodin: http://arstechnica.com/security/2014/12/meaner-poodle-bug-that-bypasses-tls-crypto-bites-10-percent-of-websites

CRIME (“Compression Ratio Info-leak Made Easy”): https://en.wikipedia.org/wiki/CRIME

“Beat the BEAST with TLS 1.1/1.2 and More” by Omar Santos: http://blogs.cisco.com/security/beat-the-beast-with-tls

Cypto Law Survey: http://www.cryptolaw.org

“Homeland Security Begs Silicon Valley to Stop the Encyption” by Annalee Newitz: http://gizmodo.com/dhs-secretary-begs-silicon-valley-to-stop-the-encryptio-1699273657

NIST Decprecates TLS 1.0 for Government Use by Bill Shelton: http://forums.juniper.net/t5/Security-Now/NIST-Deprecates-TLS-1-0-for-Government-Use/ba-p/242052

RFC 7525—Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS): https://www.rfc-editor.org/rfc/rfc7525.txt

RFC 7465—Prohibiting RC4 Cipher Suites: http://tools.ietf.org/html/rfc7465

OpenSSL: http://www.openssl.org

NSS: http://nss-crypto.org

The GnuTLS Transport Layer Security Library: http://gnutls.org

GnuTLS considered harmful: http://www.openldap.org/lists/openldap-devel/200802/msg00072.html

Hardening Your Web Server’s SSL Ciphers—Hynek Schlawack: https://hynek.me/articles/hardening-your-web-servers-ssl-ciphers

The Logjam Attack: https://weakdh.org

SSL Labs Scan Tool: https://www.ssllabs.com

Apache changelog: http://www.apache.org/dist/httpd/CHANGES_2.4

“Attack of the week: FREAK (or ’factoring the NSA for fun and profit’)” by Matthew Green: http://blog.cryptographyengineering.com/2015/03/attack-of-week-freak-or-factoring-nsa.html

“The POODLE bites again”: https://www.imperialviolet.org/2014/12/08/poodleagain.html

Stribika SSH Guide: https://stribika.github.io/2015/01/04/secure-secure-shell.html

Stribika Legacy SSH Guide: https://github.com/stribika/stribika.github.io/wiki/Secure-Secure-Shell

“Saluting the data encryption legacy” by Bruce Schneier: http://www.cnet.com/news/saluting-the-data-encryption-legacy

“Improving the security of your SSH private key files” by Martin Kelppmann: http://martin.kleppmann.com/2013/05/24/improving-security-of-ssh-private-keys.html

“Timing Analysis of Keystrokes and Timing Attacks on SSH” by Dawn Xiaodong Song, David Wagner and Xuqing Tian: http://www.cs.berkeley.edu/~dawnsong/papers/ssh-timing.pdf

“The Encryption Tools the NSA Still Can’t Crack Revealed in New Leaks” by Kelsey Campbell: http://gizmodo.com/the-encryption-tools-the-nsa-still-cant-crack-revealed-1675978237

“Prying Eyes: Inside the NSA’s War on Internet Security” by *SPIEGEL* Staff: http://www.spiegel.de/international/germany/inside-the-nsa-s-war-on-internet-security-a-1010361.html

The GNU Privacy Guard: https://gnupg.org

WEBCASTS

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

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

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

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

Sponsor: SAP | Topic: Big Data

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

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

WHITE PAPERS

White Paper: JBoss Enterprise Application Platform for OpenShift Enterprise

Sponsor: DLT Solutions

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

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

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

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

WHITE PAPERS

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

Sponsor: Red Hat | **Topic:** Linux Management

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

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

Standardized Operating Environments for IT Efficiency

Sponsor: Red Hat

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

Benefits of an SOE:

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

SOE leads to:

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

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

DOC SEARLS

The True Internet of Things

An interview with Phil Windley on how to build an IoT that isn't a forest of silos—which is all we have so far.

Before the Internet there were just nets, and they didn't get along. Each was a country or a city-state of its own, with hard boundaries that could not be crossed—or could be crossed only if the owners of the networks created closed and silo'd ways of doing it. Every company's LAN (Local Area Network), for example, had its own way of working, with its own internal e-mail system and methods for transferring files. The same went for "on-line services", such as AOL and Compuserve. Those too were closed and separate worlds of their own. You couldn't send e-mails from one to the other. Many vendors of network "solutions" considered lack of interoperability a feature and not a bug.

The Internet didn't blow existing networks away so much as it simply subordinated them to higher principles that were good for everybody and everything: 1) making every node an end point and 2) getting all the member networks to agree about how to pass data from any end to any other end. Those agreements are protocols, which are simply manners, like nods or handshakes.

That all networks agreed on the same protocols for moving data between end points, with no guards or tolls at border crossings, was as unlikely as instant world peace. Yet, that's exactly what it was—and still is. True, there are countries that censor or restrict the Net and companies that charge us for access to it, but the Net itself is as open and ownerless as gravity and light.

The main effect of the Net in our midst is elimination of distance. This

is new to human experience, but it's already as ordinary, embedded and essential to civilization and its future as were language, fire and the wheel when each of those came along—and just as easy to take for granted. So easy, in fact, that we've forgotten why the Net was such a big success in the first place.

This amnesia is made clear by pretty much everything happening in the ill-named "Internet of Things" (aka IoT) today. Rather than a true Internet of Things—namely, the Internet we already have—we have what Phil Windley calls "The Compuserve of Things" (http://www.windley.com/archives/2014/04/the_compuserve_of_things.shtml). In other words, silos of things. In his summary of that essay, Phil writes:

> On the Net today we face a choice between freedom and captivity, independence and dependence. How we build the Internet of Things has far-reaching consequences for the humans who will use—or be used by—it. Will we push forward, connecting things using forests of silos that are reminiscent of the on-line services of the 1980s, or will we learn the lessons of the Internet and build a true Internet of Things?

Right now the way to bet is on the forest of silos. Every gizmo- and system-maker either has its own silo for things or one inside a farm run by Google, Apple, Facebook or some other large operator. For a clear and depressing look at how this is already ending up badly, read Bruce Sterling's *The Epic Struggle of the Internet of Things* or Cory Doctorow's summary of it (http://boingboing.net/2014/09/13/bruce-sterlings-the-epic-s.html).

For a clear and encouraging look at where this should be going, read Phil Windley (http://www.windley.com). He not only writes eloquently about the IoT (http://www.windley.com/tags/iot.shtml), but he has been working on GPL'd open-source code for things and how they relate (https://github.com/kre/Kinetic-Rules-Engine). To me, Phil is the Linus of IoT—or will be if people jump in and help out with the code. Whether Phil fills that role or not, nobody has more useful or insightful things to say about IoT. That's why I decided to interview him here.

DS: To me, your most important insight is that *things don't need onboard intelligence to be on the Net*. Every thing can have what you call a pico—*a persistent compute*

object (http://www.windley.com/archives/2015/05/picos_persistent_compute_objects.shtml). Earlier you called these "clouds" of their own. What led you to make this leap, which to my knowledge, nobody else has done?

PW: We've made a mistake associating the Internet of Things too closely with "things". To be functional, the IoT will have to operate on a model of the world that doesn't just connect things, but people, places, organizations and even concepts. Everything will have to be on-line. I've been heavily influenced on this by David Gelernter's book *Mirror Worlds* (https://en.wikipedia.org/wiki/Mirror_Worlds). Even though it was written in 1993, pre-Web, it still reads well.

For example, I like to think about what it would take to put every pothole on the Internet of Things. Why? Imagine driving onto your street and seeing a new pothole. Your heads-up display could alert you to its presence, maybe even automatically help your car avoid it. You'd see infographics that tell you when it appeared, who reported it to the city, that city inspectors looked at it last Thursday, and when they were planning to fix it. A city engineer might see a history of potholes on your street, how they related to other infrastructure like water and sewer lines, and so on. The road crew that comes to fix it would see something else entirely.

DS: Interesting. So, if I understand it right, somebody—anybody—could create the first pico for a pothole.

PW: Right. It could be a driver, a resident, a person on a truck from the city, somebody on a crew doing electrical work....

DS: But once the pico is created, and has an OS and storage cloud of its own, it can retain information and interact with different parties. Is that right?

PW: Right.

DS: How will it work so the city engineer, the driver and the road crew see something different in a pothole's pico?

PW: Picos have relationships with other picos or services. The pico can present a different API based on the nature of the relationship. So in the case of the pothole, its pico would naturally have a relationship with the street it's part of. The homeowner and city engineer would also have a

When you start to think of the IoT this way, you realize that there could be trillions of things, and many of them won't need a microprocessor to be part of it.

relationship with the street. The street would introduce the pothole to them, and they'd create a direct relationship. This can all happen automatically and on the fly. The attributes of those relationships would control what the driver's or engineer's pico saw when they queried the pothole.

DS: Meanwhile *all* the conversation and development on IoT is on things with built-in intelligence or data storage. What made you start thinking and working outside that box?

PW: Well, the world is full of things, and only some of them have intelligence. Worse, most of the intelligence being built into things is silo'd to Apple or Google/Nest or some other company—what I call the Compuserve of Things. All this thinking and development is as primitive and isolated and broken and doomed as Compuserve was. Don't get me wrong, it's all useful up to a point, just like Compuserve was. But we didn't iterate from Compuserve to the Internet, and we won't iterate from the current work on IoT to what we really need to build.

To be fully useful and meaningful, IoT should embrace all things, whether or not they have intelligence or memory onboard. Among the infinitude of things in the world are potholes.

Of course, potholes aren't computers, so you can't program them. And we're unlikely, at least for a while, to instrument them. But we could put them on the Internet of Things, as a functional entity with a pico, or something like it. The pico represents the pothole in models, manages relationships to other picos, remembers attributes, history and past interactions, and runs programs for the pothole. The pothole has an API, responds to queries, listens for events and participates as a full-fledged member of the IoT.

When you start to think of the IoT this way, you realize that there could be trillions of things, and many of them won't need a microprocessor

to be part of it.

DS: You also have a kind of operating system for picos called CloudOS, and a language for programming them called KRL. How did those come about?

PW: KRL started off as a domain-specific language for Web page augmentation, and it got away from me. As KRL become more capable, with support for an entity-oriented event bus, event expressions and persistent variables, picos more or less sprung into existence. It was very much a process of discovery.

CloudOS came about one day when Ed Orcutt and I realized that KRL was finally powerful enough to build the functionality we needed for a new project instead of having to put it in the underlying pico hosting platform (KRE). This was a great feeling because it told us we were getting close to having a mature system that could be used for real IoT products.

I have a group of students working on rewriting CloudOS to incorporate the lessons we've learned over the last four years, most notably from using it, and picos, to build the Fuse-connected car system (http://joinfuse.com).

DS: Companies also can give anything they make or sell its own pico, along with its serial number, SKU (stock-keeping unit) or VIN (vehicle identification number). I can imagine lots of ways this can play out, but I'd rather hear where your thinking goes with it.

PW: As cheap as microprocessors are, it's still going to be a while before there's one in everything we buy. In many cases, the business model just can't support it yet. But picos are cheap and easy to create. So it's feasible to give one to every thing right now.

This lets manufacturers, retailers and almost anyone add unique functionality to products that haven't had it before. One obvious, easy-to-imagine scenario is for the product, backed with a pico, to serve as the customer service portal for the owner. The product pico has a relationship with the owner and one with the manufacturer. When there's a problem, the product itself can intermediate the customer support.

But customer support is just a start. Why doesn't my furnace order its own filters, send a text to my teenage son to install them, and then augment his allowance when he's done? Why doesn't my car schedule its own oil changes?

We've always imagined a future world where the device itself is finally smart enough to take care of itself. But we don't need to wait for a smart furnace in our home. With a cloud-based doppelganger in the form of a pico, the furnace could do it today.

The best part is that the architecture puts all of this firmly in the control of the owner, not some cloud owned by the manufacturer. Picos use a hosting model not unlike how Web pages work, so it's easy to move them from the manufacturer's hosting site to your own with no loss of functionality.

DS: How do you see this playing out in the marketplace?

PW: Socially, so to speak. In a blog post on social things, I describe how social things get along with other social things (http://www.windley.com/archives/2015/07/social_things_trustworthy_spaces_and_the_internet_of_things.shtml). And, just as important, how they can protect themselves from things that don't play well in the sandbox.

Imagine a world where things are connected, but unsociable. Every interaction would have to be explicitly scripted or it wouldn't happen. Oh wait, you don't have to imagine it. That's the current model for the IoT, and it won't scale.

On the other hand, if the things we buy are social, not just able to connect to their maker, but also have relationships with other things, people, places and concepts, they can play a variety of roles. As a small example of this, suppose I buy an electric car. The car needs to negotiate charging times with the air conditioner, the home entertainment system and so on. The charging time might change every day. How does the car find these other devices? How does it know which of them to trust? How do they know if they can trust the car? What happens if I want to charge my car at my friend's house?

The number of interaction scenarios for connected devices grows exponentially. We can't script that. It has to come into being on its own. A collection of manufacturer-provided APIs doesn't cut it. Social things act as independent agents to accomplish goals on behalf of their owner.

DS: So, while they might be social things, they're still *your* things. They work for you—not for somebody else.

PW: Right. They know their place in the world, so to speak. That's an important component of sociality.

DS: The IoT, whatever it turns out to be, comes at an interesting time for Linux, because Linux's main problem today is its success. In its first decade, Linux was a cause. In its second decade, it won. Now it's everywhere. I'd like to see the same thing happen with the IoT, based on somebody's code—yours, or anybody's. What can we do to energize development around the early code we already have, and what additional code bases are we going to need?

PW: One big problem is that there are a thousand and one IoT projects. Everyone—and I'm as guilty as anyone else—just wants to work on their own, so none of us get critical mass. Commercial efforts get traction, but aren't likely to produce the true IoT we all want to live with. You've pointed out that the Internet doesn't have a business model. Neither will the IoT.

DS: And we're a long way from it.

PW: Yes. But, we can see the way forward if IoT has a cause. As you say, Linux was a cause—and still is. Here's mine: *we won't get a true Internet of Things until we connect things together in ways that are decentralized, heterarchical and interoperable*. These are key properties of the Internet, and they're sadly lacking in almost every so-called IoT product on the market.

Furthermore, we need an IoT architecture that is biased toward personal control of things without any intervening administrative authority (like the manufacturer, the police, or the Federal Highway Administration, to name a few). What if, in some future world, a root certificate authority revokes the identity documents I use for banking, shopping, travel and a host of other things? Or if my car stops running because Ford shuts off my account? Personal control of the connected things in our lives is fundamental for our freedom.

As a result, I believe the architectural and business model choices we make around IoT will have far-reaching consequences for the humans who will use—or be used by—it. I write about social things and build connected car systems like Fuse because I'm hoping to convince people, a few at a time, that there's an alternative to the Compuserve of Things—one that actually supports freedom.

We can push forward at building forests of silos that are reminiscent the disconnected on-line services

of the 1980s, or we can learn the lessons of the Internet and build a true Internet of Things. That's our choice. And I think it's a cause worth working on.

DS: Thanks!■

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.

Internet Identity Workshop

By the way, Phil and I are two of the organizers of IIW, the Internet Identity Workshop, a three-day unconference that takes place twice a year at the Computer History Museum in Silicon Valley (http://www.internetidentityworkshop.com). It's a great place to come and bang on Phil's code or your own—and to push forward IoT or any other subject that matters to you. The next one is October 27–29, 2015.

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

Advertiser Index

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