

LINUX JOURNAL

Since 1994: The Original Magazine of the Linux Community

SPONSORED BY



JANUARY 2015 | ISSUE 249 | www.linuxjournal.com

SECURITY

PERFORM
AN INTERNAL
SECURITY
REVIEW

PROVIDE ACCESS
CONTROL WITH
SQUID PROXY

TECHNIQUES
FOR SECURING
SERVERS
IN RISKY
ENVIRONMENTS



+

Getting
Started
with
Vagrant

Detect and Block Hackers



WATCH:
ISSUE
OVERVIEW





DIGITAL GUARDIAN®
Formerly VERDASYS

The Only DLP Software Offering LINUX ENDPOINT SUPPORT

“Digital Guardian’s advanced capabilities supporting both Linux and OS X desktops are unique in this market.”

- Gartner

With millions of active agents deployed worldwide, Digital Guardian is the leading platform for data loss prevention. It proactively classifies and tags your most sensitive data and automatically enforces data protection policies on Linux-based servers and endpoints.

Gartner®

DLP Magic Quadrant Leader

Content-Aware DLP Magic Quadrant, 2013



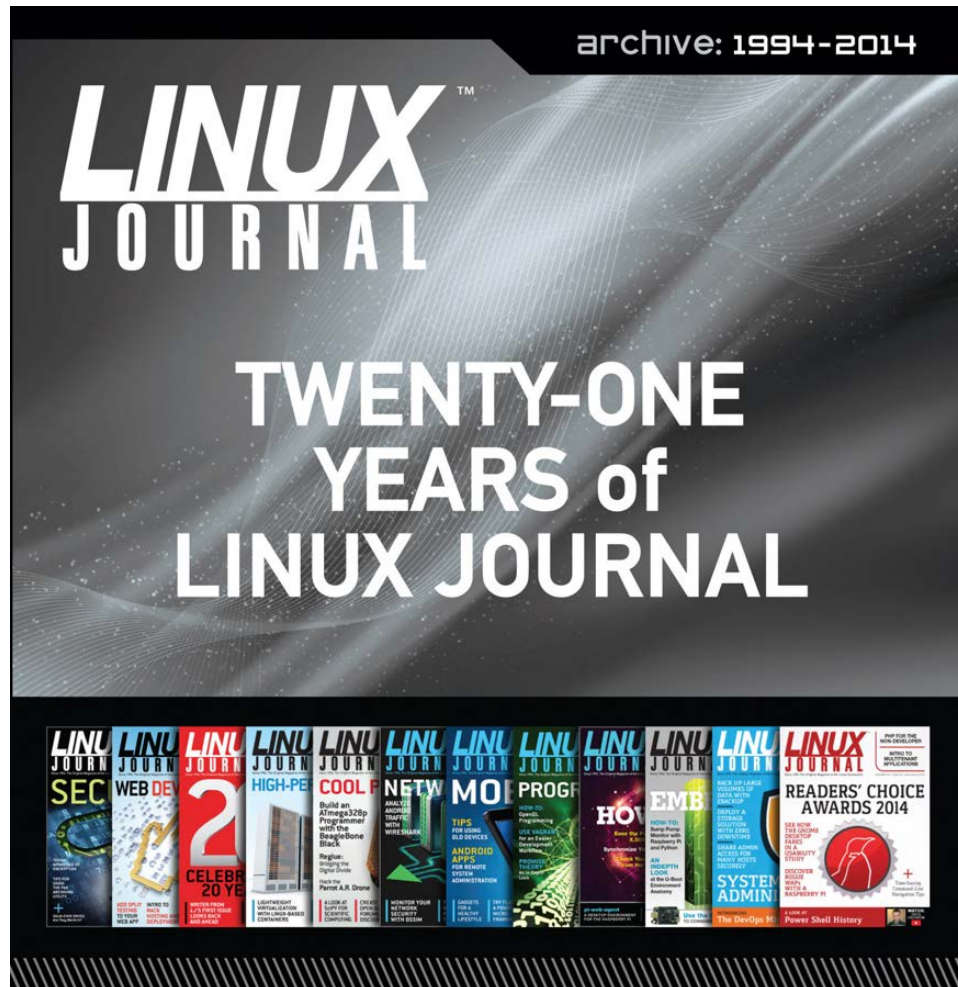
Download a complimentary copy of Gartner’s Content-Aware DLP Magic Quadrant Report.



Visit www.DigitalGuardian.com

LINUX JOURNAL ARCHIVE DVD

1994–2014



NOW AVAILABLE

Save \$10.00 by using discount code DVD2014 at checkout.

Coupon code expires 1/15/2015

www.linuxjournal.com/dvd

CONTENTS

JANUARY 2015
ISSUE 249

FEATURES

64 How to Perform an Internal Security Review

Be prepared.
Be proactive.
Take the time to review.

Jeremiah Bowling

78 Flexible Access Control with Squid Proxy

Database-driven access control for Squid.

Mike Diehl

88 Security in Three Ds: Detect, Decide and Deny

Detect hackers and block them with DenyHosts.

Federico Kereki

Interested in joining our
Reader Advisory Panel for 2015?
Please send a brief e-mail
explaining why you'd be a good
fit to ljeditor@linuxjournal.com.

COLUMNS

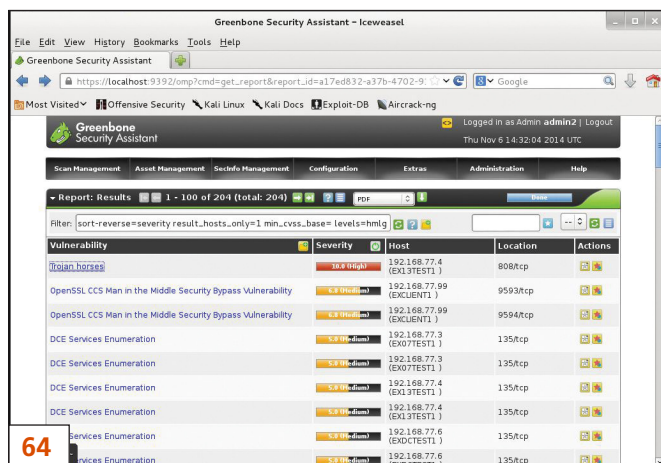
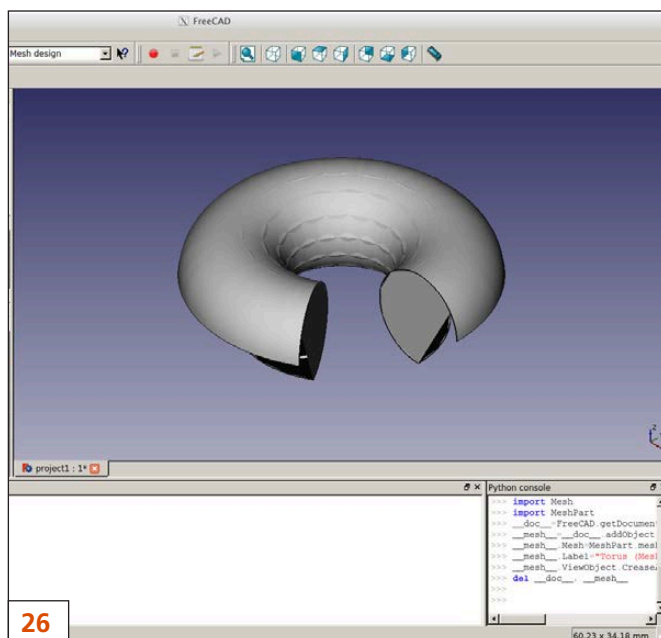
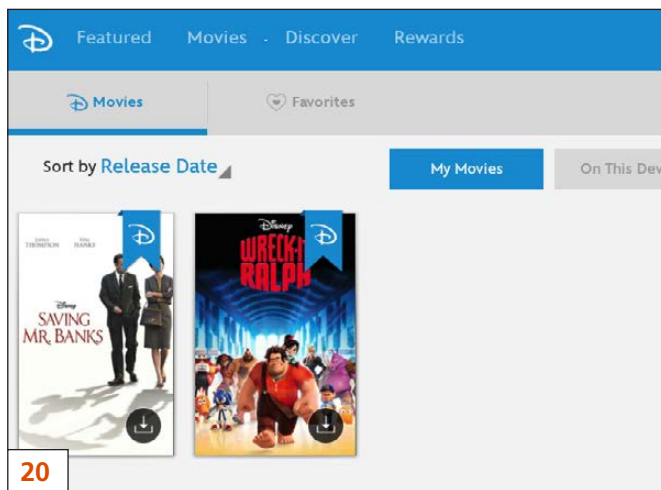
- 34 Reuven M. Lerner's At the Forge**
Users, Permissions and Multitenant Sites
- 44 Dave Taylor's Work the Shell**
The find|xargs Sequence
- 48 Kyle Rankin's Hack and /**
Secure Server Deployments in Hostile Territory
- 52 Shawn Powers' The Open-Source Classroom**
Vagrant Simplified
- 98 Doc Searls' EOF**
Hats Off to Mozilla

IN EVERY ISSUE

- 8 Current_Issue.tar.gz**
- 10 Letters**
- 16 UPFRONT**
- 32 Editors' Choice**
- 60 New Products**
- 101 Advertisers Index**

ON THE COVER

- Perform an Internal Security Review, p. 64
- Provide Access Control with Squid Proxy, p. 78
- Techniques for Securing Servers in Risky Environments, p. 48
- Detect and Block Hackers, p. 88
- Getting Started with Vagrant, p. 52



LINUX JOURNAL™

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



ENJOY:

Timely delivery

Off-line reading

Easy navigation

Phrase search
and highlighting

Ability to save, clip
and share articles

Embedded videos

Android & iOS apps,
desktop and
e-Reader versions

SUBSCRIBE TODAY!

LINUX JOURNAL

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

Contributing Editors

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

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

Editorial Advisory Panel

Nick Baronian
Kalyana Krishna Chadalavada
Brian Conner • Keir Davis
Michael Eager • Victor Gregorio
David A. Lane • Steve Marquez
Dave McAllister • Thomas Quinlan
Chris D. Stark

Advertising

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

Subscriptions

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

LINUX is a registered trademark of Linus Torvalds.



Are you tired of dealing with proprietary storage?

zStax[®]
ZFS Unified Storage
Powered by
NexentaStor

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

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

zStax StorCore 64



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

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

zStax StorCore 104



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

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



SHAWN POWERS

Security: a Method, Not a Goal

The Security issue of *Linux Journal* always makes me feel a little guilty. It turns out that although I have a fairly wide set of technology skills, I'm not the person you want in charge of securing your network or your systems. By default, Linux is designed with a moderate amount of security in mind. For that, I am incredibly grateful. If you struggle with maintaining security in your environment, this issue hopefully will encourage and educate as opposed to making you feel guilty. My goal this year is to learn and be encouraged by the Security issue, not just feel bad. Please, join me!

Reuven M. Lerner starts us out with a continuation on last month's multitenant programming, this time dealing with users and permissions.

With multiple users accessing the same program, security is crucial, and Reuven helps us design intelligently. Dave Taylor follows with a very helpful tutorial on using the `find` command with `xargs`. The `find` command is incredibly powerful, and with the ability to feed it into another program, it's indispensable. Dave walks through not only the how, but the why as well.

Kyle Rankin gets serious about security this month with a practical walk-through on the basics of running a secure server in the cloud. EC2 instances are commonplace in almost every company's infrastructure, but having your server run completely in the open is a dangerous endeavor without a very serious look at security. I go in the opposite direction from Kyle this month and discuss spinning up servers locally. Specifically, I talk about Vagrant. We've covered Vagrant in the past, but it's one of those



VIDEO:
Shawn Powers runs
through the latest issue.

One of the biggest problems with securing a network is knowing where to start.

technologies that always has confused me. This month, I break it down and explain how it works, what it does and how you can get the best use out of it in your environment. If you've ever been frustrated by Vagrant, or just avoided it altogether, I urge you to read my column.

One of the biggest problems with securing a network is knowing where to start. It's a lot easier to figure out that starting point if you know how secure your network right now. Jeramiah Bowling describes the process of doing an internal security review to identify problems. This is great for finding holes in your existing security, but it's also great if you're just starting to create your plan. It's easier to get started when you can find the starting line! Federico Kereki follows Jeramiah with an article on detecting bogus login attempts and mitigating the threat they represent. Having a good password is key to keeping hackers out, but if they have unlimited guesses, eventually your system might succumb to the attacks. Federico shows how to set up a banning system to disable logins when someone tries and fails over and over.

Finally, Mike Diehl has a great tutorial

on securing Web traffic with Squid. Every organization has different needs when it comes to a Web policy, and for Mike, he does the same sort of thing for his home. Whether you're looking to lock down your corporate Internet access, or want to protect your family from various Internet topics, Mike's process is very educational.

Like many things in the Linux world, security isn't a thing you "do", it's a "way" you do things in general. Rather than set up your system and network, and then try to secure it as an afterthought, thinking with a security-focused mindset from the beginning is key. This issue offers some great insight on security matters, and hopefully, it sparks an interest for further change in your network. At the very least, this issue should force you to take a look at your own security practices. As for me? I'm going to read Jeramiah's article and do a security review of my own systems! ■

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](https://freenode.net) IRC channel on Freenode.net.

letters



talking about the PDF version, or if you are referring to all the digital formats. Although the PDF does visually resemble the print magazine more than the other formats, EPUB and Mobi are often the better choice for reading the content, especially on smaller devices. With the Mobi version, it's possible to read Linux Journal on an E Ink Kindle, for instance. Either way, I'm sorry the experience has been unpleasant for you. Hopefully in the future, some combination of hardware and format will bring you back.—Shawn Powers

Renewal

Sorry guys, I just can't hack reading PDF files. When you switched, you effectively stopped me from renewing when my subscription finally ran out. My eyes are worn out from sitting in front of a computer ten hours a day and spending another hour or more reading the PDF just doesn't cut it. I'll go to purchasing a CD every couple years instead, thank you very much. I realize it probably saved the mag and increased revenue, but I am just not cut out to be a subscriber to a PDF. It won't stack up on my shelf and allow me to thumb through it.

—Doug Glenn

I'm not sure if you are specifically

EdgeRouter Lite

It's great that you sing the praises of the EdgeRouter Lite to your readers, but be ready to point them to the community-provided fix when it stops working. [See Shawn Powers' "EdgeRouter Lite" piece in the October 2014 UpFront section.]

Part of the cost-saving has been achieved through using a poor quality USB stick as the Flash memory. Many, including mine, start to die after a year of read/write operations. When it does, it is a simple fix, but you need to get yourself a serial cable, a new USB drive of the correct size and head

over to the community forum for dmbaturin's rescue kit.

Best to be prepared, or you might spend a week without a network while you work out what's happening.

—Harry

I haven't experienced any issues, and as such, haven't had to look for fixes. So thank you very much for pointing me and our fellow readers to the fix! (Open Source communities, we really do make the world a better place.)

—Shawn Powers

I'm Back

I had a subscription for many years to the print edition. I really enjoyed all of the articles, and over all, the journal helped me in my everyday administration of Linux servers in my company. Well, things change, and I let my subscription go.

But, now I'm back. I just subscribed again last night, and I'm going through some of the old editions. While I must admit I do like having a paper version, I find the EPUB editions work very nicely on my Kobo e-reader. I was a little skeptical at first, but since I moved from paper books to

e-books, I thought I would give it a go. Of course, I don't get the pictures in colour, but that's no a big deal.

I just wanted to say thanks for keeping the magazine alive and well. I know digital-only isn't everyone's cup of tea, but it's better than just closing the doors.

—Stephen

Thank you Stephen, and welcome back! I'm still nostalgic for the paper version of Linux Journal, but I'd be lying if I didn't admit the digital versions have some advantages over the dead-tree variety. And whether it's a full-color PDF on a big 10" tablet, or the Mobi version on a Kindle Paperwhite under the covers at night, having Linux Journal with me everywhere is pretty awesome.

—Shawn Powers

Dave Taylor's daysago.sh

I've only just recently been able to catch up on a stack of back issues, and Dave Taylor's efforts to calculate the number of days between dates (see Dave's column in the July to September 2014 issues) has been excruciating to watch, to say the least. There are much more clever ways to go about this.

[LETTERS]

The key is to find a way to map dates onto consecutive integers. Zeller's Congruence (http://en.wikipedia.org/wiki/Zeller%27s_congruence) is one way to do it that has existed since the 1880s; I have implemented a slightly modified version below. Once you have such a function, calculating days between dates or the day of the week for a particular date becomes trivial:

```
#!/bin/sh

# Calculate the number of days to the given date,
# starting from March 1 "year 0"

zeller () {
    year=$1
    month=$2
    day=$3

    # Adjust the year and month so that the "year"
    # starts on March 1, and
    # therefore the "leap day" occurs at the end of
    # the year.
    year=$(( $year + ($month+9)/12 - 1 ))
    month=$(( ($month+9) % 12 ))

    # Calculate the number of days to the
    # start of the given "year"
    leap_val=$(( ($year*365 + $year/4 - $year/100)
        ➔ + $year/400))
    zeller_val=$(( $leap_val + $month*30 +
        ➔ (6*$month+5)/10 + $day + 1))
}

# Get today's date
```

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

# Calculate its zeller value
zeller $thisyear $thismon $thisday
ztoday=$zeller_val
echo Zeller value for $thisyear-$thismon-$thisday
➔ is $ztoday

# Do the same thing for the given date
zeller $1 $2 $3
echo Zeller value for $1-$2-$3 is $zeller_val

# Show the difference
echo difference is $(( $ztoday - $zeller_val )) days
```

—Dave Tweed

Dave Taylor replies: *Sorry that you find my column excruciating, David, but we each know only so much and then just push ahead from there. I've never heard of Zeller's Congruence, but it's definitely interesting. Your code is definitely more efficient than mine.*

Do keep in mind, however, that my column has never been about "the optimal solution for a problem", something that's rarely a shell script at all, but rather "the process of solving a problem within the shell". I'm interested in the journey far more than the destination from a philosophical perspective.

Hope that clarifies things, and thanks again for your sample code.

ZFS and BTRFS

I read Russell Coker's article covering ZFS and BTRFS in the September 2014 issue, but the quoted passage below doesn't make sense:

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

The last sentence is plain wrong. Each ZFS block is limited to a maximum of three copies.

I'm no ZFS expert nor a ZFS developer, but I think I strongly believe the ZFS `copies` property works like this:

- Number of data blocks for ordinary files: $\min(\text{copies}, 3)$.

- Number of data blocks for directories: $\min(\text{copies}+1, 3)$.

- Number of data blocks for metadata: $\min(\text{copies}+2, 3)$.

That is, with `copies=1`, ordinary files are stored once, directories are stored twice, and metadata are stored three times.

With `copies=2`, ordinary files are stored twice, and both directories and metadata are stored three times.

With `copies=3`, all of the ordinary files, directories and metadata, are stored three times.

—Trond Endrestol

Long Live EPUBs

Just a word of thanks, for the interesting content first of all, but more important, for the choice of format that make a real difference to me. It was about a year ago that I was first tempted to try out a subscription to *Linux Journal*, having generally abandoned magazine reading over the past few years while living in a place with no chance of receiving physical media. What finally made me a subscriber was your offer of non-DRM EPUB files. It's now a piece of cake on the first of every

[LETTERS]

month to download the EPUB and transfer it to my e-reader, Android tablet and a backup on my hard drive. It works great and ensures that I have great reading material even where there's no network connection. To those who say the world of magazine publishing has come to an end, a great user experience like this one is a good reminder that it's just beginning! Raising a glass to the *Linux Journal* team for many years of success and an EPUB every month.

—Randall Wood

Thank you for the kind words, Randall!

One of the best things about the switch to digital was the ability to offer several formats, and releasing them to all our readers via e-mail, app and so on. And we respect our readers enough that we wouldn't wrap the issues in DRM—that only causes frustration for the folks we consider family. Plus, it would be a bit strange when I had to write an article on stripping DRM from our own magazine, LOL!—Shawn Powers

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/emailsletters>.

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



Where every interaction matters.

break down your innovation barriers

power your business to its full potential

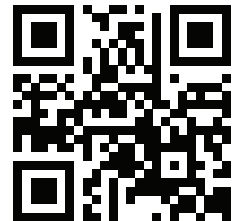
When you're presented with new opportunities, you want to focus on turning them into successes, not whether your IT solution can support them.

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 | [View Cloud Webinar:](#)



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

diff -u

WHAT'S NEW IN KERNEL DEVELOPMENT

David Drysdale wanted to add **Capsicum** security features to Linux after he noticed that **FreeBSD** already had Capsicum support. Capsicum defines fine-grained security privileges, not unlike filesystem capabilities. But as David discovered, Capsicum also has some controversy surrounding it.

Capsicum has been around for a while and was described in a **USENIX** paper in 2010: <http://www.cl.cam.ac.uk/research/security/capsicum/papers/2010usenix-security-capsicum-website.pdf>.

Part of the controversy is just because of the similarity with capabilities. As **Eric Biderman** pointed out during the discussion, it would be possible to implement features approaching Capsicum's as an extension of capabilities, but implementing Capsicum directly would involve creating a whole new (and extensive) abstraction layer in the kernel. Although David argued that capabilities couldn't actually be extended far enough to match Capsicum's fine-grained security controls.

Capsicum also was controversial within its own developer community. For example, as Eric described, it lacked a specification for how to revoke privileges. And, David pointed out that this was because the community couldn't agree on how that could best be done. David quoted an e-mail sent by **Ben Laurie** to the `cl-capsicum-discuss` mailing list in 2011, where Ben said, "It would require additional book-keeping to find and revoke outstanding capabilities, which requires knowing how to reach capabilities, and then whether they are derived from the capability being revoked. It also requires an authorization model for revocation. The former two points mean additional overhead in terms of data structure operations and synchronisation."

Given the ongoing controversy within the Capsicum developer community and the corresponding lack of specification of key features, and given the existence of capabilities that already perform a similar function in the kernel



Wearables TechCon

March 9-11, 2015
Santa Clara, CA

Registration Now Open!



Learn how to design, build and develop apps
for the wearable technology revolution
at **Wearables TechCon 2015!**

Two Huge Technical Tracks

Hardware and Design Track

Choose from 30+ classes on product design, electronic engineering for wearable devices and embedded development. The hardware track is a 360-degree immersion on building and designing the next generation of wearable devices.

Software and App Development Track

Select from 30+ classes on designing software and applications for the hottest wearable platforms. Take deep dives into the leading SDKs, and learn tricks and techniques that will set your wearable software application apart!

- 2 Days of Exhibits
- Business-Critical Panels
- Special Events
- Industry Keynotes

“Wearables DevCon blew away all my expectations, great first year. Words can’t even describe how insightful and motivating the talks were.”

—Mike Diogovanni, Emerging Technology
Lead, Isobar



and the invasiveness of Capsicum patches, Eric was opposed to David implementing Capsicum in Linux.

But, given the fact that capabilities are much coarser-grained than Capsicum's security features, to the point that capabilities can't really be extended far enough to mimic Capsicum's features, and given that FreeBSD already has Capsicum implemented in its kernel, showing that it can be done and that people might want it, it seems there will remain a lot of folks interested in getting Capsicum into the Linux kernel.

Sometimes it's unclear whether there's a bug in the code or just a bug in the written specification. **Henrique de Moraes Holschuh** noticed that the **Intel Software Developer Manual** (vol. 3A, section 9.11.6) said quite clearly that **microcode** updates required 16-byte alignment for the **P6 family of CPUs**, the **Pentium 4** and the **Xeon**. But, the code in the kernel's microcode driver didn't enforce that alignment.

In fact, Henrique's investigation uncovered the fact that some Intel chips, like the Xeon X5550 and the second-generation **i5** chips, needed only 4-byte alignment in practice,

and not 16. However, to conform to the documented specification, he suggested fixing the kernel code to match the spec.

Borislav Petkov objected to this. He said Henrique was looking for problems where there weren't any. He said that Henrique simply had discovered a bug in Intel's documentation, because the alignment issue clearly wasn't a problem in the real world. He suggested alerting the Intel folks to the documentation problem and moving on. As he put it, "If the processor accepts the non-16-byte-aligned update, why do you care?"

But, as **H. Peter Anvin** remarked, the written spec was Intel's guarantee that certain behaviors would work. If the kernel ignored the spec, it could lead to subtle bugs later on. And, **Bill Davidsen** said that if the kernel ignored the alignment requirement, and "if the requirement is enforced in some future revision, and updates then fail in some insane way, the vendor is justified in claiming 'I told you so'."

The end result was that Henrique sent in some patches to make the microcode driver enforce the 16-byte alignment requirement.—**ZACK BROWN**



SharePoint is at the Crossroads – Which Way Will You Go?

SharePoint in the cloud or on premises? Or both? Come to SPTechCon Austin 2015 and learn about the differences between Office 365, cloud-hosted SharePoint, on-premises SharePoint, and hybrid solutions and build your company's SharePoint Roadmap!

For developers, the future means a new app model and new app paradigms. For IT pros and SharePoint admins, it's trying to retain control over an installation that's now in the cloud. For information workers and their managers, it's about learning how to work 'social.' But it's not for everyone.

Where do you need to be?

The answer is simple: SPTechCon Austin. With a collection of the top SharePoint MVPs and expert speakers, more than 80 classes and tutorials to choose from and panels focused on the changes in SharePoint, SPTechCon will teach you how to master the present and plan for the future.

**Migrate to SharePoint 2013! Prepare for Office 365!
Build Your Hybrid Model!**



February 8-11, 2015
Renaissance Austin Hotel

80+ Classes

**40+ Microsoft Expert
Speakers**

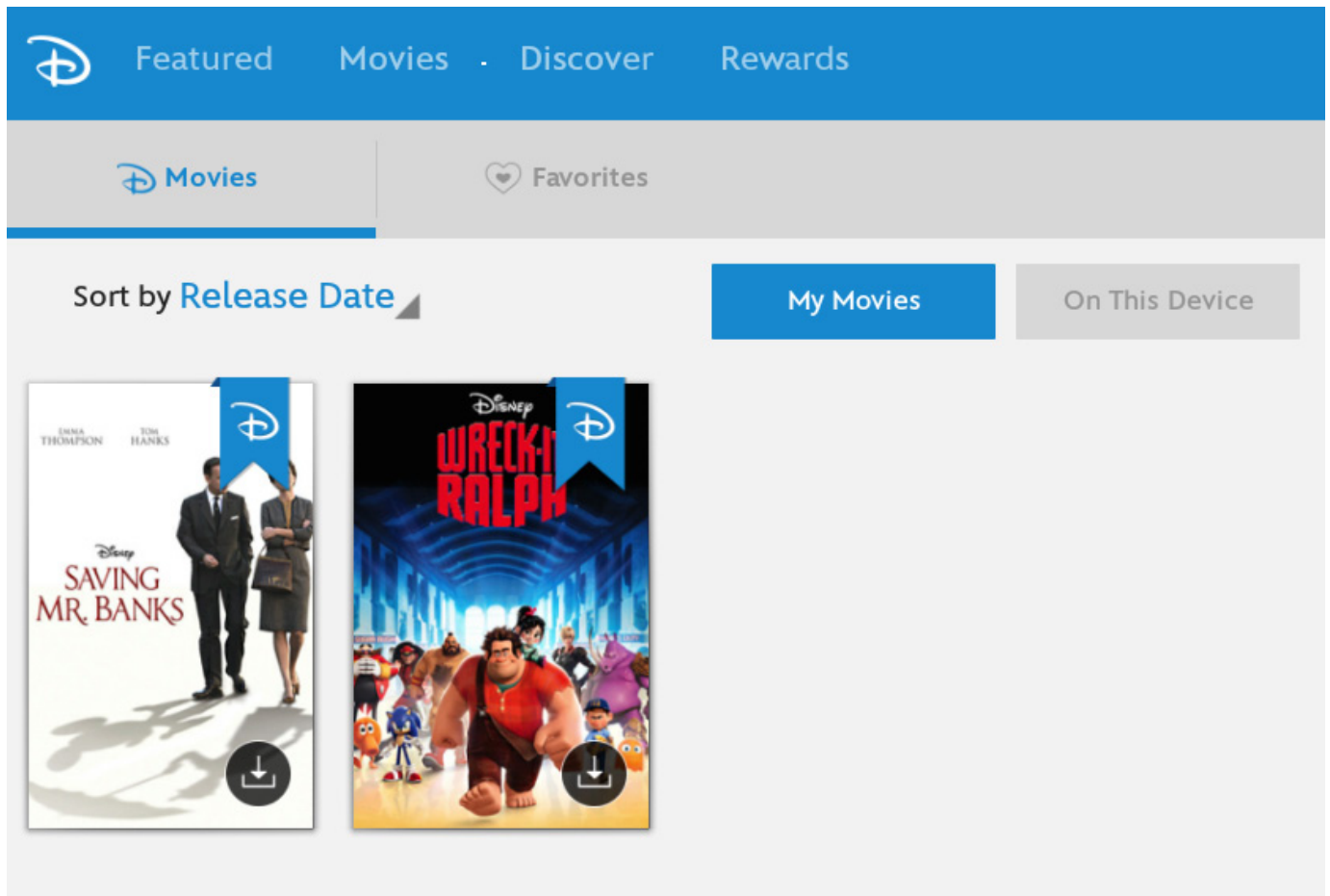
**Get Your Texas-Sized
Registration Discount—
Register NOW!**

www.sptechcon.com

A BZ Media Event

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

Android Candy: Disney Everywhere, Even Android!



As a father of three girls, I have piles and piles of Disney DVDs and Blu-rays. I occasionally look at the “Digital Copy” information and roll my eyes, because it requires some odd Windows DRM software or some other convoluted watching method that usually isn’t possible or even interesting for me.

Recently, however, Disney did

a really cool thing and released an Android app that allows you to stream any movie you have purchased from the Google Play store, iTunes store or from a department store (assuming the store copy came with that seemingly gimmicky digital version).

I tried it today, and sure enough, there was an insert in our *Saving*

Mr. Banks Blu-ray disk that allowed me to redeem a copy of the movie on my Disney account. Now that movie is accessible to me through any Android app along with any iOS app or even a Web browser (using Flash, unfortunately).

If you install the app and connect it to your Google Play account, you'll get a free copy of *Wreck it Ralph*, even if you haven't purchased any movies in the past.

Don't get me wrong, the movies still are completely crippled with

DRM, but at least they are accessible from a multitude of devices. It's the first time the "digital version" of the movies hasn't been a joke—at least in my world. If you have a collection of Disney DVDs with unclaimed codes for digital copies, you can add them to your account and stream the movies instantly. It's actually pretty cool!

To create your Disney account, head over to <http://disneymoviesanywhere.com>.

—**SHAWN POWERS**

Powerful: Rhino



Rhino M4800/M6800

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

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



Tablet: Raven



Raven X240

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

Rugged: Tarantula



Tarantula CF-31

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

EmperorLinux

...where Linux & laptops converge

www.EmperorLinux.com

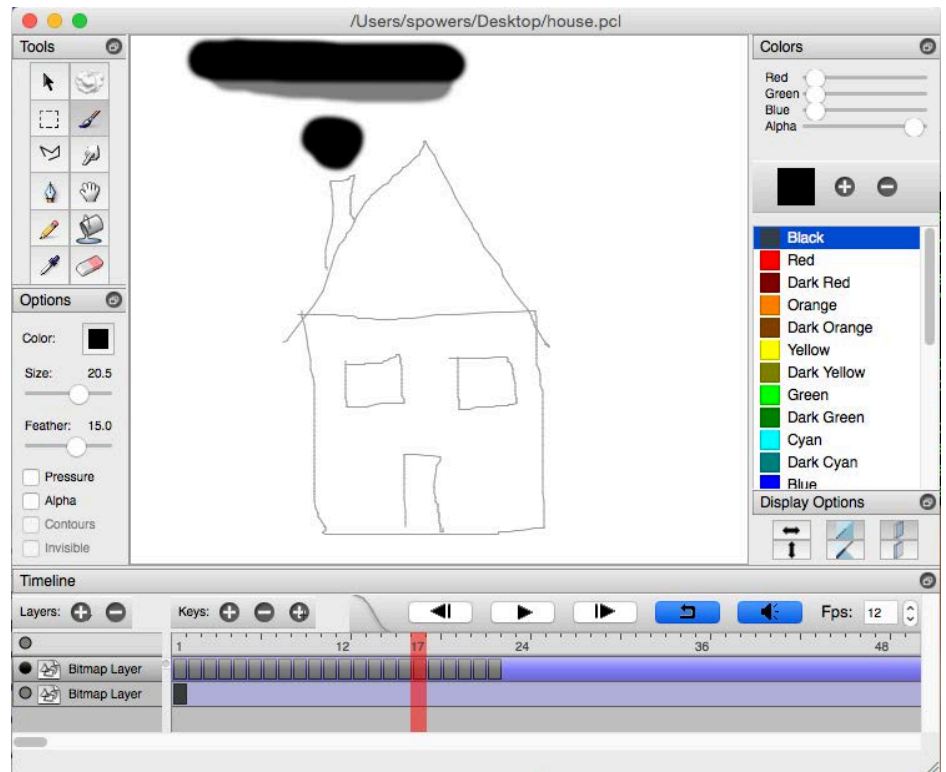
1-888-651-6686



Non-Linux FOSS: Animation Made Easy

If you've ever wanted to make an animated film, the learning curve for such software often is really steep. Thankfully, the Pencil program was released and although basic, it provided a fairly simple way to create animations on your computer (Windows, Mac or Linux) with open-source tools. Unfortunately, the Pencil program was abandoned.

And really, that's the coolest part of open-source software. Building on the incredible Pencil program, a new project was born. Pencil2D is under active development, and it's a cross-platform application allowing for a frame-by-frame animation sequence to be drawn and exported. Pencil2D supports soundtracks, multiple layers, imported graphics and a really cool onion-skin feature so that subsequent frames can be changed by increments making for smooth animations.



The program isn't perfect, and it does crash from time to time. As you can see in my screenshot, it enables a non-artist like myself to create animations. (If I pressed play, you'd be shocked and amazed at my puffing chimney!) Whether you want to make a quick animation or produce a full-length movie, Pencil2D is a neat program that will cost you nothing to try. Download your copy, and see how you can contribute to the project at <http://www.pencil2d.org>.—**SHAWN POWERS**

The Thirteenth Annual
Southern California Linux Expo

SCALE 13x

February 19-22, 2015
Hilton Hotel @ LAX
Los Angeles, CA

<http://www.socallinuxexpo.org>
Use Promo Code LJ13X for a 30%
discount on admission to SCALE



Slow System? iotop Is Your Friend

Back in 2010, Kyle Rankin did an incredible series on Linux Troubleshooting. In Part 1 (<http://www.linuxjournal.com/magazine/hack-and-linux-troubleshooting-part-i-high-load>), he talked about troubleshooting a system struggling with a high load. At that point, I'd already been a system administrator for more than a decade, but it was the first time I'd ever heard of iotop.

```

Total DISK READ :    0.00 B/s | Total DISK WRITE :    272.01 K/s
Actual DISK READ:    0.00 B/s | Actual DISK WRITE:    78.84 K/s
  TID  PRIO  USER      DISK READ  DISK WRITE  SWAPIN     IO>   COMMAND
 140  be/3  root       0.00 B/s   67.02 K/s  0.00 %    0.04 % [jbd2/sda1-8]
17263 be/4  root       0.00 B/s  201.05 K/s  0.00 %    0.04 % ./bitcoind -daemon
 493  be/4  syslog     0.00 B/s    3.94 K/s  0.00 %    0.00 % rsyslogd [rs:main Q:Reg]

```

Figure 1. The Bitcoin daemon is notorious for using a lot of disk I/O.

If you weren't a subscriber in 2010, I highly recommend you read Kyle's entire series. Either way, I use iotop so often, I felt it was prudent to mention it again all these years later. The concept is pretty simple. It's like the top program, but instead of CPU and memory usage, it monitors disk I/O. If you have a system that is extremely slow to respond, but can't seem to figure out what is going on, give iotop a try. You'll probably have to install it, as I've never found a system with iotop installed by default, but it should be in the software repository of just about every Linux distro. And, if you find it useful? Be sure to read Kyle's entire series; it's just as helpful today as it was five years ago!—**SHAWN POWERS**

They Said It

If you can't be funny, be interesting.

—**Harold Ross**

If you want change, you have to make it. If we want progress, we have to drive it.

—**Susan Rice**

Never regret something that once made you smile.

—**Amber Deckers**

Never let the future disturb you. You will meet it, if you have to, with the same weapons of reason which today arm you against the present.

—**Marcus Aurelius Antoninus**

Part of being creative is learning how to protect your freedom. That includes freedom from avarice.

—**Hugh Macleod**

 **ServerBeach**

DEDICATED SERVERS. BY GEEKS FOR GEEKS.

we get how geeks think.

PATIENT MRI EXAM

TurboClocked

CPU Cores: 64

Clock Speed: 6.66 GHz

Bandwidth: 100 Gbps

Refresh Rate: 240 Hz

Storage: 32 PB

Latency: 0.002 ms

Packet Loss: 0.00%

Load Avg: 0.01

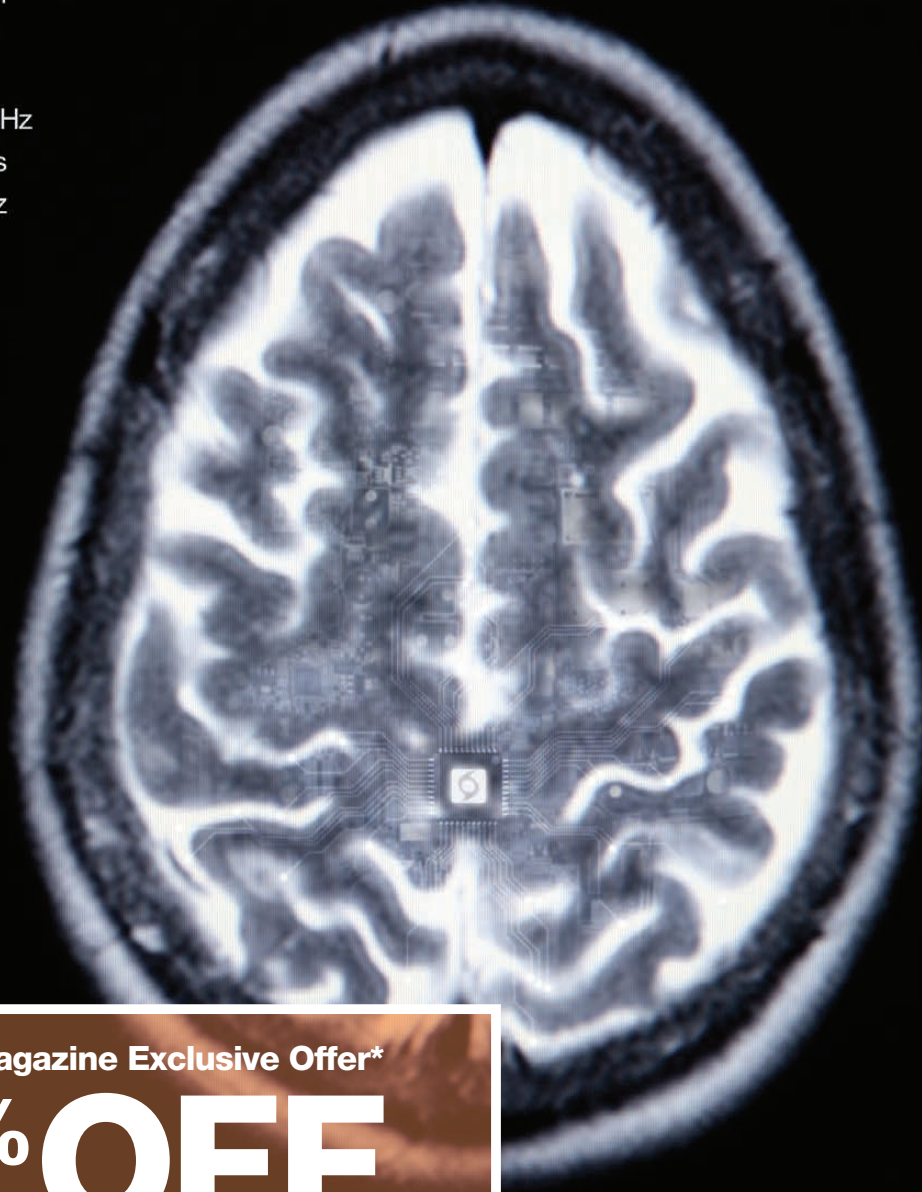
S. BEACH

GEEK, IMA

023Y MALE

1 800 7419939

23:59:59



R

L

Linux Journal Magazine Exclusive Offer*

15% OFF

Call **1.888.840.9091** | serverbeach.com

Sign up for any dedicated server at ServerBeach and get 15% off*. Use the promo code: **LJ15OFF** when ordering.

* Offer expires December 31st, 2010.

Terms and conditions:

© 2010 ServerBeach, a PEER 1 Company. Not responsible for errors or omissions in typography or photography. This is a limited time offer and is subject to change without notice. Call for details.

Designing with Linux

3-D printers are becoming popular tools, dropping in price and becoming available to almost everyone. They can be used to build parts that you can use around the house, but more and more, they also are being used to create instruments for scientific

work. Although a growing library of objects are available in several on-line databases, there is nearly an infinite number of possible things you might want to build. This means you likely will want to design and build your own creations.

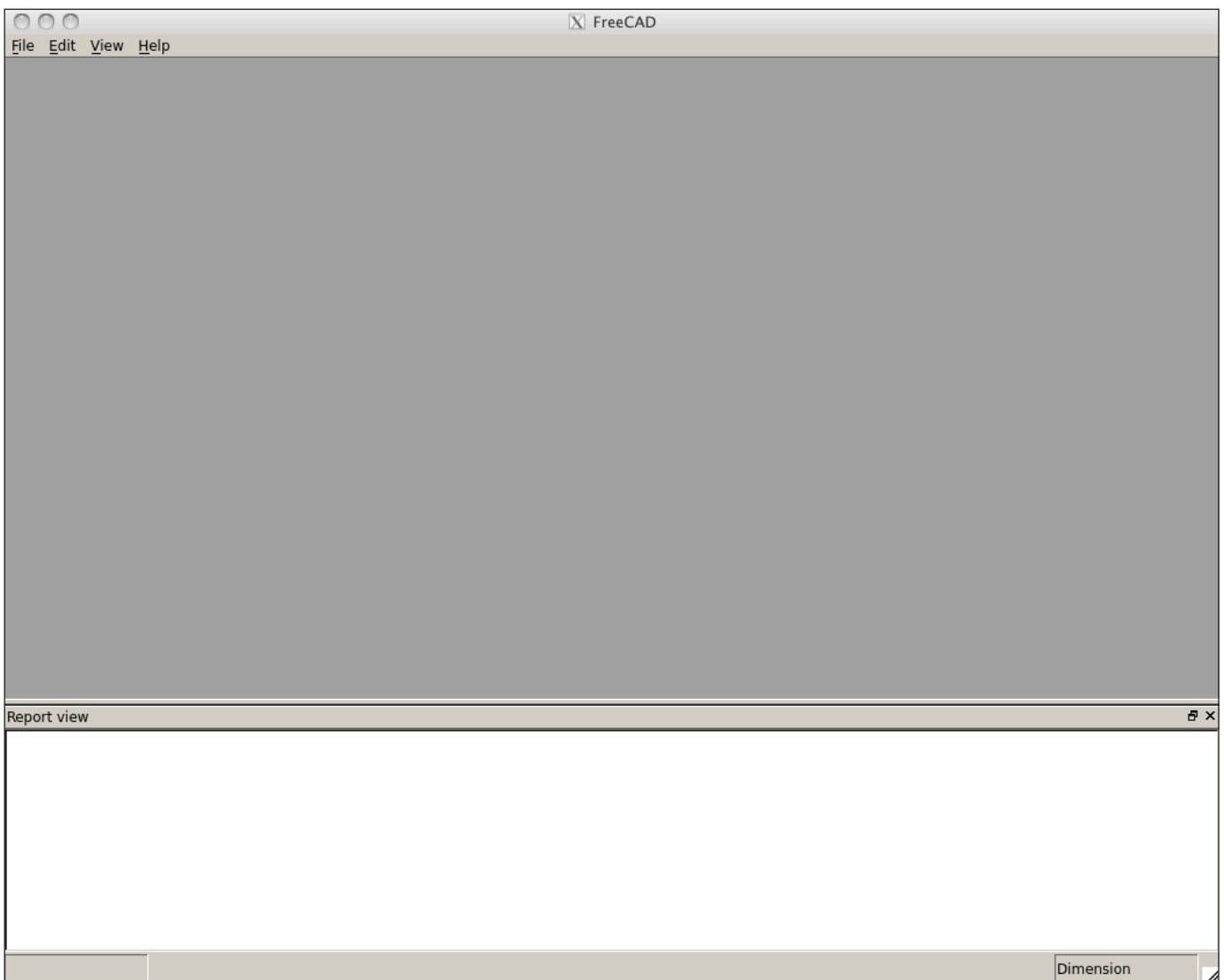


Figure 1. On start up, FreeCAD gives you a blank workspace so you can begin working.

In this article, I take a look at FreeCAD, an open-source parametric 3-D modeler (<http://www.freecadweb.org>). A parametric modeler builds the structures in the design based on a set of properties. Changing the design, thus, is simply a matter of changing the properties of said design.

FreeCAD can read and write several different file formats used in modelling and 3-D printing. It is built as a core application, with plugin modules made to handle specific jobs. Python is the language of choice, because there actually is a built-in Python engine. Additionally, FreeCAD itself can be imported into your own Python script. This gives you access to all of the geometric tools for use in your own code.

FreeCAD already should be in your distribution's package management system. In Debian-based ones, it is simply a matter of using the command:

```
sudo apt-get install freecad
```

In the latest version of Ubuntu, the latest version of FreeCAD actually is available. But, you always have the option of building FreeCAD from source, if you need some non-standard option.

To start it, you can just run `freecad`. It will pop open a window with a blank workspace in it (Figure 1).

FreeCAD uses a workbench concept to give you groups of tools based on the task you are doing at any particular time. You can access the available workbenches by clicking on the View→Workbench menu item. Here, you will get a drop-down list of all of the available options.

As I'm focusing on the idea of building a 3-D object, let's start by activating the parts workbench. Clicking on the View→Workbench→Part menu item will rebuild the interface and introduce all sorts of new tools for you to use (Figure 2).

On the left-hand side, you should see two sub-windows providing a tree view of the objects in your design and a property view of specific objects. Along the bottom are a report view and a Python window. The majority of the display currently is a blank slate. You need to start a new design by clicking on File→New. This will open a new, unnamed document in the main part of the GUI. It will be renamed when you first save the project. You can do this by clicking File→Save. You can add primitive objects to this new document either by clicking on the associated icon in the top toolbar

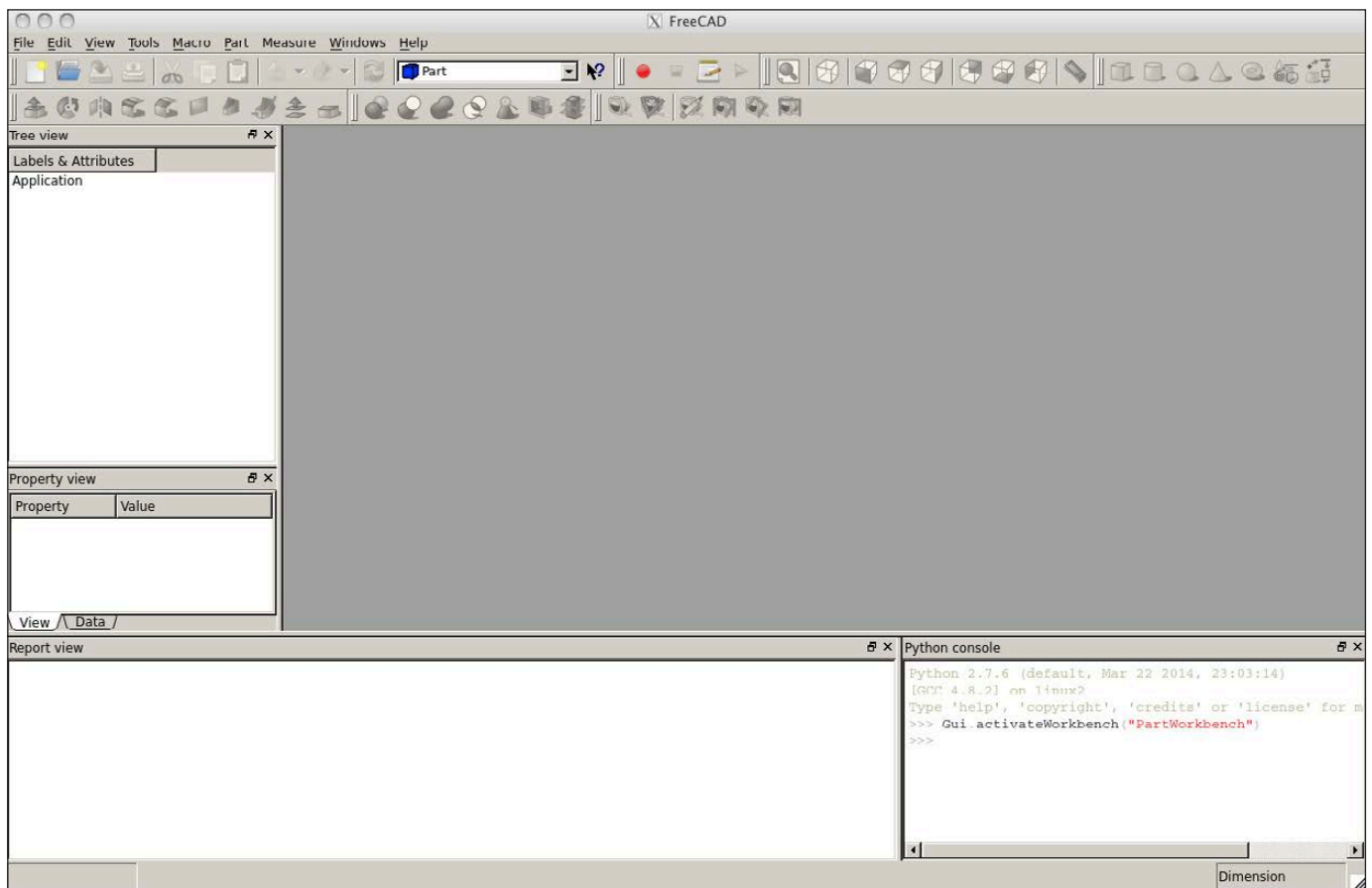


Figure 2. Selecting the parts workbench will change the interface, giving you access to all sorts of new tools to use for building.

or by clicking on the Part→Primitives menu item.

Once you have an object in the document, you can click on it to see its properties on the left-hand side. There are two tabs in the property view pane. The initial tab is the View tab, which provides details on the presentation of your object. The second tab is the Data tab, which contains the details for the construction of your object.

In Figure 3, you can see a torus

that I have started to construct. A torus is described by three angles and two radii. The first two angles describe how much of the torus exists in the cylindrical direction. The third angle describes how much of the torus exists around the circle. The first radius gives the overall radius of the torus, and the second radius represents the cylindrical radius.

Lots of other tools are available as well. You can select a different workbench by clicking on the

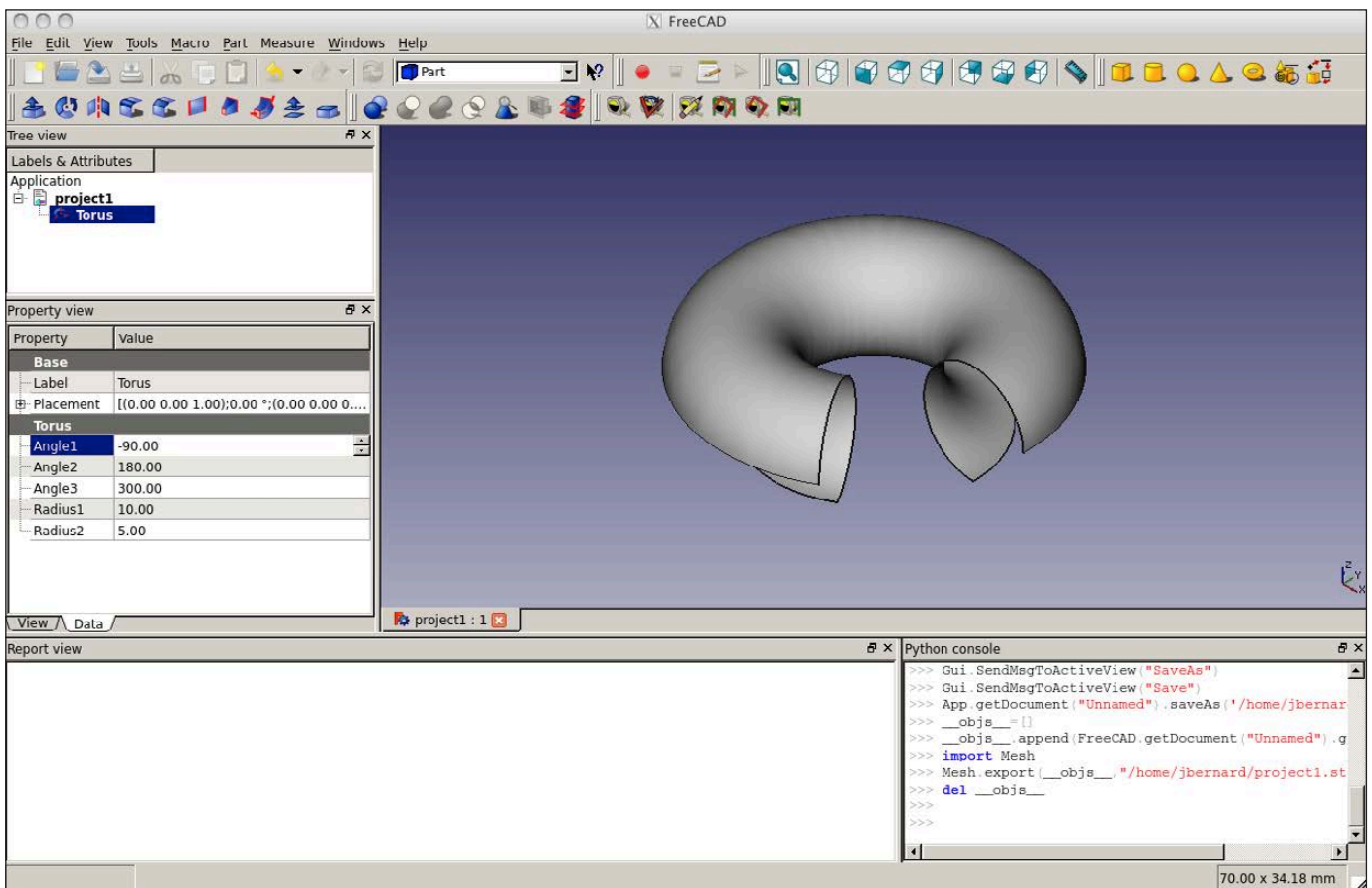


Figure 3. You can build your design based on primitive objects.

View→Workbench menu item, or you can select a new workbench from the drop-down menu in the center of the top toolbar.

To look at another example problem, say you wanted to generate a mesh from your design to feed into some other program, like a computational fluid dynamics program. To start, you will need to select the mesh workbench. This will bring in a new set of tools to work with meshes. You can generate a mesh from your design by clicking

the Meshes→Create mesh from shape menu item. This will pop open a new pane on the left-hand side to control the meshing process. You can choose either standard or mefisto meshing, along with the maximum edge length. Once these are set, you can select the shape you want to mesh and click OK. This will generate the mesh and leave you with a mesh information box describing the number of points, facets and so on (Figure 4).

Tools are available to work on these meshes as well. For example, click

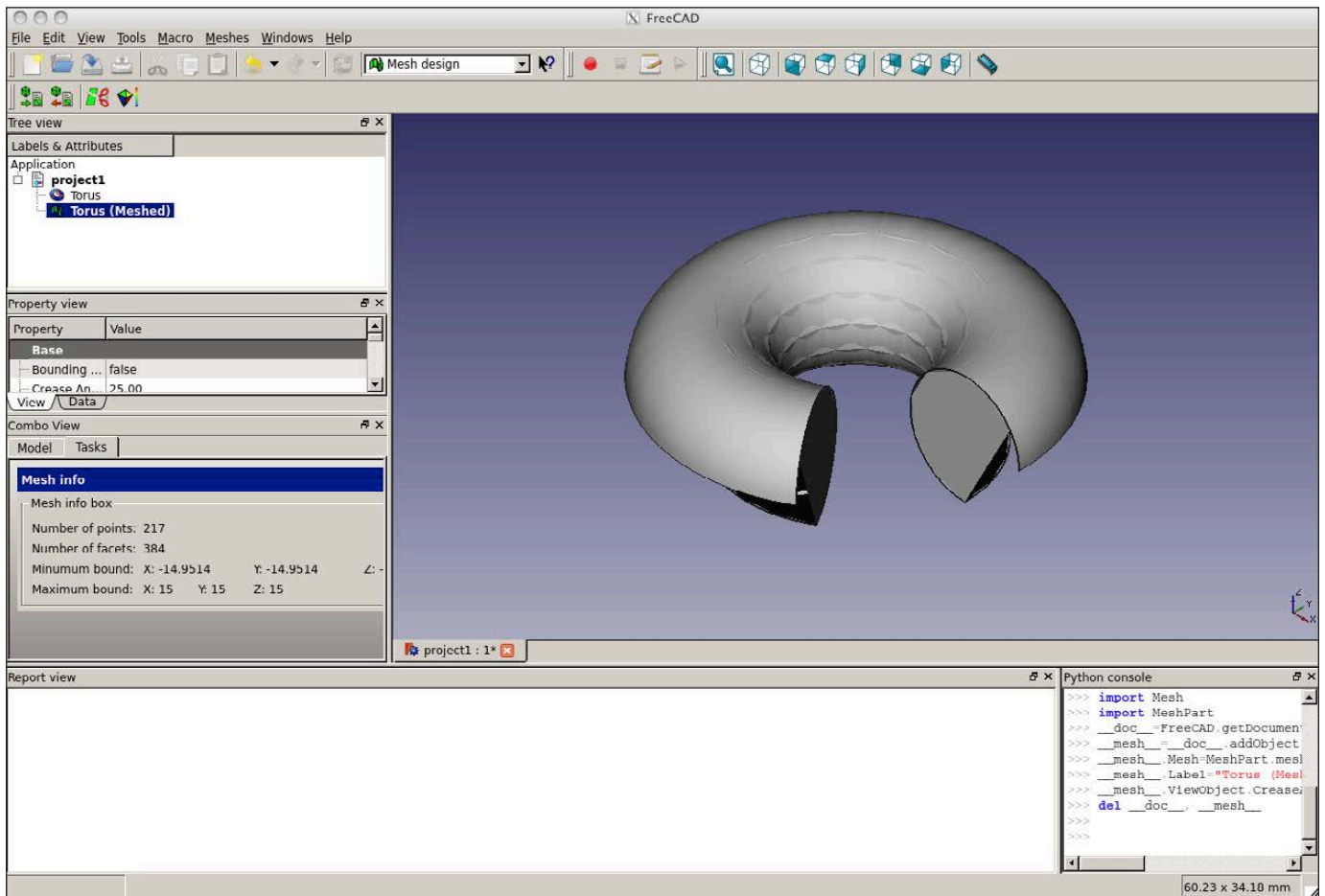


Figure 4. You even can do work on meshes with FreeCAD.

on the Meshes→Analyze→Evaluate & Repair mesh menu item. This will pop up a new pane on the right-hand side where you can analyze your new mesh and find problems like duplicated faces, duplicated points or degenerated faces. If any of those issues are found, a repair button is available to go ahead and fix those issues. This is something that is more likely to happen with very complex objects created from several primitives stitched together.

Once you have your mesh generated and properly tweaked and optimized, you can click on the Meshes→Export mesh menu item, and select the file format you need. For example, if you want to use it in some project in CFX, you could export it as a NASTRAN file. There are several options that should be supported by most other programs that can take mesh files as input. So FreeCAD can act as a very good pre-processing step in many projects.

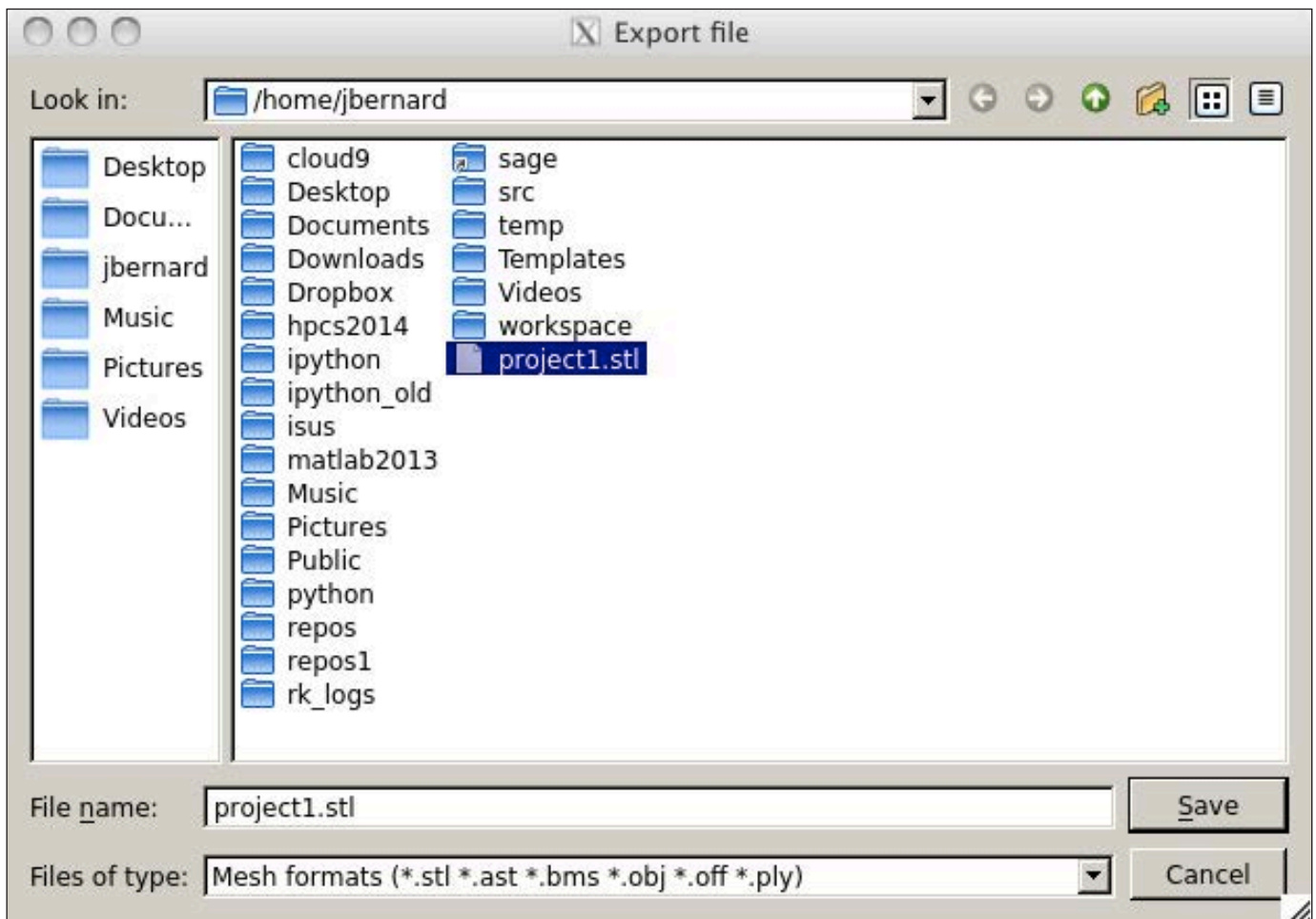


Figure 5. You can export your design to the STL file format.

Once you have finished your object, you will want to have it in a format that can be used in some other context. To do this, you can export your design in one of any number of available formats. For example, if you want to send your design to a 3-D printer, you can export it in an STL file format. To do that, click on File→Export and select the stl option from the drop-down list (Figure 5). You then can take this file and use it in the printing software for your 3-D printer.

Now that you can build your own objects, you can design the exact piece you need for that scientific experiment, or create that specific replacement part you need. If you can, you also should consider sharing your work in one or more of the on-line libraries. That way, you may become famous for your miracle widget or amazing do-hickey. In any case, it is always good to share information so others can build on it to do even greater things.—**JOEY BERNARD**



Wondershaper— QOS in a Pinch

In past articles, I've discussed my BirdCam setup and how it automatically archives video footage from my bird feeders to YouTube every night. That's a really cool process, but unfortunately, it saturates my upstream bandwidth in the evening. I could get crafty with my firewall and limit the priority of traffic from my BirdCam server, but because it's using standard Web protocols to upload to YouTube, the firewall rules would be fairly complex. Thankfully, there's Wondershaper.

Because I'm running my BirdCam server on an independent machine, it doesn't affect any other programs on the server if I just limit the throughput on my Ethernet interface. There are three basic ways I know of to limit bandwidth on a Linux machine: using `tc`, `trickle` and Wondershaper.

The `tc` program is by far the most powerful. It's also by far the most complex. In fact, it's so complex that for my lazy purposes, it just

isn't ideal. If you're looking for protocol-specific QOS on a local interface, `tc` is probably the tool for the job. It is overkill for me.

The `trickle` (and `trickled`) program seems like an ideal way to control the bandwidth on a per-application level. Supposedly, you simply can start a program with `trickle`, and it will limit the bandwidth available. Try as I might, however, I never could get it to do its job. Yes, I'm sure it's a perfectly wonderful tool, but again, I want something dead simple. Enter: Wondershaper.

After installing Wondershaper on your system, it works by controlling the bandwidth on an entire interface. So this:

```
sudo wondershaper eth0 1250 125
```

will limit the `eth0` interface to 10mbps down and 1mbps up. The numbers are listed in KB/s (that wasn't clear to me at first, but Google will convert mbps to KB/s for you). The setting is system-wide and should limit bandwidth for

any network application. To clear the limit:

```
sudo wondershaper clear eth0
```

I simply put the command in my root user's crontab to execute on startup, and my BirdCam server doesn't monopolize my bandwidth when it uploads nightly to YouTube. It seems like a simple tool, and it is. But, it works so well and provides such a useful service, I'm making Wondershaper

this month's Editors' Choice winner. It's not a new program, but it's something I'd never used before, and I can't imagine living without it!

(NOTE: before I get hundreds of e-mail messages, yes, I know Wondershaper is a program from 2002. I mentioned other newer, more robust alternatives above, but the truth is, Wondershaper still is a powerful, simple tool that does one thing and does it really well.)—**SHAWN POWERS**

The White Paper Library on LinuxJournal.com



www.linuxjournal.com/whitepapers



REUVEN M.
LERNER

Users, Permissions and Multitenant Sites

Your multitenant site can become even more flexible once you add a database.

In my last article, I started to look at multitenant Web applications. These are applications that run a single time, but that can be retrieved via a variety of hostnames. As I explained in that article, even a simple application can be made multitenant by having it check the hostname used to connect to the HTTP server, and then by displaying a different set of content based on that.

For a simple set of sites, that technique can work well. But if you are working on a multitenant system, you more likely will need a more sophisticated set of techniques.

For example, I recently have been working on a set of sites that help people practice their language skills. Each site uses the same software

but displays a different interface, as well as (obviously) a different set of words. Similarly, one of my clients has long operated a set of several dozen geographically targeted sites. Each site uses the same software and database, but appears to the outside world to be completely separate. Yet another reason to use a multitenant architecture is if you allow users to create their own sites—and, perhaps, add users to those private sites.

In this article, I describe how to set up all of the above types of sites. I hope you will see that creating such a multitenant system doesn't have to be too complex, and that, on the contrary, it can be a relatively easy way to provide a single software service to a variety of audiences.

It's true that you can do sophisticated things with Sinatra, but when it comes to working with databases and large-scale projects, I prefer to use Ruby on Rails.

Identifying the Site

In my last article, I explained how to modify `/etc/passwd` such that more than one hostname would be associated with the same IP address. Every multitenant site uses this same idea. A limited set of IP addresses (and sometimes only a single IP address) can be mapped to a larger number of hostnames and/or domain names. When a request comes in, the application first checks to see which site has been requested, and then decides what to do based on it.

The examples in last month's article used Sinatra, a lightweight framework for Web development. It's true that you can do sophisticated things with Sinatra, but when it comes to working with databases and large-scale projects, I prefer to use Ruby on Rails. So here I'm using Rails, along with a back end in PostgreSQL.

In order to do that, you first need to create a simple Rails application:

```
rails new -d postgresql multiatf
```

Then create a "multiatf" user in your PostgreSQL installation:

```
createuser multiatf
```

Finally, go into the multiatf directory, and create the database:

```
rake db:create
```

With this in place, you now have a working (if trivially simple) Rails application. Make sure you still have the following two lines in your `/etc/hosts` file:

```
127.0.0.1 atf1  
127.0.0.1 atf2
```

And when you start up the Rails application:

```
rails s
```

you can go to `http://atf1:3000` or `http://atf2:3000`, and you should see the same results—namely, the basic "hello" that you get from a

Rails application before you have done anything.

The next step then is to create a default controller, which will provide actual content for your users. You can do this by saying:

```
rails g controller welcome
```

Now that you have a “welcome” controller, you should uncomment the appropriate route in `config/routes.rb`:

```
root 'welcome#index'
```

If you start your server again and go to `http://atf1:3000`, you’ll now get an error message, because Rails knows to go to the “welcome” controller and invoke the “index” action, but no such action exists. So, you’ll have to go into your controller and add an action:

```
def index
  render text: "Hello!"
end
```

With that in place, going to your home page gives you the text.

So far, that’s not very exciting, and it doesn’t add to what I explored in my last article. You can, of course, take advantage of the fact that your “index” method is rendering text, and

that you can interpolate values into your text dynamically:

```
def index
  render text: "Hello, visitor to #{request.host}!"
end
```

But again, this is not what you’re likely to want. You will want to use the hostname in multiple places in your application, which means that you’ll repeatedly end up calling “request.host” in your application. A better solution is to assign a `@hostname` variable in a `before_action` declaration, which will ensure that it takes place for everyone in the system. You could create this “before” filter in your welcome controller, but given that this is something you’ll want for all controllers and all actions, I think it would be wiser to put it in the application controller.

Thus, you should open `app/controllers/application_controller.rb`, and add the following:

```
before_action :get_hostname

def get_hostname
  @hostname = request.host
end
```

Then, in your welcome controller, you can change the “index” action

Why use a scaffold? I know that it's very popular among Rails developers to hate scaffolds, but I actually love them when I start a simple project.

to be:

```
def index
  render text: "Hello, visitor to #{@hostname}!"
end
```

Sure enough, your hostname now will be available as `@hostname` and can be used anywhere on your site.

Moving to the Database

In most cases, you'll want to move beyond this simple scheme. In order to do that, you should create a "hosts" table in the database. The idea is that the "hosts" table will contain a list of hostnames and IDs. It also might contain additional configuration information (I discuss that below). But for now, you can just add a new resource to the system. I even would suggest using the built-in scaffolding mechanism that Rails provides:

```
rails g scaffold hosts name:string
```

Why use a scaffold? I know that it's very popular among Rails

developers to hate scaffolds, but I actually love them when I start a simple project. True, I'll eventually need to remove and rewrite parts, but I like being able to move ahead quickly and being able to poke and prod at my application from the very first moments.

Creating a scaffold in Rails means creating a resource (that is, a model, a controller that handles the seven basic RESTful actions and views for each of them), as well as the basic tests needed to ensure that the actions work correctly. Now, it's true that on a production system, you probably won't want to allow anyone and everyone with an Internet connection to create and modify existing hosts. And indeed, you'll fix this in a little bit. But for now, this is a good and easy way to set things up.

You will need to run the new migration that was created:

```
rake db:migrate
```

And then you will want to add your

two sites into the database. One way to do this is to modify `db/seeds.rb`, which contains the initial data that you'll want in the database. You can use plain-old Active Record method calls in there, such as:

```
Host.create([{:name: 'atf1'}, {:name: 'atf2'}])
```

Before you add the seeded data, make sure the model will enforce some constraints. For example, in `app/models/host.rb`, I add the following:

```
validates :name, {:uniqueness => true}
```

This ensures that each hostname will appear only once in the “hosts” table. Moreover, it ensures that when you run `rake db:seed`, only new hosts will be added; errors (including attempts to enter the same data twice) will be ignored.

With the above in place, you can add the seeded data:

```
rake db:seed
```

Now, you should have two records in your “hosts” table:

```
[local]/multiatf_development=# select name from hosts;
-----
| name |
-----
```

```
| atf1 |
-----
| atf2 |
-----
(2 rows)
```

With this in place, you now can change your application controller:

```
before_action :get_host

def get_host
  @requested_host = Host.where(name: request.host).first

  if @requested_host.nil?
    render text: "No such host '#{request.host}'.", status: 500
    return false
  end
end
```

(By the way, I use `@requested_host` here, so as not to collide with the `@host` variable that will be set in `hosts_controller`.)

`@requested_host` is no longer a string, but rather an object. It, like `@requested_host` before, is an instance variable set in a before filter, so it is available in all of your controllers and views. Notice that it is now potentially possible for someone to access your site via a hostname that is not in your “hosts” table. If and when that

happens, `@requested_host` will be nil, and you give an appropriate error message.

This also means that you now have to change your “welcome” controller, ever so slightly:

```
def index
  render text: "Hello, visitor to #{@requested_host.name}!"
end
```

This change, from the string `@requested_host` to the object `@requested_host`, is about much more than just textual strings. For one, you now can restrict access to your site, such that only those hosts that are active can now be seen. For example, let’s add a new boolean column, `is_active`, to the “hosts” table:

```
rails g migration add_is_active_to_hosts
```

On my machine, I then edit the new migration:

```
class AddIsActiveToHosts < ActiveRecord::Migration
  def change
    add_column :hosts, :is_active, :boolean, default: true,
      ↳null: false
  end
end
```

According to this definition, sites

are active by default, and every site must have a value for `is_active`. You now can change your application controller’s `get_host` method:

```
def get_host

  @requested_host = Host.where(name: request.host).first

  if @requested_host.nil?
    render text: "No such host '#{request.host}'.", status: 500
    return false
  end

  if !@requested_host.is_active?
    render text: "Sorry, but '#{@requested_host.name}'
      ↳is not active.", status: 500
    return false
  end
end
```

Notice how even a simple database now allows you to check two conditions that were not previously possible. You want to restrict the hostnames that can be used on your system, and you want to be able to turn hosts on and off via the database. If I change `is_active` to false for the “atf1” site:

```
UPDATE Hosts SET is_active = 'f' WHERE name = 'atf1';
```

immediately, I’m unable to access the “atf1” site, but the “atf2” site works

just fine.

This also means that you now can add any number of sites—without regard to host or domain—so long as they all have DNS entries that point to your IP addresses. Adding a new site is as simple as registering the domain (if it hasn't been registered already), configuring its DNS entries such that the hostname points to your IP address, and then adding a new entry in your Hosts table.

Users and Permissions

Things become truly interesting when you use this technique to allow users to create and manage their own sites. Suddenly, it is not just a matter of displaying different text to different users, but allowing different users to log in to different sites. The above shows how you can have a set of top-level administrators and users who can log in to each site. However, there often are times when you will want to restrict users to be on a particular site.

There are a variety of ways to handle this. No matter what, you need to create a “users” table and a model that will handle your users and their ability to register and log in. I used to make the foolish mistake of implementing such login

systems on my own; nowadays, I just use “Devise”, the amazing Ruby gem that handles nearly anything you can imagine having to do with registration and authentication.

I add the following line to my project's Gemfile:

```
gem 'devise'
```

Next, I run `bundle install`, and then:

```
rails g devise:install
```

on the command line. Now that I have Devise installed, I'll create a user model:

```
rails g devise user
```

This creates a new “user” model, with all of the Devise goodies in it. But before running the migrations that Devise has provided, let's make a quick change to the Devise migration.

In the migration, you're going to add an `is_admin` column, which indicates whether the user in question is an administrator. This line should go just before the `t.timestamps` line at the bottom, and it indicates that users are not administrators by default:

```
t.boolean :is_admin, default: false, null: false
```


With this in place, you now can run the migrations. This means that users can log in to your system, but they don't have to. It also means that you can designate users as administrators. Devise provides a method that you can use to restrict access to particular areas of a site to logged-in users. This is not generally something you want to put in the application controller, since that would restrict people from logging in. However, you can say that your "welcome" and "host" controllers are open only to registered and logged-in users by putting the following at the top of these controllers:

```
before_action :authenticate_user!
```

With the above, you already have made it such that only registered and logged-in users are able to see your "welcome" controller. You could argue that this is a foolish decision, but it's one that I'm comfortable with for now, and its wisdom depends on the type of application you're running. (SaaS applications, such as Basecamp and Harvest, do this, for example.) Thanks to Devise, I can register and log in, and then...well, I can do anything I want, including adding and removing hosts.

It's probably a good idea to

restrict your users, such that only administrators can see or modify the hosts controller. You can do that with another `before_action` at the top of that controller:

```
before_action :authenticate_user!
before_action :only_allow_admins
before_action :set_host, only: [:show, :edit, :update, :destroy]
```

Then you can define `only_allow_admins`:

```
def only_allow_admins
  if !current_user.is_admin?
    render text: "Sorry, but you aren't allowed there",
           status: 403
    return false
  end
end
```

Notice that the above `before_action` filter assumes that `current_user` already has been set, and that it contains a user object. You can be sure that this is true, because your call to `only_allow_admins` will take place only if `authenticate_user!` has fired and has allowed the execution to continue.

That's actually not much of a problem. You can create a "memberships" table that joins "users" and "hosts" in a many-to-many relationship. Each user thus

can be a member of any number of hosts. You then can create a `before_action` routine that checks to be sure not only whether users are logged in, but also whether they are a member of the host they're currently trying to access. If you want to provide administrative rights to users within their site only, you can put such a column (for example, `"is_host_admin"`) on the memberships table. This allows users to be a member of as many sites as they might want, but to administer only those that have been specifically approved.

Additional Considerations

Multitenant sites raise a number of additional questions and possibilities. Perhaps you want to have a different style for each site. That's fine. You can add a new "styles" table, which has two columns: `"host_id"` (a number, pointing to a row in the host table) and `"style"`, text containing CSS, which you can read into your program at runtime. In this way, you can let users style and restyle things to their heart's content.

In the architecture described here, the assumption is that all data is in

Now Available: *Practice Makes Python* by Reuven M. Lerner

My new e-book, *Practice Makes Python*, is now available for purchase. The book is aimed at people who have taken a Python course or learned it on their own, but want to feel more comfortable with the "Pythonic" way of doing things—using built-in data structures, writing functions, using functional techniques, such as comprehensions, and working with objects.

Practice Makes Python contains 50 exercises that I have used in nearly a decade of on-site training classes in the US, Europe, Israel and China. Each exercise comes with a solution, as well as a detailed description of why the solution works, often along with alternatives. All are aimed at improving your proficiency with Python, so that you can use it effectively in your work.

You can read more about the book at <http://lerner.co.il/practice-makes-python>.

Linux Journal readers can get 10% off the purchase price by using the coupon code LINUXJOURNAL at checkout. Questions or comments can be sent to me by e-mail at reuven@lerner.co.il or @reuvenmlerner on Twitter.



DAVE TAYLOR

The find | xargs Sequence

find | xargs: the magic of smart pipes versus filenames with spaces.

In my last article, I dug into the weird but powerful `find` command, a tool that I find to be an essential part of working with the command line on a Linux system, and as a key tool for shell scripts too. Although it's super powerful, `find` has some odd quirks and does a really poor job with filenames that have spaces.

Indeed, in the good-old days, UNIX was developed with a standard rule of "no spaces in filenames", so it's only recently with the addition of far longer filename options that spaces have shown up to plague us Linux users. The problem, of course, is that the standard field separator in the shell is, you guessed it, the space. So if you have a file called "My Latest Story", just about every command is going to hiccup.

Try this, and it'll fail:

```
cat My Latest Story
```

saying that file "My", file "Latest" and file "Story" are not found.

Savvy command-line users have long since learned that filename completion is the easiest solution to this, typing in the fragment `cat My` then pressing <Tab> to have it completed:

```
cat My\ Latest\ Story
```

Aesthetically yechy, but it's functional. You also can quote filenames, of course, so this also would work:

```
cat "My Latest Story"
```

But, again, it's a hassle. The

real solution simply is never to use spaces in Linux filenames, but as a shell script writer, you can't guarantee that your script users meet the same criteria, so you've got to cope. And, that's where `find` tends to fall down.

Mutual Incompatibility: `find` and Spaces

There's a rather kludgy solution that's now part of the complicated `find` language, fortunately, and it's just a simple variant on the basic `-print` predicate: `-print0`.

Run it by itself, however, and you'll get really odd output, because for every matching filename, `find` ends the filename with an ASCII 0 rather than the usual end of line. Try it, you'll see the output is a bit confusing!

To get this all to work with `find`, the most common solution is to pipe the output of `find` into the `xargs` command and specify the `-0` flag:

```
find . -name "*.c" -print0 | xargs -0 ls -l
```

The above snippet would work for source files with names like "black box 2.c" and "chapter 3 problem 8.c".

Let's start with just a simple `find`:

```
$ find . -name "*.c"
./black box 2.c
./chapter 3 problem 8.c
./helloworld.c
./sample.c
```

Add the `-print0`, and the output is a bit wonky, as expected:

```
$ find . -name "*.c" -print0
./black box 2.c./chapter 3 problem 8.c./helloworld.c./sample.c$
```

Messy. Worse, what if you use the `find` command and forget to compensate for those pesky space-filled filenames? Oh, it's not pretty:

```
$ find . -name "*.c" | xargs ls -l
ls: ./black: No such file or directory
ls: ./chapter: No such file or directory
ls: 2.c: No such file or directory
ls: 3: No such file or directory
ls: 8.c: No such file or directory
ls: box: No such file or directory
ls: problem: No such file or directory
-rw-r--r-- 1 taylor staff 0 Nov 5 14:39 ./helloworld.c
-rw-r--r-- 1 taylor staff 0 Nov 5 14:39 ./sample.c
```

I warned you up front that spaces in filenames cause trouble, and here's that trouble come to roost.

Add the `-print0` instead of the assumed default of `-print`, pipe that directly to `xargs`, and now it all

If you've ever seen a “stack overflow” or “buffer overflow” on the command line, you'll appreciate the -n flag.

makes sense:

```
$ find . -name "*.c" -print0 | xargs -0 ls -l
-rw-r--r-- 1 taylor staff 0 Nov 5 14:39 ./black box 2.c
-rw-r--r-- 1 taylor staff 0 Nov 5 14:39 ./chapter 3 problem 8.c
-rw-r--r-- 1 taylor staff 0 Nov 5 14:39 ./helloworld.c
-rw-r--r-- 1 taylor staff 0 Nov 5 14:39 ./sample.c
```

I've written about dealing with spaces in filenames within shell scripts in the past. It's a pain. Now at least with `find`, you now know how to work in a space-friendly way.

A Bit More about xargs

Before moving on to the dynamic duo of `find` and `xargs`, however, let's spend a time bit more time on `xargs` itself. The `xargs` command is designed to let you invoke another command with arguments received in a pipe.

Commonly, you'll see `find|xargs`, but it turns out you can do other things with it too, as you'll see.

More important, remember that the first argument given to `xargs` itself is the command you want to run. A common usage might be something

like this:

```
xargs grep -i "pattern"
```

as part of a pipeline.

Where `xargs` really shines though is with its many command-line arguments. One of the most useful of those is `-n`, which lets you specify the maximum number of entries it should accumulate before running the specified command. If you've ever seen a “stack overflow” or “buffer overflow” on the command line, you'll appreciate the `-n` flag. Here's a simple example:

```
$ echo this is a demo of the xargs -n flag | xargs -n3
this is a
demo of the
xargs -n flag
```

As you can see, the `-n` flag causes `xargs` to push out its buffer every `n` items—darn useful with really big directories!

Even more useful is the `-p` option that has `xargs` prompt you to proceed with the given command. Want to



KYLE RANKIN

Secure Server Deployments in Hostile Territory

When your server is running in a risky environment, what steps should you take to secure it?

Would you change what you said on the phone, if you knew someone malicious was listening? Whether or not you view the NSA as malicious, I imagine that after reading the NSA coverage on *Linux Journal*, some of you found yourselves modifying your behavior. The same thing happened to me when I started deploying servers into a public cloud (EC2 in my case).

Although I always have tried to build secure environments, EC2 presents a number of additional challenges both to your fault-tolerance systems and your overall security. Deploying a server on EC2 is like dropping it out of a helicopter behind enemy lines without so much as an IP address.

In this article, I discuss some of the techniques I use to secure servers when they are in hostile

territory. Although some of these techniques are specific to EC2, most are adaptable to just about any environment.

Behind Enemy Lines

So, what makes EC2 so hostile anyway? When you secure servers in a traditional environment, you may find yourself operating under a few assumptions. First, you likely assume that the external network is the main threat and that your internal network is pretty safe. You also typically assume that you control the server and network hardware, and if you use virtualization, the hypervisor as well. If you use virtualization, you probably also assume that other companies aren't sharing your hardware, and you probably never would think it is possible that

a malicious user might share your virtualization platform with you.

In EC2, all of those assumptions are false. The internal and external network should be treated as potentially hostile. The server and network hardware are under someone else's control. Someone else also controls the hypervisor that manages all of the virtual machines, and your virtual machines definitely share hardware with other companies. Finally, although it may not be something that happens every day, it's definitely possible that your virtualization neighbor might be malicious.

EC2-Specific Practices

Although many of the practices I describe here could be used in just about any environment, a few of them are specific to EC2, but even then, you may find ways to map these notions to other cloud environments. Most of these practices revolve around Security Groups. EC2 Security Groups can be thought of in some ways like a VLAN in a traditional network. With Security Groups, you can create firewall settings to block incoming traffic to specific ports for all servers that are members of a specific group. Unlike traditional

VLANs, you can create firewall rules within Security Groups that block traffic between members of that group. Servers can be members of multiple Security Groups, although it's important to know that Security Groups are assigned only when an instance is created—you can't add or remove Security Groups from an instance after you create it.

I generally use Security Groups like most people might use VLANs only with some changes. Every group of servers that share a common purpose have their own Security Group. All groups block all outside traffic by default, and I open ports only as I need them. For the most part, Security Groups allow no external access. I also have kept the "default" Security Group EC2 puts in place and make every server a member of that group as well; however, I lock down that group and use it only when I want to grant access from a different Security Group to all of my servers. For instance, I might use changes to the default Security Group to allow all servers to talk to my Puppetmaster server on its custom port. As another example, I use a VPN to access my cloud network, and that VPN is granted access to SSH into all of the servers in my environment.

Finally, I never store a secret in my userdata file. Often when you spawn a server in EC2, you provide the server with a userdata file. A number of AMIs (Amazon Machine Images—the OS install image you choose) are configured to execute the userdata script. Although in some cases this file is used to pass specific configuration values on to the server, many people (myself included) use the file as a post-install script. In my case, I use it to configure my configuration management system (Puppet) and from that point on let it take over the configuration of the system. What you may not know is that the contents of the userdata script are available via an API call to any user who is on the system throughout the life of the instance. If you use the userdata file to inject any sort of secrets (certificates or SSH private keys, passwords or shared secrets the system uses in its configuration, or anything you wouldn't want a regular user to see), those secrets will be visible to any user on the system. In fact, if you happen to use Puppet yourself (or otherwise have `facter` installed on the system), `facter` itself will return the contents of that userdata script for you.

Handling Secrets

It's incredibly important to think about how you manage secrets in a cloud environment beyond just the userdata script. The fact is, despite your best efforts, you still often will need to store a private key or password in plain text somewhere on the system. As I mentioned, I use Puppet for configuration management of my systems. I store all of my Puppet configuration within Git to keep track of changes and provide an audit trail if I ever need it. Having all of your configuration in Git is a great practice, but the first security practice I recommend with respect to secrets is to avoid storing any plain-text secrets in your configuration management system. Whenever possible, I try to generate secrets on the hosts that need them, so that means instead of pushing up a GPG or SSH key pair to a server, I use my configuration management system to generate one on the host itself.

Another practice I put in place is to store secrets in ramdisk whenever possible. This avoids the problem of securely deleting files on a hard drive that may go away at any moment. We just wrote a basic module in our configuration



SHAWN POWERS

Vagrant Simplified

Vagrant can be overwhelming, but don't let that stop you from taking advantage of this awesome tool.

I admit it, some tools confuse me. I know they must be amazing, because programs don't get popular by being dumb (well, reality TV, but that's another story). I have the same sort of confusion with Vagrant that I have with Wine, Docker, Chef and countless other amazing tools people constantly rave about. So in this article, I'm going to break down Vagrant into its simplest form.

Don't get me wrong, I could follow along with the tutorials and get a virtual machine running by typing the magic `vagrant up` command. The thing is, I really don't like magic when it comes to computers. I like to know what is happening, why it's happening and where to look when things go wrong. Ultimately that's my goal, to be able to fix it when it breaks. Without an understanding of how things truly work, it gets really scary when the magic button quits working.

What It Is

Simply put, Vagrant is a front end to an underlying virtualization program. By default, the back-end program is VirtualBox, although Vagrant can work with other underlying virtualization systems. This realization was important for me, because the line between what had to be inside VirtualBox and what Vagrant actually did on its own was murky. If you use Vagrant, you don't ever need to start VirtualBox—truly. It won't hurt anything if you do start it, but Vagrant uses VirtualBox more like a tool than a system.

Another reason this is important is because it means there is no intermingled dependencies between Vagrant and VirtualBox. By that I mean you can take your Vagrantfiles to another computer, and it will work just fine. It simply will use the copy of VirtualBox you have installed on the new computer and work exactly the same.

Much like brushing your teeth with a hairbrush doesn't make much sense, using Vagrant for setting up your permanent data center might not be the best idea.

When Does It Make Sense to Use Vagrant?

Much like brushing your teeth with a hairbrush doesn't make much sense, using Vagrant for setting up your permanent data center might not be the best idea. Sure, you could use it, but Vagrant really excels at building VMs very fast and destroying them when you're finished. In fact, most people use Vagrant for one of two things: creating a development environment to test their code and creating temporary servers on demand when the workload requires it.

One of the nice side effects of using Vagrant is that it forces you to think of your persistent data as separate from your server. I've found that even in situations where I'm not using Vagrant, I'm now smarter about making sure my data isn't dependent on a single point of failure. An example is my `/usr/local/bin` folder. Most of my machines have tons of little scripts I've written that live in the `/usr/local/bin` folder. Since I've

been using Vagrant, I think about my scripts as something that should be accessible by my machines, but maybe not stored in the local file space on a server. Sure, I have backups, but if I can keep my data separate from my server filesystem, moving to a new server is much easier.

What It Actually Does

I already mentioned that Vagrant is a front end to VirtualBox. Of course, VirtualBox already has a command-line interface, but Vagrant is far more powerful. Rather than install an operating system, Vagrant takes a "template" of a fully installed machine and creates a clone. That means you can have a fully running system in seconds instead of going through the installation process. The "templates" are referred to as "boxes" in the Vagrant world, and there's no need to make your own. You can download generic boxes of most Linux distributions, which means zero setup time.

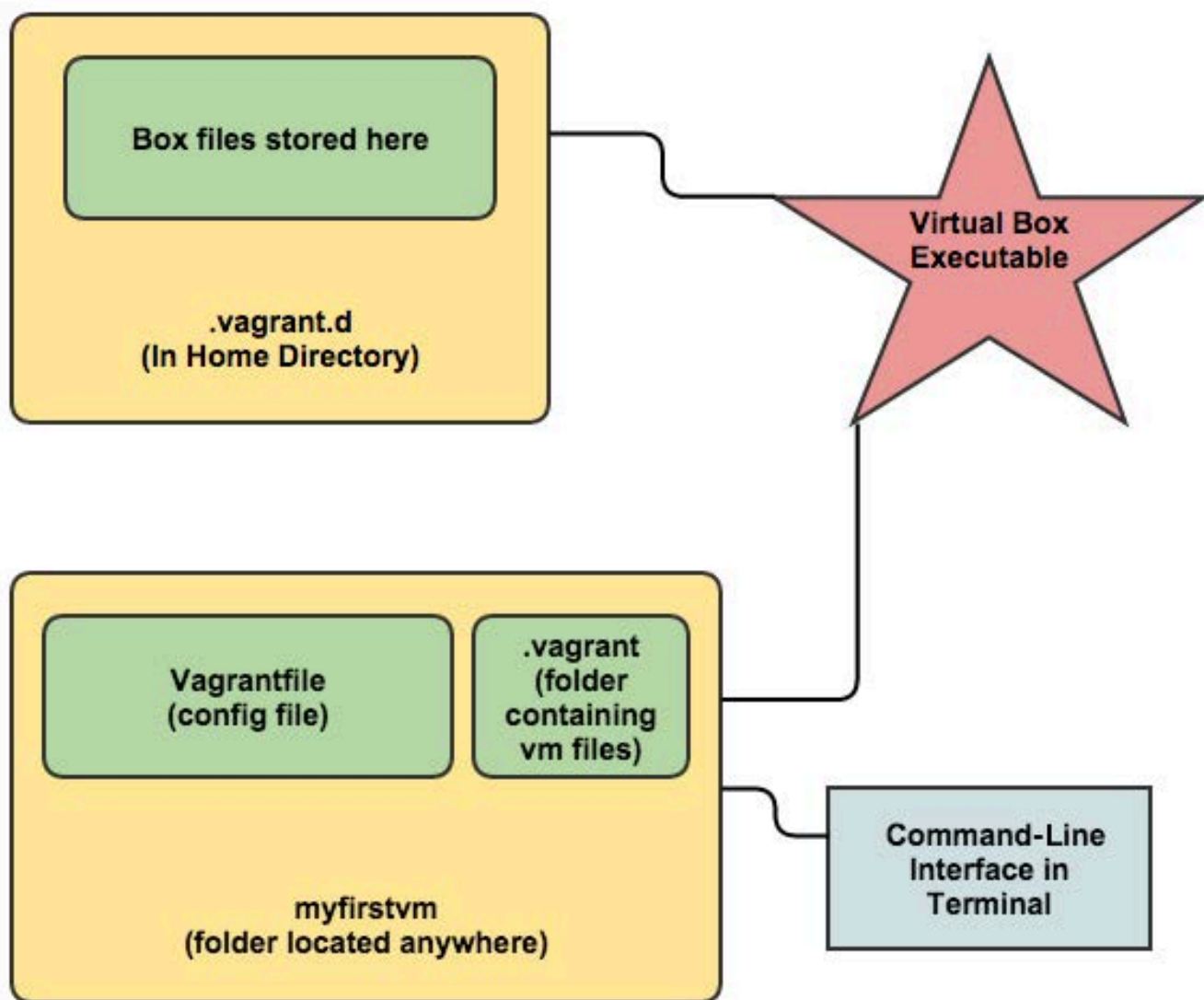


Figure 1. There are really only two locations to worry about: the `.vagrant.d` folder in your home directory and the project folder you create.

There are three main pieces to the Vagrant puzzle. Figure 1 shows a diagram of those parts and their connection to each other.

1) The Virtualization System: By default, this is VirtualBox. It is the engine that makes Vagrant boxes work, but it doesn't keep track of any VM files or configuration. It's a bit like

using Python. The Python executable needs to be installed, but it's just an interpreter that executes the Python code. With Vagrant, it's the same sort of thing. VirtualBox is just the program Vagrant uses to run its own code. If VirtualBox is installed, your work is done.

2) The `.vagrant.d` Folder: This

My favorite feature of Vagrant is that every VM lives inside its own folder.

actually took me a while to figure out. Those “boxes” I mentioned earlier are downloaded to this folder so that when you create a new VM, it doesn’t have to re-download the box, it just uses your local cached copy. Once I knew where the folder was, it was easier to fix things when I messed up. I tend to learn by experimentation, and so I invariably break things. At first, I couldn’t figure out how to get rid of the boxes I incorrectly downloaded, but clearing them out of the `.vagrant.d` folder in my home directory did the trick.

3) The Project Folder: My favorite feature of Vagrant is that every VM lives inside its own folder. Everything to do with the VM is in that folder, so you don’t have to worry about the configuration file being in one folder, the hard drive image being in another folder and so on. The folder can also be created anywhere, and it functions independently from other project folders. They don’t even depend on the original “box” once they’re created, because the box is cloned

into the project folder when you create the Vagrant instance.

The project folder contains the Vagrantfile, which is the single configuration file for the virtual machine. It contains the settings for what type of hardware you want to have VirtualBox use (how much RAM and so on), and it can contain startup scripts that will customize your VM as it is created. In fact, a common thing to do with the Vagrantfile is to start Chef and automatically configure the entire machine from a Chef server! Thankfully, the Vagrantfile can be very simple, and Vagrant creates one for you by default the first time you create a VM.

The Step-by-Step Process

I’m almost to the point where most Web tutorials start, which is to create a VM with Vagrant. Now that you know what Vagrant is doing, the process is far more interesting and less mysterious. Here is the step-by-step process:

1) Install VirtualBox and Vagrant: On a Debian-based distro like Ubuntu,

you simply type:

```
sudo apt-get install vagrant virtualbox
```

and allow the programs to be installed. There's no need to open VirtualBox, just have it installed.

2) Download a Box File: This is how you populate the `.vagrant.d` folder in your home directory with a template or box file. To choose what box you want, head over to <http://vagrantbox.es>, and copy the URL of whatever base you want to start with. Lots of contributed boxes are available, so pick one that makes sense for you, and copy the URL to your clipboard. Then, to get the box to your local cache, type:

```
vagrant box add NAME http://example.com/boxurl/mybox.box
```

Note that `NAME` is just an arbitrary name that you pick to name your box. If you go with an Ubuntu 14.04 image, you could name it something like "trusty64" so you can reference it easily later. The command will download the box file and store it with whatever name you chose.

3) Create a Project Folder: Somewhere on your system, create a folder for the VM. It truly can be anywhere. I usually make it on my

desktop at first, so I easily can see it—basically:

```
mkdir ~/Desktop/myfirstvm
```

Then, switch into that folder:

```
cd ~/Desktop/myfirstvm
```

4) Initialize Your Vagrant Image: Now you simply type:

```
vagrant init NAME
```

The vagrant program will create a Vagrantfile with a default configuration. Basically, it contains information on what box file to use when creating the VM. If you want to do advanced configuration on the Vagrantfile (I'll touch on that later), you can edit it, but by default, it will work with the file untouched.

5) Start Your VM: The VM configuration file has been created in the `myfirstvm` folder, but the VM hasn't been created yet. In order to do that, type:

```
vagrant up
```

Note that you have to be inside the `myfirstvm` folder in order to have this work. Because you could have multiple folders with VMs

(mysecondvm, for instance), the only way to tell Vagrant what VM you want to start is to be in the correct folder. Vagrant now will create a virtual machine based on the box file you downloaded earlier. The virtual hard drive and all configuration files will be stored in a folder called `.vagrant` inside the `myfirstvm` folder (refer back to Figure 1 for details). On the screen, you will see the computer starting, and once it's all set, you have a fully running VM.

6) Do Something with Your VM:

This was another stumbling point for me. I had a running VM, but how to interact with it? Thankfully, Vagrant offers a cool feature, namely SSH. Still inside that `myfirstvm` folder, type:

```
vagrant ssh
```

and you will be logged in via SSH to the running VM! Even though you don't know any passwords on the machine, you always can access it via the `vagrant ssh` command because Vagrant automatically creates a keypair that allows you to log in without entering a password. Once you're logged in, you can do whatever you want on the system, including changing passwords, installing software and so on.

7) Manipulate Your VM:

After

you exit out of your VM, you'll drop back into your local machine inside the `myfirstvm` folder. The VM still is running, but you can do other things with Vagrant:

- `vagrant halt` — shuts down the VM.
- `vagrant suspend` — pauses the VM.
- `vagrant resume` — resumes a paused VM.
- `vagrant destroy` — erases the VM (not the Vagrantfile).

Where to Go from Here

I hope that makes the Vagrant workflow clear. If it seems simple enough, but not terribly useful, that's about all the basics will get you. Although it's cool to be able to create and destroy VMs so quickly, by itself, the process isn't very useful. Thankfully, Vagrant bakes in a few other awesome features. I already mentioned the `vagrant ssh` command that allows you to SSH in to the VM instantly, but that is only the tip of the iceberg.

First, you have the `/vagrant` folder inside the VM. This is a folder that is mounted automatically inside the running VM, and it points to the project folder itself on the

What makes that useful is that you can destroy the VM, create a new VM, and the files in your project folder won't be erased.

main system. So any files you store in "myfirstvm" alongside the Vagrantfile will be accessible from inside the VM in the /vagrant folder. What makes that useful is that you can destroy the VM, create a new VM, and the files in your project folder won't be erased. That is convenient if you want to have persistent data that isn't destroyed when you do a `vagrant destroy`, but it's even more useful when you combine it with the scripting capability of the Vagrantfile itself.

Admittedly, it gets a little complicated, but the Vagrantfile can be edited using basic Ruby language to automate the creation and bootup of the VM. Here's a simple example of an edited Vagrantfile:

```
# Sample Vagrantfile with startup script
VAGRANTFILE_API_VERSION = "2"
Vagrant.configure(VAGRANTFILE_API_VERSION) do |config|
  config.vm.box = "NAME"
  config.vm.provision :shell, path: "startup.sh"
end
```

The only line I added was the

`config.vm.provision` line. The other lines were created automatically when I initially typed `vagrant init`. Basically, what the provision statement does is tell Vagrant that once it starts the VM to execute the `startup.sh` script. Where does it find the `startup.sh` file? In that shared space, namely the "myfirstvm" folder. So create a file in your myfirstvm folder alongside the Vagrantfile itself called `startup.sh`:

```
# This is the startup.sh file called by Vagrantfile
apt-get update
apt-get install -y apache2
```

Make sure the file is executable:

```
chmod +x startup.sh
```

Then, have Vagrant create a new VM. If you haven't "destroyed" the existing VM, do that, and then type `vagrant up` to create a new one. This time, you should see the machine get created and booted, but you also should see the system download and install Apache, because it executes the `startup.sh` file on boot!

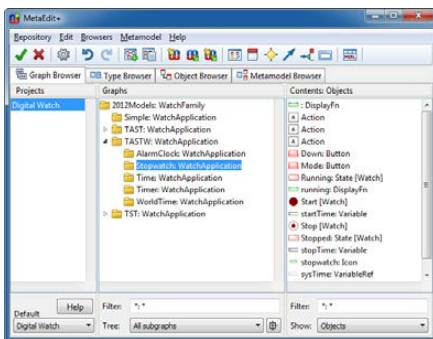
Acromag's ARCX box

The new ARCX box is the solution that gives users in the military/aerospace industries the mission computer they truly want, proffers industrial I/O specialist Acromag.

ARCX box is Acromag's new, small-form-

factor mission computer with Intel Multi-Core CPU that is also rugged, customizable and conduction-cooled. Targeted applications include vetronics, C4ISR, payload management, and command and control. The computer's unique expandable features include PMC, XMC, mini PCIe, mSATA module slots, optional front I/O panel and secondary connectors, and it was engineered rugged with Size, Weight and Power (SWaP) to address space requirements of vehicle electronics. Compatible with industry standards, manufactured to IP67 standards and shock-and-vibration tested to MIL-STD-810G, the ARCX box computers are available either as single or double PMC/XMC slot versions.

<http://www.acromag.com>



MetaCase's MetaEdit+

The new v5.1 release of MetaEdit+ from MetaCase adds a wide range of new features to the company's flagship domain-specific modeling and code generation tool.

MetaEdit+ 5.1 is aimed at expert developers who seek to generate efficient, complete code directly from domain-

specific models. MetaCase notes the advantages of MetaEdit+, which gives language engineers the means to create graphical domain-specific languages and code generators in a few hours. The new MetaEdit+ offers collaboration for both language creation and use: multiple team members can define domain-specific modeling languages together and share them instantly to the whole team. As a result, domain-specific languages created by different language engineers can be integrated, tested and shared easily. MetaCase further notes the advantages of collaboration in language development, for which MetaEdit+ 5.1 offers "unparalleled support", for improving both the quality and acceptance of domain-specific languages. Versions of MetaEdit+ are available for Linux, Windows and Mac OS X.

<http://www.metacase.com>



Panasas ActiveStor

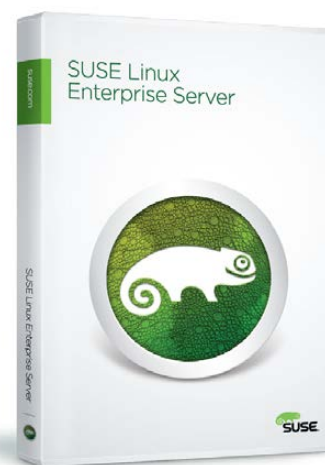
Though the claim about its new ActiveStor 16 appears counterintuitive, Panasas says it's real. The company promises that the ActiveStor 16 hybrid scale-out NAS appliance delivers an industry first: performance that increases with scale alongside enterprise-grade reliability that improves at scale. With help from the company's most advanced storage operating system release to date, PanFS 6.0, ActiveStor eliminates the fundamental compromises between performance, capacity and reliability that storage users previously have come to accept, states the company. The means to reach the increased performance results from the ActiveStor 16 appliance's 50% increase in storage density and from PanFS 6.0's delivery of RAID 6+ triple-parity data protection for a 150x increase in reliability over dual-parity products.

<http://www.panasas.com>

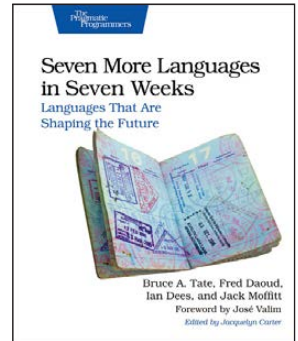
SUSE Linux Enterprise

More than six years have passed since the last major release of SUSE Linux Enterprise, but the company says that it is worth the wait. The SUSE Linux Enterprise 12 platform release, with its array of operating systems and extensions, is now available. The most notable innovations are to be found in SUSE Linux Enterprise Server for x86_64, IBM Power Systems and IBM System z. A feature sampling includes full system rollback, live-kernel patching ability, software modules, integration of both BTRFS and XFS filesystems, more advanced Linux containers technology and the Docker framework as an integral part of the OS. Also noteworthy is the updated customer portal, the SUSE Customer Center, which features a new dashboard that simplifies subscriptions, access to patches and updates and communication with SUSE customer support. Finally, SUSE integration with Microsoft now will be easier as well thanks to a new Virtual Machine driver pack for Windows servers. The upshot of these and many other additions, says SUSE, is that the new platform helps enterprises stay agile, reclaim budget and easily leverage future open-source innovation, helping them compete more effectively.

<http://www.suse.com>

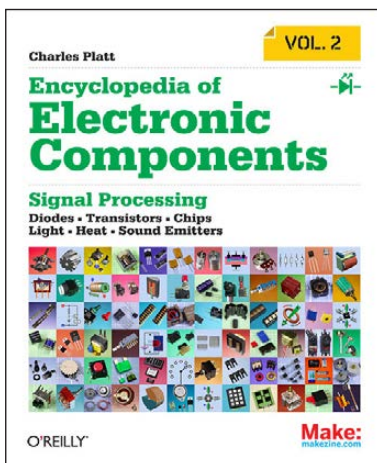


Bruce A. Tate, Fred Daoud, Ian Dees and Jack Moffitt's *Seven More Languages in Seven Weeks* (Pragmatic Programmers)



The publisher Pragmatic Programmers compares programming languages to spoken languages. Both can make you smarter and give you new tools and abstractions to address problems that come your way. They add that if you make the commitment to read books like *Seven More Languages in Seven Weeks: Languages That Are Shaping the Future*, you will experience profound change in how you perform. And how you perform over time will require “radical improvement”, because the industry is changing in profound ways—that is, from object-oriented to functional languages. The publisher says that with this book, it aims higher than simply a “Hello, World” treatment, instead taking readers on a step-by-step journey through the most important paradigms of our time. This profound journey occurs through exploration of the Lua, Factor, Elixir, Elm, Julia, MiniKanren and Idris languages.

<http://pragprog.com>



Charles Platt's *Encyclopedia of Electronic Components, Volume 2* (Maker Media/O'Reilly Media)

If you are anything like the geek writing this blurb, you see books like Charles Platt's *Encyclopedia of Electronic Components, Volume 2*, and you feel an insatiable instinct to browse. This second book of a three-volume set covers signal processing, including LEDs, LCDs, audio, thyristors, digital logic and amplification. Photographs, schematics and diagrams

are included. Readers will learn what each component does, how it works, why it's useful and what variants exist. No matter how much readers know about electronics, says publisher Maker Media, they'll find fascinating details they've never come across before. The *Encyclopedia* is targeted at teachers, hobbyists, engineers and students of all ages and abilities who seek reliable, fact-checked information right at their fingertips.

<http://www.oreilly.com>



Thinlabs Device Manager

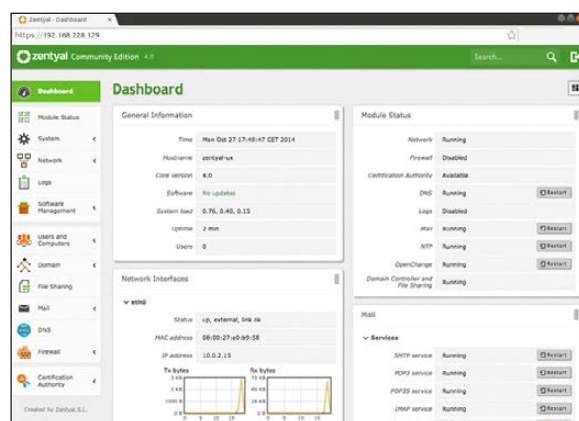
Administering your thin-client or PC environment is the job of the upgraded Thinlabs Device Manager 4.0, a solution for Linux and Windows embedded systems. TDM 4.0 features an intuitive graphical interface powerful enough to manage both PCs and thin clients over a LAN, large multi-site WAN or isolated devices behind a NAT firewall router. Advances vis-à-vis the previous edition include remote script deployment and execution, automatic provisioning of new or existing devices using profiles and silent software distribution. TDM 4.0 is included with all Thinlabs integrated thin clients and thin-client computers or is also available on a per-site or per-seat license model with non-Thinlabs products.

<http://www.thinlabs.com>

Zentyal Server

What is slick about Zentyal Server 4.0 is that it integrates both Samba and OpenChange technologies, enabling native support for mixed IT environments that include Linux, Windows and Mac OS clients, as well as mobile devices with ActiveSync. Therefore, Zentyal Server 4.0 is able to offer full native Microsoft Outlook compatibility without plugins or connectors. Zentyal says that the Small Business Server can be set up in less than 30 minutes and is both easy to use and affordable. Complementary to the version 4 release's mail and mail-related directory features, other improvements also have been added, such as a restructured and improved L2TP module, free configuration backup in the cloud via the Zentyal Server UI and quality-assurance processes that improve the stability of product releases.

<http://www.zentyal.org/server>



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.

How to Perform an **INTERNAL SECURITY REVIEW**

How does your network stack up
to potential security threats?
You won't know until you look.

JERAMIAH BOWLING

One of the most persistent themes in the past decade of computing has been security. When asked what concerns them most, IT managers consistently list security in their top three answers. It *should* concern them. In this year alone, we've seen numerous data breaches hit big companies, and we've had the disclosures of the Heartbleed and Shellshock vulnerabilities. Prognosticators say more vulnerabilities are on the way. One way to make sure your house is in order is to perform an internal security review—a multi-pronged process that discovers and documents what is on your network and what is in place to protect your users and your data.

In this article, I walk through performing a very basic security review of a fictitious company and produce a report of the findings and recommendations for any deficiencies discovered. Before starting, let's look at the reasons for performing a review.

For one, a healthy paranoia is a good quality for any security professional to have. You want to know what is on your network, what it does and how it operates. Two, you may be required to do audits or subject to audits based on regulatory

requirements (PCI or HIPAA) for your industry. If your company is subject to an outside compliance audit, an internal review is an excellent way to prepare for a real one. It shows potential auditors that you are prepared and proactive against security threats. Finally, it can identify weaknesses in your current defenses so that you can address them prior to an incident. If there is any time to have an honest discussion about your company's approach to security, it is well before an audit, or worse, before a breach takes place.

Define the Scope

Let's kick off the process by defining the scope. Start with identifying the drivers behind your review. Are you doing it as a best practice or are there any regulatory requirements you need to satisfy? If the answer to the latter is yes, there may be existing guidance or checklists specific to that standard you will want to consult. If the answer is no, who is your audience? Is it departmental? Is it upper management? Know the answer so you can tailor your report effectively for the best outcome.

If you are presenting the report to decision-makers, you may have influence over budget and personnel

decisions. If it is being done for the IT department, you may want to highlight technical issues that affect day-to-day operations.

Next, decide how you want to collect the data needed for the review. Are you limited in scope by time, personnel, expertise or budgeting concerns? How do you take all of the information and craft it into a coherent report? There are a multitude of questions to answer before getting started, and your mileage may vary. Make sure to document your planned scope for the report and come to an agreement with decision-makers before performing any other work.

For my example company, I'm limiting the work to the tools included in this article and the portions of the sample environment that they will test. I have opted to use several industry-standard tools found on the Kali Linux (formerly Backtrack) distribution. My fictional bosses have specified that I perform less-intensive data-gathering tasks (scanning, discovery) during production hours and save the vulnerability scans for overnight due to the traffic generated and possible side effects on client machines. Below is the sample scope that I will follow for the rest of the process.

Sample Scope:

1. Gather internal documentation regarding network and systems.
2. Perform host discovery on production networks during normal business hours.
3. Perform network traffic baseline testing during production and after hours.
4. Perform an inventory scan of all found hardware and software.
5. Perform a vulnerability scan on workstations and servers.

If you have multiple people involved in the day-to-day operations of your IT group, the review should be prepared as a team. The review process can be invaluable as a cross-training and knowledge-sharing tool.

Data Gathering

After the defining scope, let's move on to the process of data gathering. I want to collect as much existing information about my enterprise to give the most complete picture possible. This information can come from many sources both on-line and off-line. I'm going to collect any network drawings,

server build documents, operating procedures, policies and so on that pertain to daily operations or security for analysis with later findings. It may seem odd to include policies, but often security directives can come in the form of non-technical controls. If you aren't sure what constitutes a policy, they are usually in the form of a handbook or written document/memo. Sometimes they are simply a verbal or known policy. If there are such policies, be sure to document them. It's always better to have a policy than not have one. It often happens in the course of the review process that a known or verbal policy becomes a documented policy. I also will want to gather information about any or all hardware, software, network devices and their placement in the enterprise. This will help me identify any unusual and/or unauthorized devices found in the next step.

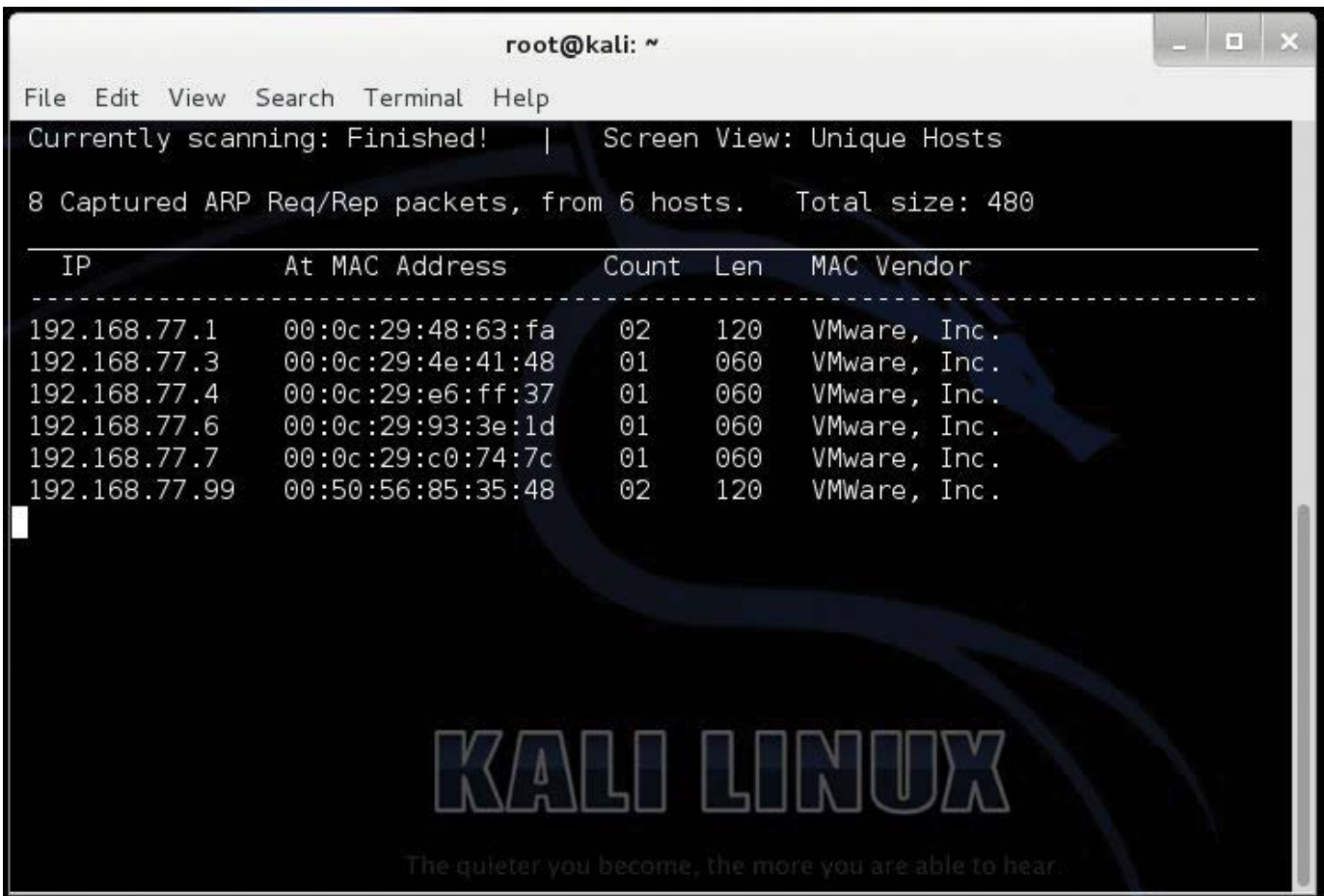
Network and Host Discovery

Now I'm going to move from collecting data to generating it. I do this in order to compare what I find to what actually is on our network as well as to create a list of hosts for our vulnerability scans. Here again I have some decisions to make. What do I want to scan and how deep? The longer you scan, the more information

you generate. There are many types of scans you can perform, but you always need to be aware of the cost and labor. Sometimes a penetration test is worth it, but it may have to be done after hours. Scanning tools can break an application or OS with the wrong options selected. How much data are you generating and where do you store it? As has been the pattern, document these decisions for the final report.

To get started, let's generate a listing of the devices on the network. There are multiple tools to do this, but the easiest to get going is netdiscover. Launch your Kali workstation and click on the Applications menu, then Kali Linux→Information Gathering→Network Scanners→netdiscover. This will launch a terminal prompt. Type `netdiscover -r YOURIPSEGMENT/SUBNET` (for example, `192.168.77.0/24`) to scan the local subnet (Figure 1).

You can scan entire ranges and classes, but to save time let's limit the results to known subnets. You also could use nmap/Zenmap, which also is included in Kali, to discover as well as scan ports and create basic maps. Run netdiscover on each of your subnets and pipe the results to a text file. If you share the same address classes, you could try running the program without any switches, which will look at everything



```
root@kali: ~
File Edit View Search Terminal Help
Currently scanning: Finished! | Screen View: Unique Hosts
8 Captured ARP Req/Rep packets, from 6 hosts. Total size: 480
-----
IP                At MAC Address    Count  Len  MAC Vendor
-----
192.168.77.1      00:0c:29:48:63:fa  02    120  VMware, Inc.
192.168.77.3      00:0c:29:4e:41:48  01    060  VMware, Inc.
192.168.77.4      00:0c:29:e6:ff:37  01    060  VMware, Inc.
192.168.77.6      00:0c:29:93:3e:1d  01    060  VMware, Inc.
192.168.77.7      00:0c:29:c0:74:7c  01    060  VMware, Inc.
192.168.77.99     00:50:56:85:35:48  02    120  VMWare, Inc.
-----
KALI LINUX
The quieter you become, the more you are able to hear.
```

Figure 1. netdiscover Searching for IPs

in the class range (for example, 192.168.77.0/16). When you have run netdiscover on each of your subnets, combine and filter the text files into one file that contains all the IP addresses you want to scan for vulnerabilities. Compare your findings with any documents/drawings collected in the previous section to detect systems not included in those items. Note any exceptions found for the report.

Next, you're going to take some sample captures of network traffic to check for any unusual activity.

I've found more rogue activity on networks using sniffers and IDS systems than I have with any other tools. You also can use sniffers to troubleshoot faulty or misconfigured applications as unauthorized channels. You will want to take captures at both peak times and off hours to get a feel for what traffic is traversing your network and establish a baseline of normal flow.

In Kali, click on Applications→Kali Linux→Sniffing/Spoofing→Network Sniffers→Wireshark. Wireshark will

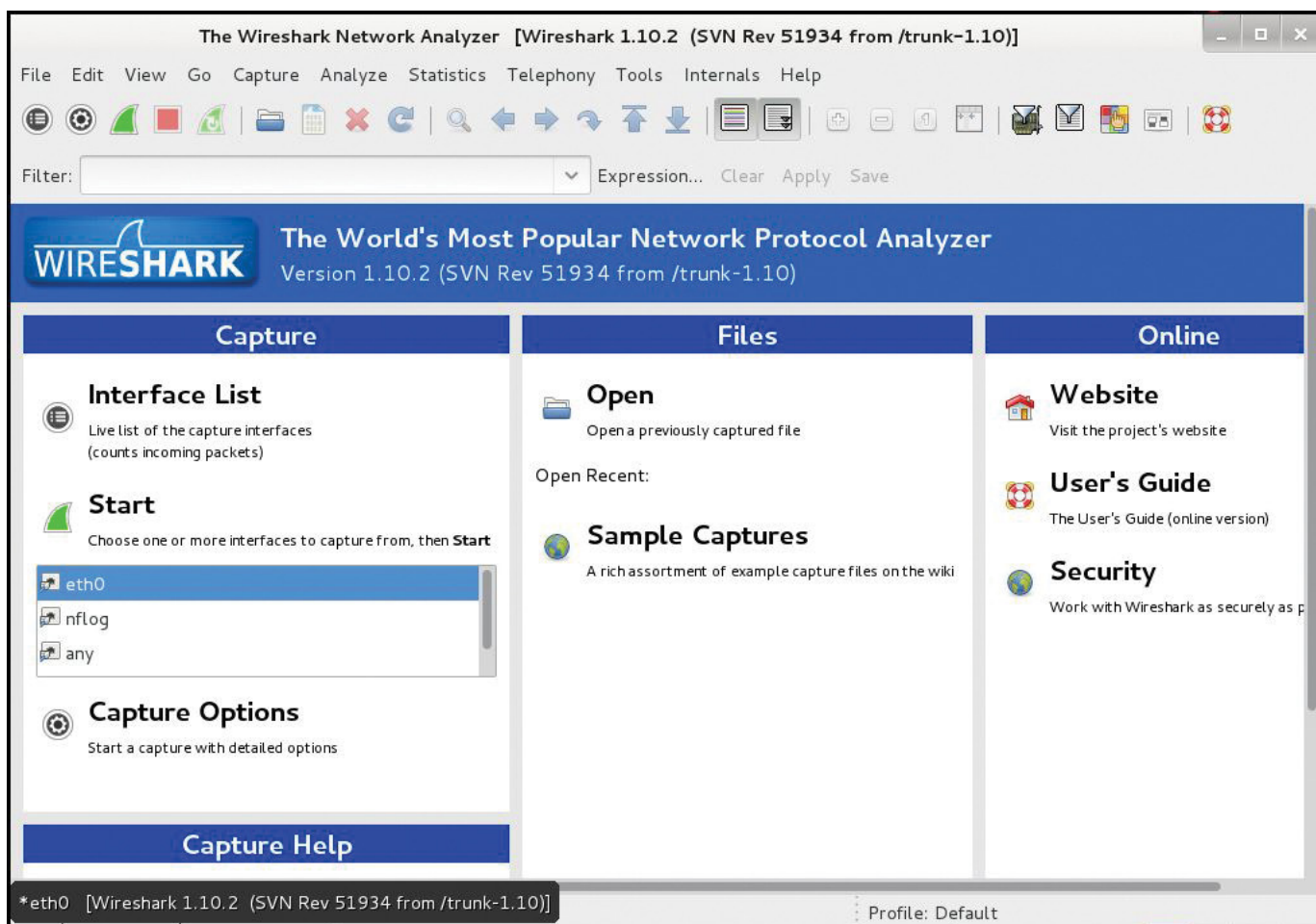


Figure 2. Wireshark Start Screen

start with its basic screen (Figure 2). Select the interface you want to listen on and click Start. The default setting on Kali is set to promiscuous mode, but you may need to do further configuration if you want a more thorough capture. Allow the capture to run for as long as you deem necessary. This may depend on the size of your subnet, bandwidth and storage. My rule of thumb is to take samples of 5–10 minutes at peak times (morning,

middle of the day).

Review your captures and look for items of big concern. Are there any protocols being used that you don't recognize or that are unauthorized? Are there any IPs or traffic-related behavior that look out of place? Make sure to note any unusual findings for the final report. In my test capture (Figure 3), I can see an abundance of traffic between one of my Windows clients another external host. As I inspect the packet, I see

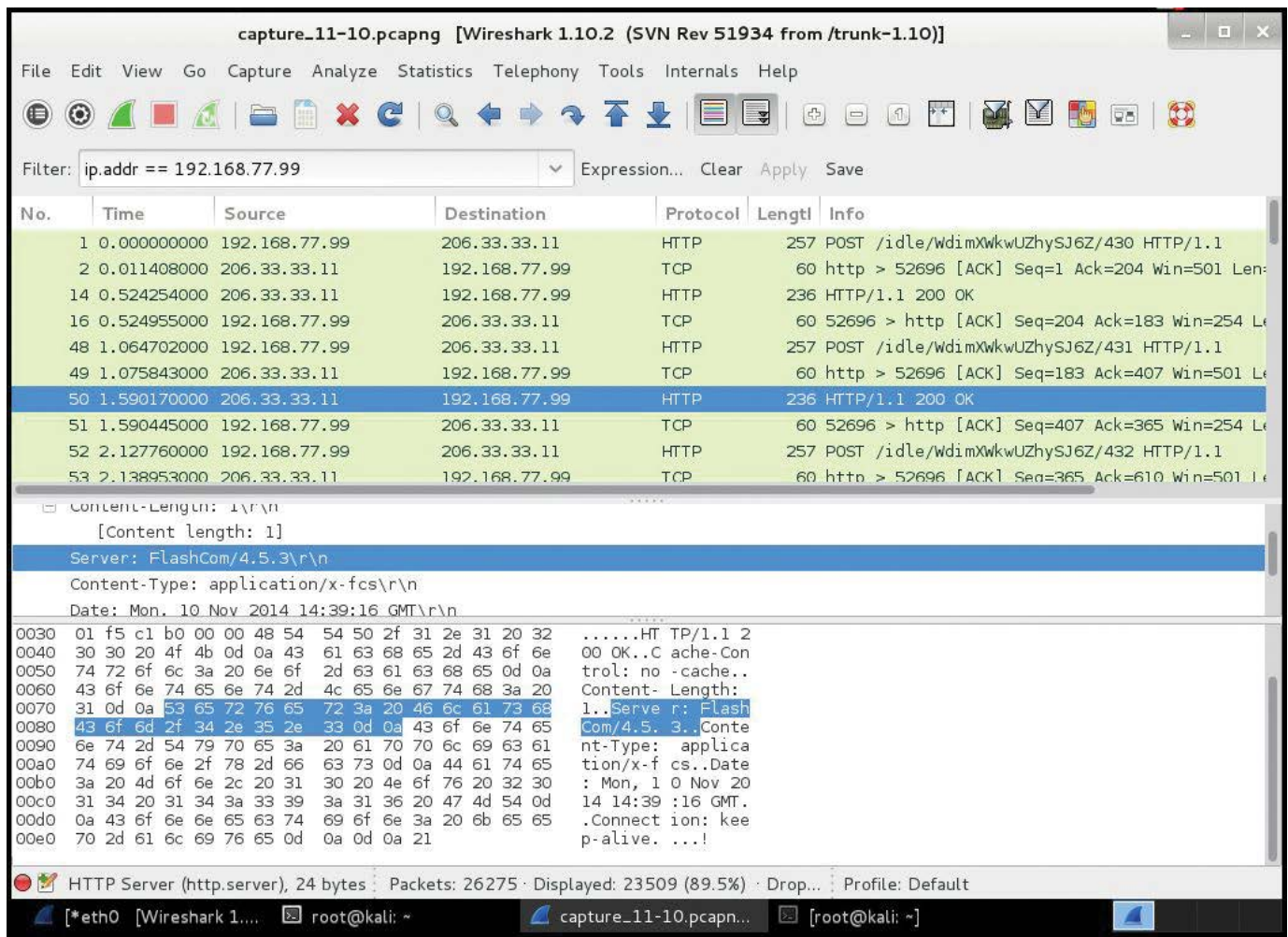


Figure 3. The results of my scan—looks like I found some Flash traffic.

it is connecting to a Flash-based server. It turns out my test user is using a streaming service from his workstation. This could be allowed or it might not be. Either way, Flash player has plenty of vulnerabilities to exploit. As you review the captures, note any exceptions to normal or expected traffic.

Inventory

This step may or may not be required

for what you need. I consider it a best practice. Just like it is a good idea to know everything that is running on your network, you also should maintain an inventory of your hardware and software. Plenty of software exists that can do this in an automated fashion, but most of it is commercial. Let's use Open-Audit here, an open-source alternative. Installation is easy, and the program is straightforward to use. I deployed

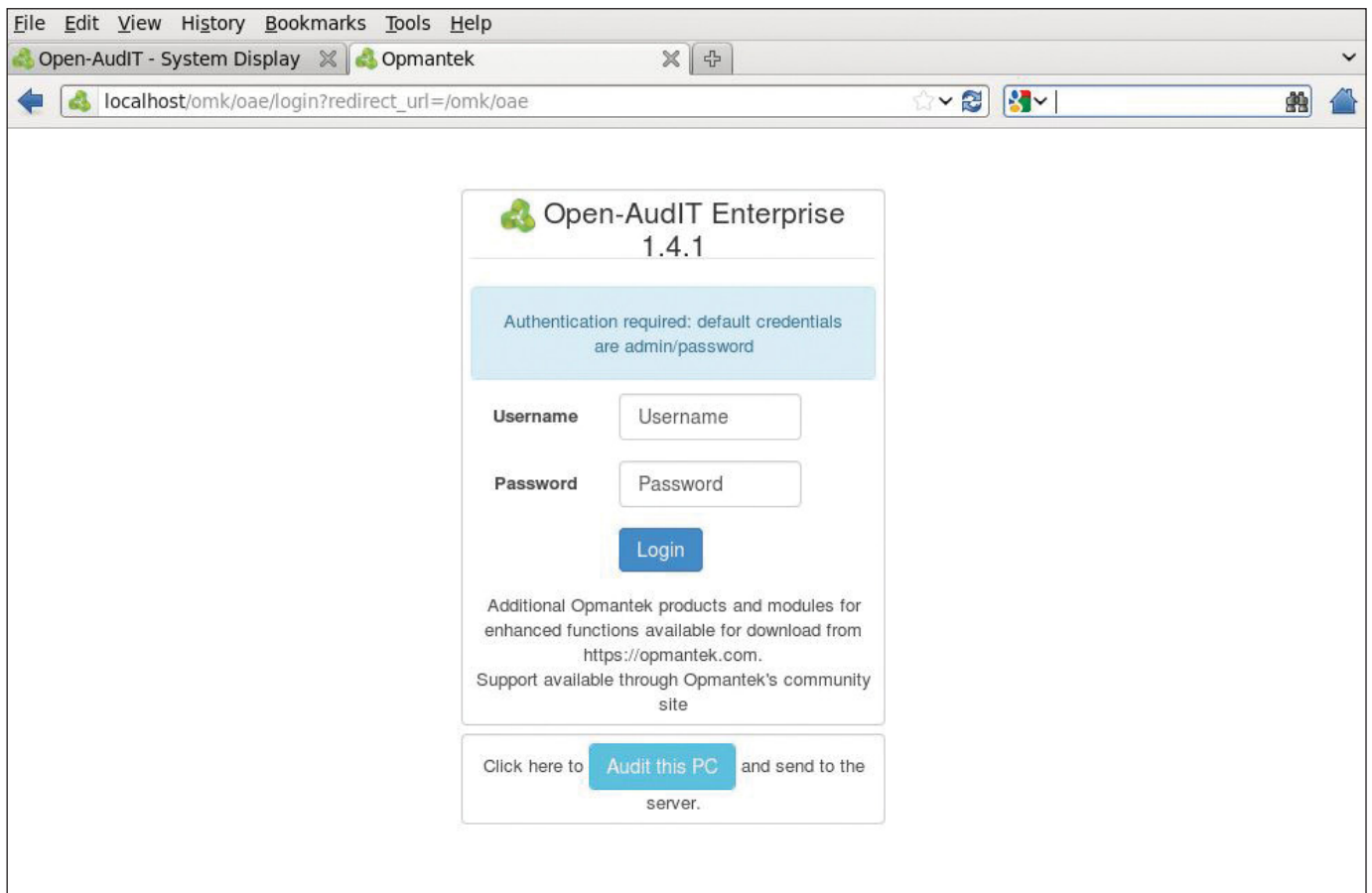


Figure 4. Open-Audit login screen—you also can use the “Audit this PC” button to inventory the client.

Open-Audit on a CentOS 6 server, as it unfortunately is not included in Kali.

Once installed, you can use a Web interface to manage the program. You can use some of the automated methods included in the program (SSH, SNMP) to add systems, or you simply can visit the server’s Web page and use the “Audit this PC” button from a client machine (Figure 4).

Once the inventory process is completed, review the findings and look for any unexpected software

installations (Figure 5). Some users love to install their own software. This can be a problem, because you can’t update or patch what you don’t know is there. As before, note any findings of concern related to hardware or software you encounter, specifically those you discover vulnerabilities for in the next section.

Vulnerability Scans

Now let’s move on to the task of vulnerability scanning your systems.

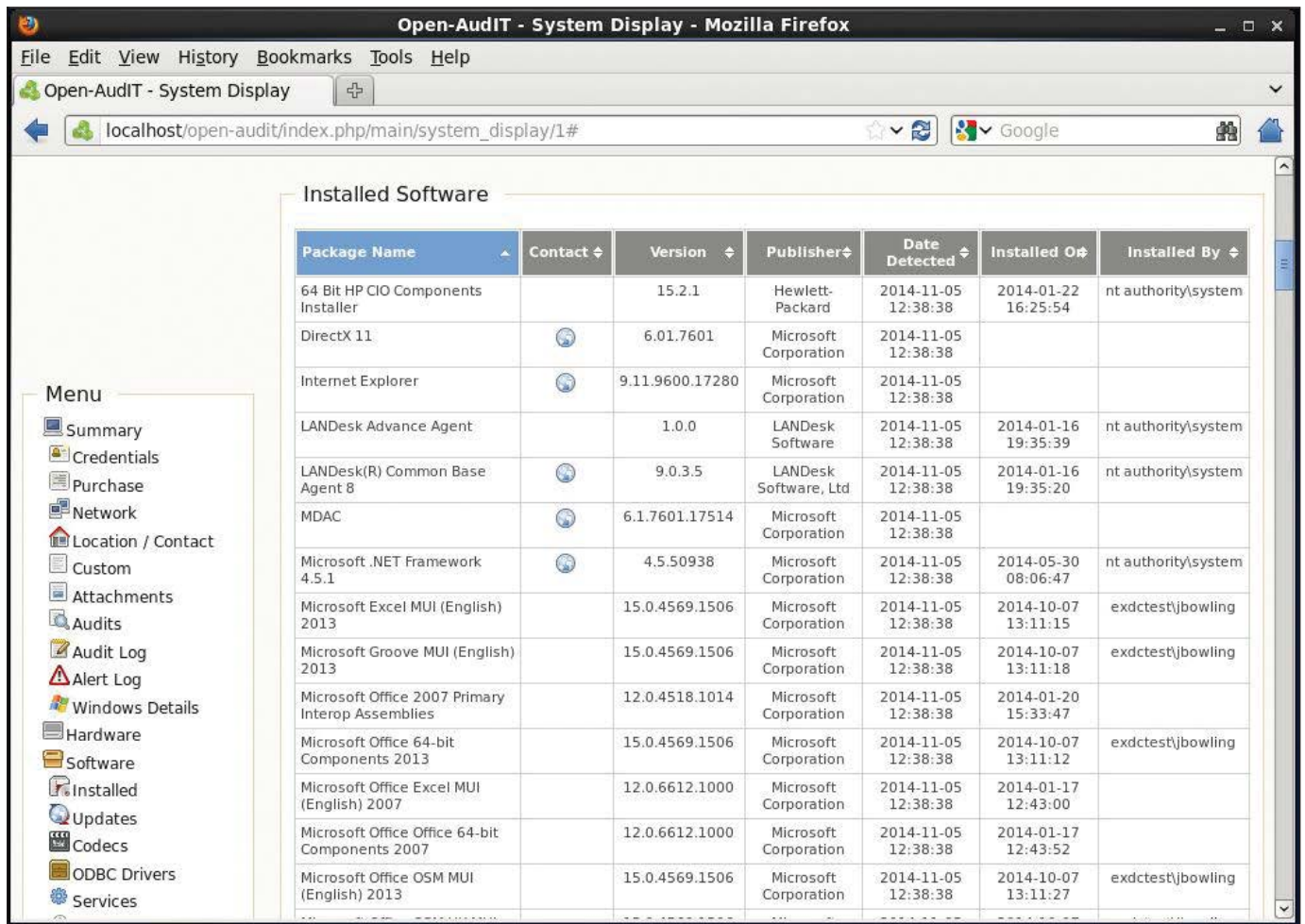


Figure 5. Inventory of Found Software

Let's use an open-source tool called OpenVAS (Vulnerability Assessment System) for the scans. OpenVAS can be notoriously difficult to install, but on Kali, it is a snap, and should you run into any install issues, there is plenty of information on the Internet.

In Kali, click on the Applications menu, then mouse over Vulnerability Analysis→OpenVAS→"openvas initial setup". This will start the install

process. At the end of the setup, an administrator login and password are displayed. Save these for later use.

After the install is complete, come back to the same section and select "openvas start". After a few seconds, load a browser and navigate to <https://yourkailipaddress:932>. Log in with the admin account and the password you recorded during set up. Ignore the welcome screen, as you will save time by performing a bigger scan.

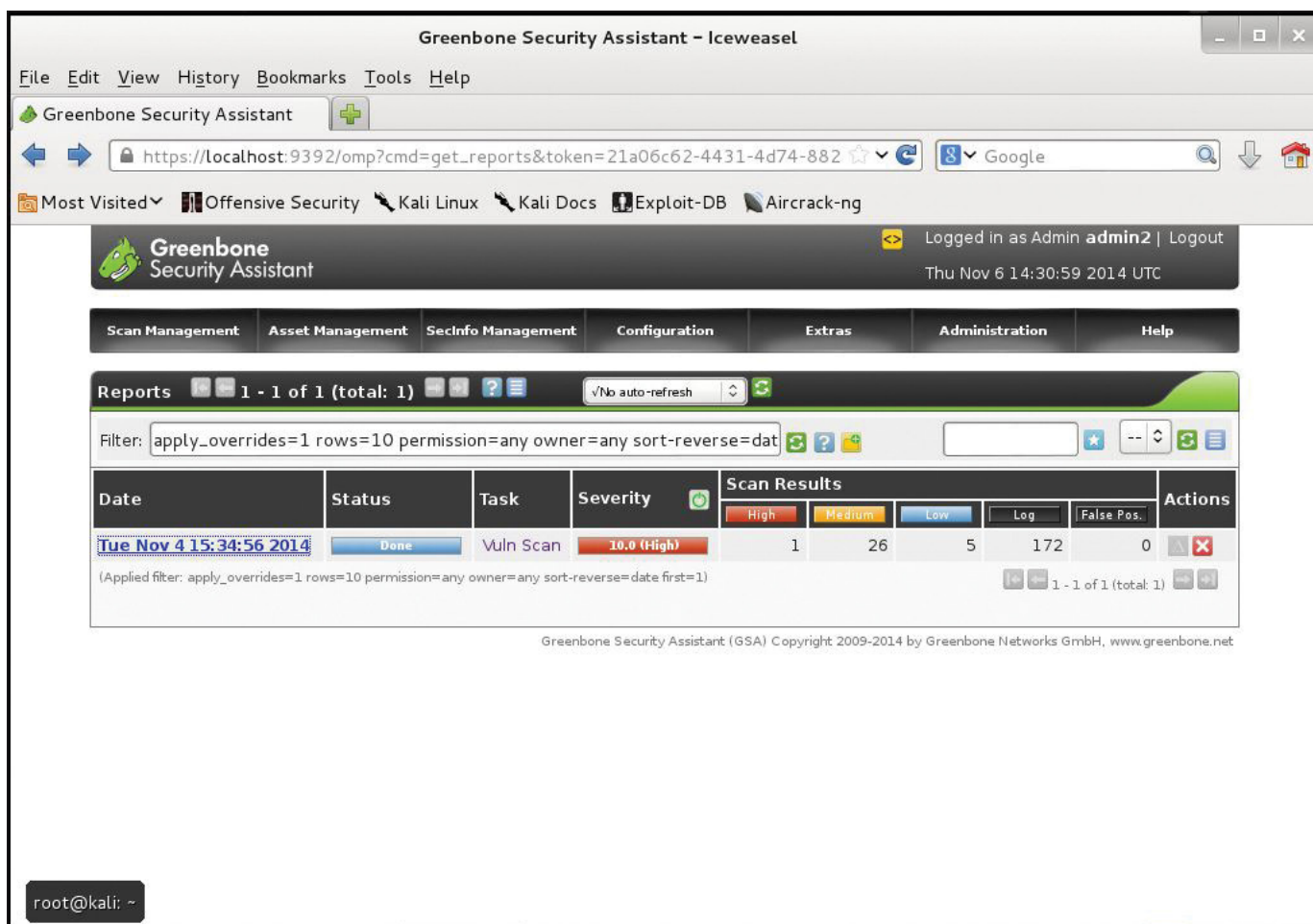


Figure 6. The Results of the OpenVAS Scan

Click on the Configuration→Targets link. On the Targets page, click on the white star in the menu bar to create a new target. Enter a name for the Scan and any comments. Click on the browse button to find the text file with the list of hosts generated from netdiscover. Under Port List, select “OpenVAS Default”. In the “Alive Test” drop-down, select Consider Alive. Click Save Target. Now click on the Scan Management link in the

menu and then click on Tasks. On the page, click on the star in the menu again to create a new task. Give the scan a name and add any comments. In the “Scan Config” section, select “Full and fast”. Under Targets, select your target. Check the box marked yes next to “Add results to Asset Management”, and click on Create Task. This takes you back to the Task Details page. Click on the green play button in the menu bar to start the

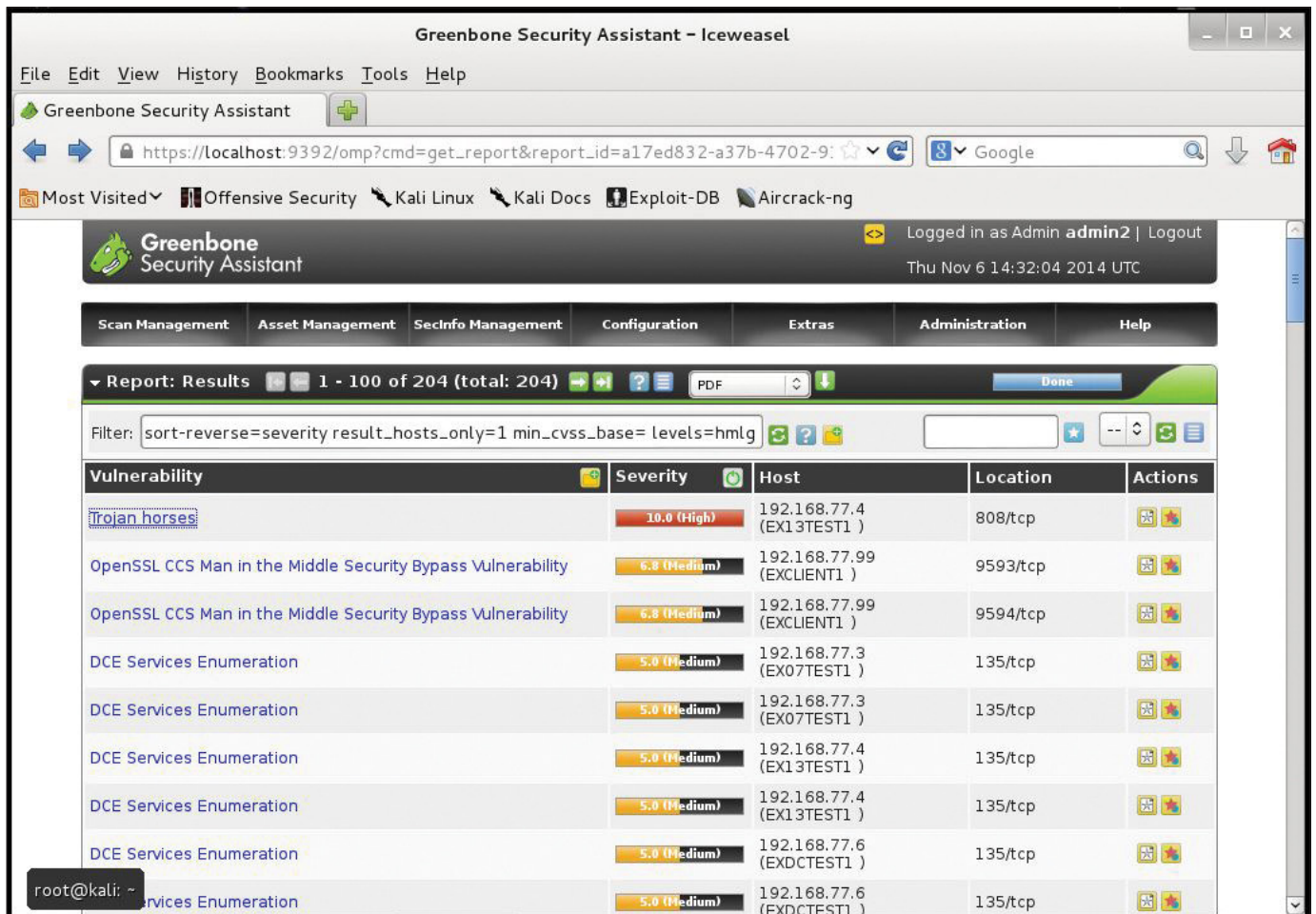


Figure 7. Vulnerabilities Found in the Scan

task. You can set the refresh rate of the scan status in the drop-down in the middle of the page. I set ours to 60 seconds. This scan will take a while (depending on the number of hosts scanned). When the scan is complete, OpenVAS will generate a report.

If you navigate back to Scan Management → Reports, you should see your report listed (Figure 6). As you can see, the scan found a high severity vulnerability (Figures 7 and 8).

This turned out to be a false positive from my test Microsoft Exchange server. However, there still were several medium-level vulnerabilities that should be addressed. Review the rest of the scan results and note any items of concern. Save the report to a separate location when you are done.

OpenVAS does a nice job of exporting formatting reports in .pdf (Figure 9). As you analyze the report, make your own determinations of

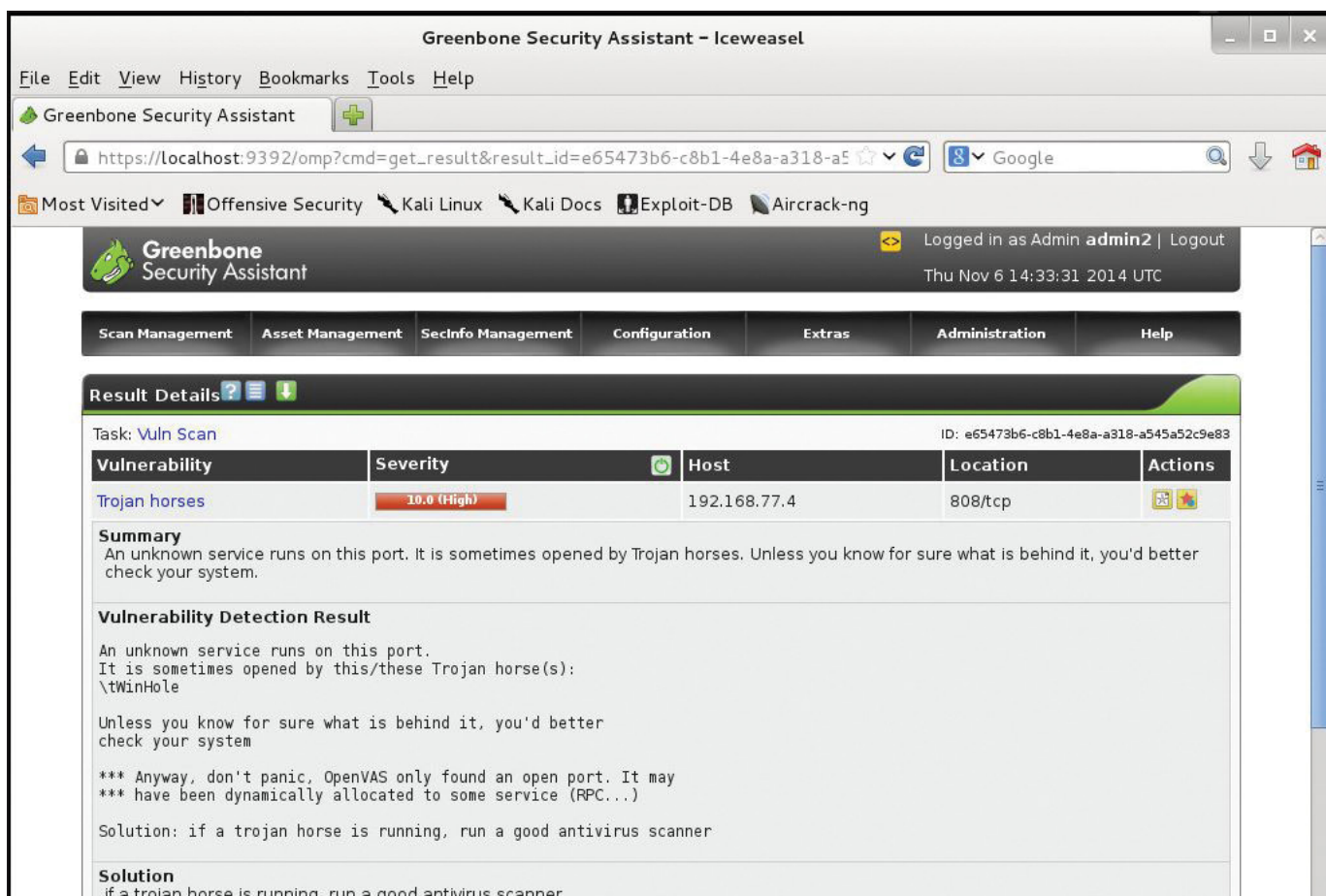


Figure 8. Description of One of the Found Vulnerabilities

risk and assign a level you feel is commensurate with the vulnerability. Use labels you feel comfortable with (High, Medium, Low and so on). You know your enterprise better than anyone, so you should feel comfortable assigning risk to any found vulnerabilities on your systems. Although "risk" is an immensely broad topic, if you don't have a lot of experience assigning it, you always can use the commonly accepted risk formula to assign levels:

$Risk = Threat \times Vulnerability \times Impact.$

In a nutshell, the formula breaks down to what is the threat (hackers, malicious employees and so on), times how serious is the vulnerability (high/well known, easy to execute), times the damage done in money, reputation or data loss if the event occurs. Make sure you differentiate between your defined levels of risk and the severity of a vulnerability. For example, if you discovered a medium severity vulnerability on your



Figure 9. The OpenVAS Scan Report in PDF Format

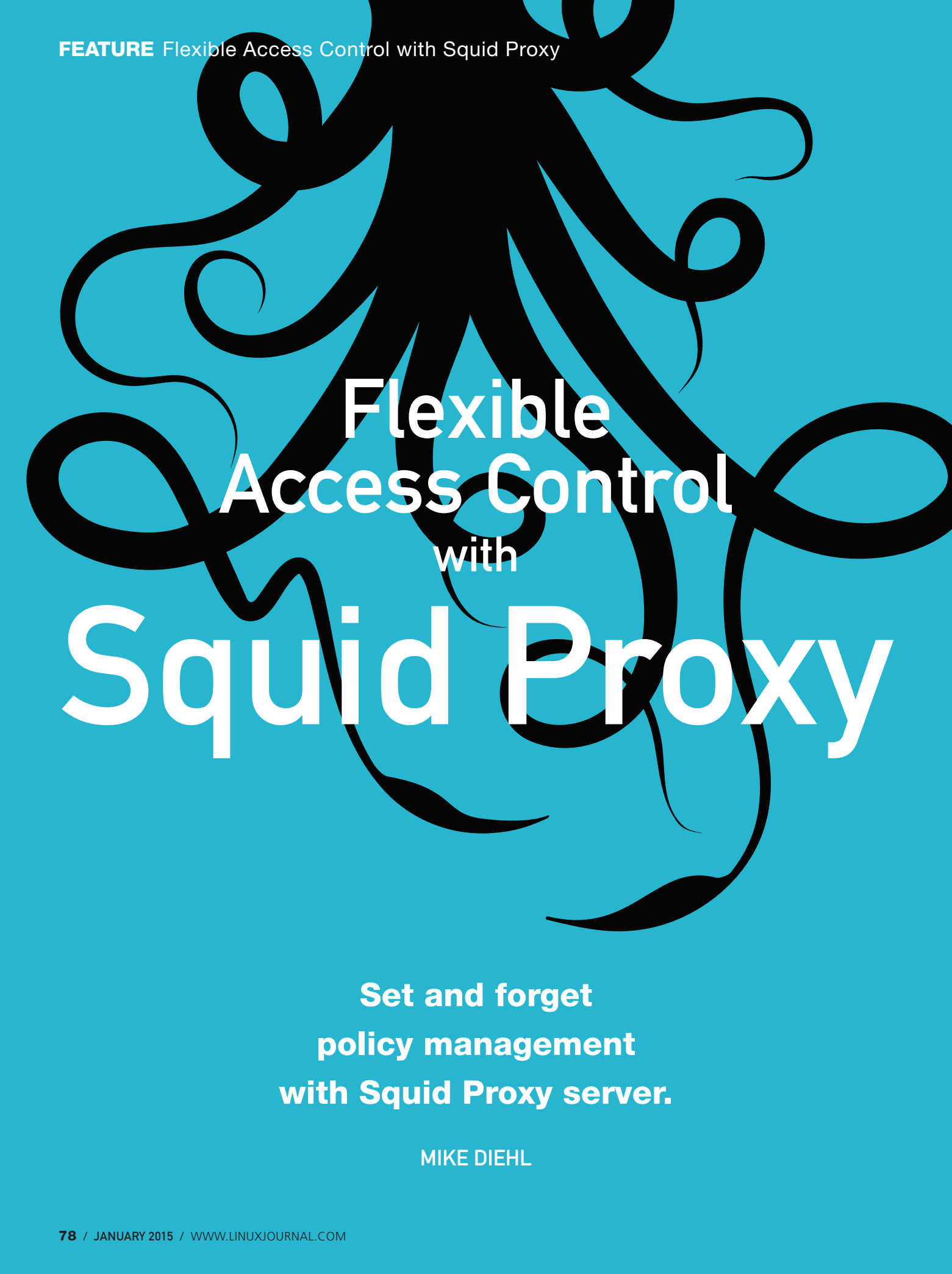
public-facing Web server, because it is exposed to a public network and your company’s reputation would take a hit if the site is defaced, the risk is higher than if this were an internal Web server with the same vulnerability, but accessible only to employees. So you would classify the vulnerability as High to the external Web server and Medium (or low) to an internal server.

Writing the Report

Now you’ve come to the all-important task of organizing the final report. If

you’ve been documenting the review process as you went along, it should be a straightforward task. Take the data you have collected and analyzed, and organize it into a coherent, flowing document. I have provided a basic outline, but I’ve included links to some on-line sample reports in the Resources section that are broader and far more detailed. Any report you write should include the following sections at a minimum:

1. Executive summary: defines the purpose of the report and your



Flexible
Access Control
with
Squid Proxy

**Set and forget
policy management
with Squid Proxy server.**

MIKE DIEHL

Large enterprises and nuclear laboratories aren't the only organizations that need an Internet access policy and a means of enforcing it. My household has an Internet access policy, and the technique I've used to enforce it is applicable to almost any organization. In our case, I'm not too concerned about outside security threats. Our network is behind a NAT router, and our Wi-Fi has a ridiculously ugly password. Our workstations are either Linux or properly patched Windows machines (if there is such a thing). No, our concerns come from inside our network: our kids like to play Web-based games, and that often gets in the way of chores and homework.

We're also concerned they might stumble upon Web content that we'd rather they not access. So no, we're not protecting nuclear secrets or intellectual property, but we are enabling the household to run smoothly without undue distractions.


In general, my wife and I don't care if our kids play games on-line or stream media. But, if their homework or chores don't get completed, we want a means of "grounding" them from this content. The problem is that we also home school, and much of their educational content is also on-line. So, we can't simply block their access. We need something a bit more flexible.

When I set out to solve this problem, I made a list of the goals I wanted to accomplish:

1. I don't want managing my kid's Internet access to become a full-time job. I want to be able to set a policy and have it implemented.
2. My wife doesn't want to know how to log in, modify a configuration file and restart a proxy daemon. She needs to be able to point her browser, check a few boxes and get on with her life.
3. I don't want to write too much code. I'm willing to write a little bit of code, but I'm not interested in re-inventing the wheel if it already exists.
4. I want to be able to enforce almost any policy that makes sense for our household.
5. I don't want anything I do to break their Internet access when they take their laptops outside the house.

I'm sure my household isn't the only organization interested in these results. However, I made an assumption that may not make sense in other organizations: my kids won't be taking any sophisticated measures

My code will tell the proxy server how to handle each request as it comes in. The proxy either will complete the request for the user or send the user a Web page indicating that the site the user is trying to access has been blocked.



to circumvent our policy. However, I do reserve the right to participate in the arms race if they do.

For the purpose of this article, anytime this assumption leads to a configuration that may not make sense in more sophisticated environments, I'll try to discuss a few options that will allow you to strengthen your configuration.

I wasn't able to find any single software package that was flexible enough to do what I wanted and also easy enough to use, so that it wouldn't take considerable effort on the part of my wife and me to employ it. I was able to see that the Squid proxy server had the potential of doing what I wanted with just a little bit of coding on my part. My code will

tell the proxy server how to handle each request as it comes in. The proxy either will complete the request for the user or send the user a Web page indicating that the site the user is trying to access has been blocked. This is how the proxy will implement whatever policy we choose.

I've decided that I want to be able to give my family members one of four levels of Internet access. At the two extremes, family members with "open" access can go just about anywhere they want, whereas family members with "blocked" access can't go anywhere on the Internet. My wife and I will have open access, for example. If one of the boys is grounded from the Internet, we'll simply set him as blocked.

However, it might be nice to be able to allow our kids to go to only a predetermined list of sites, say for educational purposes. In this case, we need a “whitelist-only” access level. Finally, I’m planning on a “filtered” access level where we can be a bit more granular and block things like music download, Flash games and Java applets. This is the access level the boys generally will have. We then can say “no more games” and have the proxy enforce that policy.

Because I don’t want to write an actual interface for all of this, I simply use phpMyAdmin to update a database and set policy (Figure 1). In order to grant a particular access level, I simply update the corresponding cell in the grid, with 1 being on, and 0 being off.

Policy enforcement also will require some client configuration, which I’ll discuss in a moment. However, I’m

also going to discuss using OpenDNS as a means of filtering out things that I’d rather not spend my time testing and filtering. This is a good example of a security-in-depth posture.

I’ve configured OpenDNS to filter out the content that I don’t anticipate ever changing my mind about. I don’t think there’s any reason for my family to be able to access dating sites, gambling sites or porn sites (Figure 2). Although not perfect, the OpenDNS people do a pretty good job of filtering this content without me having to do any testing myself. When that kind of testing fails, it has the potential for some really awkward moments—I’d just assume pass.

Earlier in this article, I mentioned that this would require some client configuration. Most Web browsers allow you to configure them to use a proxy server to access the Internet. The naïve approach is simply to turn

id	name	ip	blocked	whitelist_only	filtered	open	games	flash	java	video	music
1	Julie's Laptop	10.1.1.1	0	0	0	1	1	1	1	1	1
2	Julie's Laptop	192.168.1.20	0	0	0	1	1	1	1	1	1
3	Brandon's PC	192.168.1.30	0	0	1	0	1	1	1	1	1
4	Brandon's Laptop	192.168.1.31	0	0	1	0	1	1	1	1	1
5	Brian's Laptop	192.168.1.32	0	0	1	0	1	1	1	1	1
9	Tyler's Laptop	192.168.1.33	0	0	1	0	1	1	1	1	1

Figure 1. phpMyAdmin Interface for Changing Access Policy

FEATURE Flexible Access Control with Squid Proxy

on proxy access by checking the check box. However, if my kids take their laptops to the library, where our proxy isn't available, they won't be able to

access the Internet, and that violates goal number five. So, I've opted to use the automatic proxy configuration that most modern browsers support.

Custom Choose the categories you want to block.

<input checked="" type="checkbox"/> Academic Fraud	<input checked="" type="checkbox"/> Adult Themes	<input checked="" type="checkbox"/> Adware
<input checked="" type="checkbox"/> Alcohol	<input type="checkbox"/> Anime/Manga/Webcomic	<input checked="" type="checkbox"/> Auctions
<input type="checkbox"/> Automotive	<input type="checkbox"/> Blogs	<input type="checkbox"/> Business Services
<input type="checkbox"/> Chat	<input type="checkbox"/> Classifieds	<input checked="" type="checkbox"/> Dating
<input checked="" type="checkbox"/> Drugs	<input type="checkbox"/> Ecommerce/Shopping	<input type="checkbox"/> Educational Institutions
<input type="checkbox"/> File Storage	<input type="checkbox"/> Financial Institutions	<input type="checkbox"/> Forums/Message boards
<input checked="" type="checkbox"/> Gambling	<input type="checkbox"/> Games	<input checked="" type="checkbox"/> German Youth Protection
<input type="checkbox"/> Government	<input checked="" type="checkbox"/> Hate/Discrimination	<input type="checkbox"/> Health and Fitness
<input type="checkbox"/> Humor	<input type="checkbox"/> Instant Messaging	<input type="checkbox"/> Jobs/Employment
<input checked="" type="checkbox"/> Lingerie/Bikini	<input type="checkbox"/> Movies	<input type="checkbox"/> Music
<input type="checkbox"/> News/Media	<input type="checkbox"/> Non-Profits	<input type="checkbox"/> Nudity
<input type="checkbox"/> P2P/File sharing	<input checked="" type="checkbox"/> Parked Domains	<input type="checkbox"/> Photo Sharing
<input type="checkbox"/> Podcasts	<input type="checkbox"/> Politics	<input checked="" type="checkbox"/> Pornography
<input type="checkbox"/> Portals	<input type="checkbox"/> Proxy/Anonymizer	<input type="checkbox"/> Radio
<input type="checkbox"/> Religious	<input type="checkbox"/> Research/Reference	<input type="checkbox"/> Search Engines
<input checked="" type="checkbox"/> Sexuality	<input type="checkbox"/> Social Networking	<input type="checkbox"/> Software/Technology
<input type="checkbox"/> Sports	<input checked="" type="checkbox"/> Tasteless	<input type="checkbox"/> Television
<input checked="" type="checkbox"/> Tobacco	<input type="checkbox"/> Travel	<input type="checkbox"/> Video Sharing
<input type="checkbox"/> Visual Search Engines	<input type="checkbox"/> Weapons	<input checked="" type="checkbox"/> Web Spam
<input type="checkbox"/> Webmail		

Looking for [security categories](#)?

Figure 2. OpenDNS filters out the easy stuff.

This requires that I write a JavaScript function that determines how Web sites are to be accessed, either directly or via a proxy (Listing 1).

Every time your browser accesses a Web site, it calls the `FindProxyForURL()` function to see what method it should use to access the site: directly or via a proxy. The function shown in Listing 1 is just an example, but it demonstrates a few use cases that are worth mentioning. As you can see from line 15, you can return a semicolon-delimited list of methods to use. Your browser will try them in turn. In this case, if the

proxy happens to be inaccessible, you will fall back to `DIRECT` access to the Web site in question. In a more strict environment, that may not be the correct policy.

On line 11, you can see that I'm ensuring that Web sites on our local network are accessed directly. On line 7, I'm demonstrating how to test for particular hostnames. There are a few Web sites that I access through a VPN tunnel on my workstation, so I cannot use the proxy. Finally, on line 3, you see something interesting. Here, I'm testing to see if a particular hostname is resolvable to an IP address. I've

Listing 1. Automatic Proxy Configuration Script

```
1 function FindProxyForURL(url, host) {
2
3     if (!isResolvable("proxy.example.com")) {
4         return "DIRECT";
5     }
6
7     if (shExpMatch(host, "*.example.com")) {
8         return "DIRECT";
9     }
10
11    if (isInNet(host, "10.0.0.0", "255.0.0.0")) {
12        return "DIRECT";
13    }
14
15    return "PROXY 10.1.1.158:3128; DIRECT";
16 }
```



The fool-proof method to enforce this policy is at the gateway router: simply set a firewall rule that prevents access to the Web coming from any IP address except the proxy.

configured our LAN's DNS server to resolve that name, but no other DNS server would be able to. This way, when our kids take their laptops out of our network, their browser doesn't try to use our proxy. Sure, we simply could fail over to direct access like we did on line 15, but fail over takes time.

The automatic proxy configuration is something that a more sophisticated user could circumvent. There are add-ins for various browsers that would prevent the user from changing this configuration. However, that wouldn't prevent the user from installing a new browser or starting a new Firefox profile. The fool-proof method to enforce this policy is at the gateway router: simply set a firewall rule that prevents access to the Web coming

from any IP address except the proxy. This even could be done for specific client-host combinations, if needed.

While you're adding firewall rules to your gateway router, you might be tempted to configure the router to forward all Web traffic through the proxy, forming what often is called a transparent proxy. However, according to RFC 3143, this isn't a recommended configuration, because it often breaks things like browser cache and history.

So now that I've discussed client, DNS and possible router configuration, it's time to look at the Squid proxy server configuration. The installation itself was pretty straightforward. I just used my distribution's package management system, so I won't discuss that here. The Squid proxy provides a lot of knobs that you can

turn in order to optimize its cache and your Internet connection. Even though performance improvements are a nice ancillary benefit from implementing the proxy server, those configuration options are beyond the scope of this discussion. That leaves the single configuration change that is necessary in order plug my code into the system. All that was needed was to edit the `/etc/squid/squid.conf` file and add a single line:

```
redirect_program /etc/squid/redirector.pl
```

This one directive essentially tells the Squid proxy to “ask” my program how to handle every request that clients make. The program logic is pretty simple:

1. Listen on STDIN for requests.
2. Parse the request.
3. Make a decision based on policy.
4. Return the answer to the proxy.

Let’s look at the sample code in Listing 2.

The main loop begins on line 11, where it reads from STDIN. Lines 11–24 mostly are concerned with parsing the request from the Squid proxy.

Lines 25–28 are where the program queries the database to see what the particular client’s permissions are. Lines 29–57 check to see what permissions were read in from the database and return appropriately. In the case where the client is allowed “filtered” access to the Internet, I have a skeleton of the logic that I have in mind. I didn’t want to bog this article down with trivial code. It was more important to demonstrate the structure and general logic of a Squid proxy redirector than it was to supply complete code. But you can see that I could implement just about any conceivable access policy in just a few lines of code and regular expressions.

The `send_answer()` function starting on line 67 really doesn’t do much at this point, but in the future, I could add some logging capability here pretty easily.

The `is_on_list()` function starting on line 72 is perhaps a bit interesting. This function takes the hostname that the client is trying to access and breaks it up into a list of subdomains. Then it checks if those subdomains are listed in the database, whose name was passed in as a parameter. This way, I simply can put `example.com` in the database, and it will match `example.com`, `www.example.com` or

Listing 2. The Proxy Redirector

```

1  #!/usr/bin/perl
2
3  use DBI;
4
5  $blocked = "http://192.168.1.10/blocked.html";
6
7  my $dbh = DBI->connect("dbi:mysql:authentication:host=
↳192.168.1.10", "user", "password") || die("Can't
↳connect to database.\n");
8
9  $|=1;
10
11 while (<STDIN>) {
12     my($sth, $r, $c);
13     my($url, $client, $d, $method, $proxy_ip, $proxy_port);
14
15     chomp($r = $_);
16
17     if ($r !~ m/\S+/) { next; }
18
19     ($url, $client, $d, $method, $proxy_ip, $proxy_port)
↳= split(/\s/, $r);
20
21     $client =~ s/\/-//;
22     $proxy_ip =~ s/myip//;
23     $proxy_port =~ s/myport//;
24
25     $sth = $dbh->prepare("select * from web_clients
↳where ip='\$client'");
26     $sth->execute();
27     $c = $sth->fetchrow_hashref();
28
29     if ($c->{blocked} eq "1") {
30         send_answer($blocked);
31         next;
32     }
33
34     if ($c->{whitelist_only} eq "1") {
35         if (!is_on_list("dom_whitelist", $url)) {
36             send_answer($blocked);
37             next;
38         }
39     }
40
41     if ($c->{filtered} eq "1") {
42         if ($c->{games} eq "0") {
43             # Check URL to see if it's
↳on our games list
44         }
45
46         if ($c->{flash} eq "0") {
47
48             # Check URL to see if it looks
↳like flash
49         }
50         send_answer($url);
51         next;
52     }
53
54     if ($c->{open} eq "1") {
55         send_answer($url);
56         next;
57     }
58
59     send_answer($url);
60     next;
61 }
62
63 exit 0;
64
65 #####
66
67 sub send_answer {
68     my($a) = @_;
69     print "$a\n";
70 }
71
72 sub is_on_list {
73     my($list, $url) = @_;
74     my($o, @a, $i, @b, $b, $sth, $c);
75
76     $url =~ s/^https*:\//;
77     $url =~ s/^\.+@//;
78     $url =~ s/[:\./].*//;
79
80     @a = reverse(split(/\./, $url));
81
82     foreach $i (0 .. $#a) {
83         push(@b, $a[$i]);
84         $b = join(".", reverse(@b));
85
86         $sth = $dbh->prepare("select count(*) from
↳$list where name='\$b'");
87         $sth->execute();
88         ($c) = $sth->fetchrow_array();
89
90         if ($c > 0) { return $c; }
91     }
92
93     return $c+0;
94 }
95

```

webmail.example.com, for example.

By passing in different table names, I can use the same matching algorithm to match any number of different access control lists.

As you can see, the code really isn't very complex. But, by adding a bit more complexity, I should be able to enforce just about any access policy I can imagine. There is, however, one area that needs to be improved. As written, the program accesses the database several times for *each* access request that it handles. This is extremely inefficient, and by the time you read this, I probably will have implemented some sort of caching mechanism.

However, caching also will make the system less responsive either to changes to access policy or access control lists, as I will have to wait for the cached information to expire or restart the proxy daemon.

In practice, I've seen something that is worth mentioning. Most Web browsers have their own caching mechanism. Because of this cache, if you change an access policy at the proxy, your clients aren't always aware of the change. In the case where you "open up" access, customers will need to refresh their cache in order to access previously blocked content.

In the case where you restrict access, that content still may be available until the cache expires. One solution is to set the local cache size to 0 and simply rely upon the proxy server's cache.

Also, once the clients have been configured to talk to a proxy on the local network, it becomes possible to swap in different proxies or even to daisy-chain proxies without the client needing to do anything. This opens up the possibility of using Dan's Guardian, for example, to do content filtering in addition to access control.

By this time, many of you might think I'm some kind of uber-strict control freak. However, my family spends a lot of time on the Internet—sometimes to a fault. Most of the time, my family members use the Internet in an appropriate manner, but when they don't, my wife and I need a means of enforcing household rules without having to keep a constant watch over our kids. ■

Mike Diehl has used Linux since it came on 5.25" floppy disks. He lives in Blythewood, South Carolina, with his wife and four sons.

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

SECURITY IN THREE Ds: **DETECT, DECIDE and DENY**

How to detect
dubious login attempts
and deny access
to hackers.

FEDERICO KEREKI

Whenever a server is accessible via the Internet, it's a safe bet that hackers will be trying to access it. Just look at the SSH logs for any server you use, and you'll surely find lots of "authentication failure" lines, originating from IPs that have nothing to do with you or your business. Brute-force attempts (such as "dictionary attacks") try different passwords over and over to try to get into your box, and there's always a chance that they eventually will succeed. Thus, it's a good idea to apply these "three Ds" for your security: *detect* intruder attempts, *decide* when they've gone "over the top" (past what would be acceptable for honest-to-goodness typing mistakes), and *deny* them access at least for a (longish!) while.

Several tools manage this kind of monitoring (see the Resources section). In this article, I describe installing, configuring and running DenyHosts. With it, you'll have a

running background *dæmon* that will check your system continuously for access attempts, decide if they look unsafe, block them and inform you. DenyHosts even can be configured to share information with other servers, so whenever a hacker is detected on one system, it will be blocked on other systems too.

Installation and Configuration

DenyHosts' current version is 2.6 (from June 2013). It is a Python script, and you probably already have that language installed. If not, you'll need to use your distribution package tool to set it up first. You need version 2.3 or higher.

Many distributions already provide a "denyhosts" package, and using your system tools is the simplest installation method. For example, type `sudo apt-get install denyhosts` for Ubuntu, `sudo yum install denyhosts` for Red Hat or `sudo zypper install denyhosts` for OpenSUSE. Otherwise, you can download the tar.gz file

DenyHOSTS

(see Resources) and then do:

```
$ tar zxvf DenyHosts-2.6.tar.gz
$ cd DenyHosts-2.6
$ sudo python setup.py install
```

No matter how you install DenyHosts, a `/usr/share/denyhosts` will be created, with all configuration and script files within it. You'll want to

edit the `denyhosts.cfg` file to configure DenyHosts for your environment. (Should that file not exist, do `cp denyhosts.cfg-dist denyhosts.cfg` to get a basic sample configuration file to start with.) Blank lines and lines starting with `#` are ignored. There are plenty of comments to help you understand each configuration item, but pay close attention to some key items (Table 1).

Table 1. Key Configuration Items for DenyHosts

PARAMETERS	EXPLANATIONS
SECURE_LOG	Location of the access log: <code>/var/log/secure</code> for Red Hat, <code>/var/log/auth.log</code> for Ubuntu or OpenSUSE, and so on. DenyHosts will scan this file to detect possible hacking attempts.
HOSTS_DENY	Location of the restricted host file, usually <code>/etc/hosts.deny</code> . DenyHosts will add lines to this file whenever a possible intruder is detected.
LOCK_FILE	A file path and name: this file is created when DenyHosts starts and is deleted when it exits. If you try to run DenyHosts, and this file exists, Denyhosts will exit immediately, preventing more than one instance from running at the same time.
WORK_DIR	The directory DenyHosts will use for its own data.
DAEMON_LOG	Location of the log file that DenyHosts will use to report its status when run in <code>dæmon</code> mode.
DAEMON_SLEEP	Amount of time that DenyHosts will wait before checking the <code>SECURE_LOG</code> .
DAEMON_PURGE	Amount of time between purges of old entries in <code>HOSTS_DENY</code> (see <code>PURGE_DENY</code> below).
BLOCK_SERVICE	What service should be blocked in <code>HOSTS_DENY</code> . Set it to <code>"sshd"</code> to disable only SSH access or to <code>"ALL"</code> to block every attempt from the remote host.
ADMIN_EMAIL	Address to which e-mail messages regarding blocked hosts and suspicious logins should be sent. If you set this to a non-blank value, you'll need to set <code>SMTP_HOST</code> , <code>SMTP_PORT</code> , <code>SMTP_USERNAME</code> and <code>SMTP_PASSWORD</code> as well, so DenyHosts can connect to your mail server.
DENY_THRESHOLD_INVALID, DENY_THRESHOLD_VALID and DENY_THRESHOLD_ROOT	After how many failed login attempts DenyHosts should block a host (due to wrong user account names, correct user names but wrong passwords or failed root access attempts). But, you shouldn't allow remote root access at all; see the Resources section for some suggestions about this!
RESET_ON_SUCCESS	If set to <code>"yes"</code> , the failed count for the server will be reset to zero after a successful login.
AGE_RESET_INVALID, AGE_RESET_VALID and AGE_RESET_ROOT	After what period of time the failed count for the host will be reset to 0 for wrong user account attempts, wrong password attempts and root access attempts.
PURGE_DENY	Time after which <code>HOSTS_DENY</code> entries will be purged if you run DenyHosts with the <code>--purge</code> flag. I usually go with <code>"1w"</code> (one week), but you can leave it blank (never purge), or you can use minutes, hours, days or weeks.
PURGE_THRESHOLD	How many times a host can be purged until DenyHosts decides it's a confirmed risk and stops purging it, thus disabling it forever. Set this to zero to disable the feature.

As an extra assurance that you won't be banned from your own server, you should edit `/etc/hosts.allow` and add a line for each IP you use, in a format like `sshd: 111.222.33.44`. DenyHosts has a file of its own, allowed-hosts, that defines IPs (one per line) that won't ever be blocked.

Now you are ready to start running DenyHosts—let's move on to that!

Running DenyHosts

You can run DenyHosts periodically (let's say as a cron job), but it's better to run it in `dæmon` mode. DenyHosts then will run in the background, checking the access logs to detect possible intruders. For this, you need to create an extra configuration file by executing `cp daemon-control-dist daemon-control`. Then, edit the file to make sure its three parameters are

Table 2. Running DenyHosts in `dæmon` mode requires extra configuration.

PARAMETERS	EXPLANATIONS
DENYHOSTS_BIN	Should point to the <code>denyhosts.py</code> script.
DENYHOSTS_CFG	Should point to the <code>denyhosts.cfg</code> configuration file.
DENYHOSTS_LOCK	Should point to the same file as <code>LOCK_FILE</code> in Table 1.

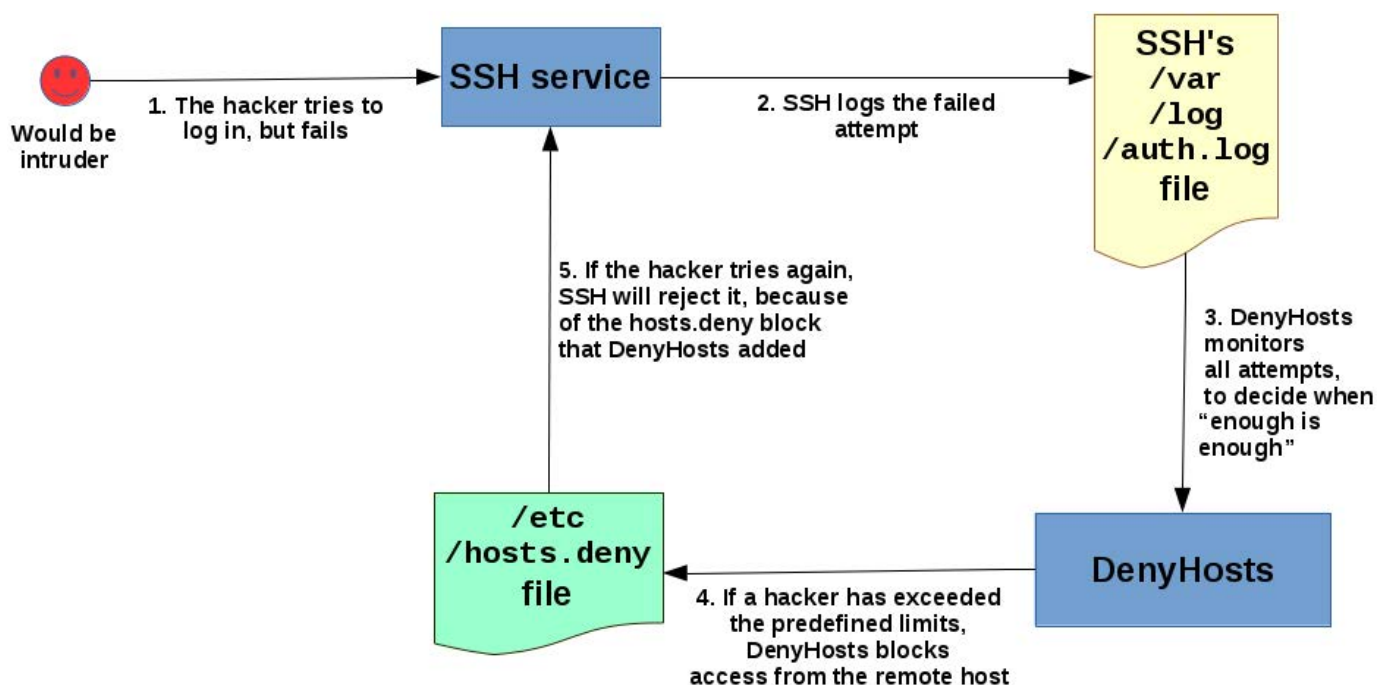


Figure 1. DenyHosts monitors the access logs and eventually locks out the hacker's host.

ONCE DENYHOSTS IS RUNNING, IT WILL MONITOR THE LOGS, AND AFTER DETECTING REPEATED FAILED LOGIN ATTEMPTS, WHEN CERTAIN THRESHOLDS ARE REACHED, IT WILL DECIDE TO BLOCK THE REMOTE IP TO DENY THE WOULD-BE HACKER ALL POSSIBILITIES OF CONNECTION.

defined correctly (Table 2).

Then, all you need to do is start the service with `/etc/init.d/denyhosts restart` or `service denyhosts restart`, depending on your server, and you are set. From that moment on, DenyHosts will monitor login attempts, and upon detection of oft-repeated failures, it will decide to deny all access to the would-be intruder (Figure 1). Just in case, you also should set DenyHosts to start automatically on reboot—doing `sudo chkconfig denyhosts on` is a way to manage this.

Once DenyHosts is running, it will monitor the logs, and after detecting repeated failed login attempts, when certain thresholds are reached, it will decide to block the remote IP to deny the would-be hacker all possibilities of connection. For example, I deliberately entered wrong passwords time after time, and eventually I was blocked (Figure 1 also explains why DenyHosts needs to run as root; otherwise, it

wouldn't be able to edit the hosts.deny file):

```
> ssh fkereki@your.own.server
fkereki@your.own.server's password:
Permission denied, please try again.
fkereki@your.own.server's password:
Permission denied, please try again.
fkereki@your.own.server's password:
Permission denied (publickey,password).
> ssh fkereki@your.own.server
fkereki@your.own.server's password:
Permission denied, please try again.
fkereki@your.own.server's password:
Permission denied, please try again.
fkereki@your.own.server's password:
Permission denied (publickey,password).
.
.
.
several attempts, and then...
> ssh fkereki@your.own.server
ssh_exchange_identification: read: Connection reset by peer
```

DON'T GET LOCKED OUT!

It goes without saying, you should test DenyHosts to verify that it's running. Of course, and most important, keep at least an open shell in your server while you do your testing; otherwise, should anything go wrong, you could get locked out of your own IP address, and you'd have a nice problem!

While you try accessing the server from a different machine, you can use `tail -f` to monitor both the `hosts.deny` file and DenyHosts' own log file (see the `HOSTS_DENY` and `DAEMON_LOG` configuration parameters in Table 1). After a few failed attempts, you should see changes in both files, and you should be getting an e-mail message as well, confirming that your configuration is fine.

Table 3. Sharing blocked IPs with other servers helps everybody.

PARAMETERS	EXPLANATIONS
SYNC_SERVER	To what server you should connect for central synchronization. Currently, <code>xmlrpc.denyhosts.net:9911</code> is the only available server, but in the future, organizations may install their own synchronizing servers for internal usage.
SYNC_INTERVAL	How often to synchronize with the server.
SYNC_UPLOAD and SYNC_DOWNLOAD	Whether DenyHosts should upload and download banned IPs; the logical option is YES in both cases.
SYNC_DOWNLOAD_THRESHOLD	Download only IPs that have been detected and blocked at least by this number of DenyHosts distinct servers.
SYNC_DOWNLOAD_RESILIENCY	Download only IPs that have been detected over this period. The larger this value, the more probable the IP is an actual hacker.

DenyHosts eventually will lift the restriction (depending on the configuration), but if a valid user somehow managed to get banned from a valid server, you can fix the problem without further delay (see the Forgiven a User sidebar).

Sharing Results

DenyHosts provides another service that allows you to share whatever IPs you block with other servers around the world, and vice versa. The idea is simple. If anybody detects that hacking attempts are coming from a certain

FORGIVING USERS

No matter how careful, once in a while someone messes up, enters the wrong password too many times and gets banned by DenyHosts. How can you fix this? Once you've decided that the errant user should be forgiven, take the following steps:

1. Learn the user's current IP so you can enable it again.
2. Stop DenyHosts.
3. Check every file in the

WORK_DIR directory (see Table 1), and delete every line in which the IP appears (using `grep` will help).

4. Edit `/etc/hosts.deny` in the same way.
5. Restart DenyHosts.

Now, if the user tries again, access should be granted. Don't be tempted to skip step 3. If you do, as soon as you restart DenyHosts, the remote IP will be blocked again!

IP, every server that blocks it without waiting to be attacked is made a bit safer. There are a few parameters you need to consider, as shown in Table 3.

Sharing this information is usually a good idea, so you can stave off would-be hackers even before they start sniffing at your server. The sharing service currently is free (although donations are accepted), but eventually businesses might be charged for it,

as the DenyHosts FAQ states. You can contact the DenyHosts creator to license the server component and run your own sharing server.

Conclusion

Security is a 24/7 job, and all the tools that you can add into the mix go a long way toward making your server secure. DenyHosts is easy to install, and it provides quick detection of hacking attempts, blocking remote hackers and

Resources

You can find DenyHosts at <http://denyhosts.sourceforge.net>, and download it from the project’s site at <http://sourceforge.net/projects/denyhosts>. Several common questions are answered at <http://denyhosts.sourceforge.net/faq.html>.

Other tools with similar goals are BlockHosts at <http://www.aczoom.com/tools/blockhosts> and Fail2Ban at <http://www.fail2ban.org>.

I’ve previously written other articles on security measures for *Linux Journal*: “PAM—Securing Linux Boxes Everywhere” (January 2009), “Implement Port-Knocking Security with knockd” (January 2010) and “More Secure SSH Connections” (January 2014).

impeding brute-force attacks. Although it’s not a “silver bullet” solution for all possible intruder problems, it’s a worthy tool that you should add to your security toolkit. Give it a try! ■

Federico Kereki is a Uruguayan systems engineer with more than 25 years of experience developing systems, doing consulting work and teaching at universities. He is currently working as a UI Architect at Globant, using a good mixture of development frameworks, programming tools and operating systems—and FLOSS, whenever possible! A couple years ago, he wrote the *Essential GWT* book, in which you also can find some security concerns for Web applications. You can reach Federico at fkereki@gmail.com.

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

LINUX JOURNAL ARCHIVE DVD



NOW AVAILABLE

www.linuxjournal.com/dvd

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

Hats Off to Mozilla

Appreciating our friend in the browser non-business.

Firefox turned ten years old last November and celebrated the occasion with a new version (33.1) that featured a much-welcomed developer edition. It also featured a “forget” button that lets you backspace through time, blowing away history, cookies and open tabs: one more privacy tool for the shed.

Those were two among many new moves by Mozilla, Firefox’s parent, all siding with individuals leaning against two prevailing winds that have blown across the on-line world for at least a decade.

The first is centralization.

Ten years ago, we still were in what Tantek Çelik calls “the heyday of the independent Web”. Back then, it was easy to homestead on the Net’s frontier with your own domain, site, blog, e-mail and so on. “We all assumed that it was sort of our inevitable destiny that the Web

was open, the Internet was open, everyone had their own identity”, Tantek says. Now most of us live and work in feudal fiefdoms: the Kingdom of Google, the Duchy of Facebook, the Empire of Apple, the Electorate of Amazon, the Principality of Twitter. That we can travel between these castles does not diminish our dependent stature.

On mobile devices, we also live inside the castles of carriers, plus every app’s own walled garden inside those castles. This is very different from the personal computing world, where the Net and the Web are the infrastructural contexts. The Net by nature (its base protocols) has no national boundaries, no tariffs, no “roaming” between countries and carrier networks. The Web by nature is all about links. But apps aren’t about links. They are silos by design. Worse, we don’t acquire them in the open marketplace, but

Now most of us live and work in feudal fiefdoms: the Kingdom of Google, the Duchy of Facebook, the Empire of Apple, the Electorate of Amazon, the Principality of Twitter.

through company stores inside Apple, Google and Microsoft.

The second is surveillance.

We are watched constantly on the commercial Net: in our browsers, though our mobile devices and now by our cars as well. Our overlords rationalize surveillance with five assumptions:

1. People can be better known by machines than by themselves.
2. People are always looking to buy something.
3. The best form of advertising is the most personalized.
4. Secretly following people is good for business, law enforcement, government and other institutional graces of civilization.
5. Nobody's stopping us, so it must be okay.

We now have massive data centers

devoted to crunching data gathered about us and gurgling billions (trillions?) of ads back at us everywhere, whether we like it or not, with utter disregard for collateral damage in the form of ill will and waste levels of 99% and up.

For all the talk about being “conversational” and “personal”, most marketing on-line today is programmatic—in other words, done by robotic algorithms. Fraud is also rampant and rarely discussed, despite being obvious and huge. Meanwhile, on the lack-of-demand side of our non-relationships with advertising machinery, nearly all of us lie and hide to protect our privacy on-line.

The massively clear market message sent by adblocking (144 million people do it, including 41% of 18–29-year-olds) is also dismissed by the advertising industry, which would have us believe that blocking surveillance (aka tracking) would “break the Web”. They forget that both advertising and the Web got along fine without surveillance before the craze started.

Mozilla makes the only popular browser that is open source, uncompromised by commercial parentage and on the side of the individual.

No other name-brand entity, with hundreds of millions of users already, is in a better position than Mozilla to help us fight against all this. Mozilla makes the only popular browser that is open source, uncompromised by commercial parentage and on the side of the individual. Yes, the company does get major funding from Google, but it also has an extreme need to differentiate Firefox from Chrome. Guiding that differentiation are who they work for—you and me—and with.

It's no accident that Mozilla is now partnering with the Tor Project and the Center for Democracy & Technology to (among other things) integrate Tor with the Firefox code base and to host Tor middle relays. Mozilla is also working on tracking protection services to give users more control. Both of these efforts are part of a new effort called the Polaris Privacy Initiative.

There's also Mozilla's Linux-based Firefox OS for phones. It includes the Personal Interest Dashboard and a new initiative being tested (and

not yet public as I write this) called Subscribe the Web. From what I've gathered so far, it offers a new funding mechanism for publishers as an alternative (or a supplement) to advertising models.

The list goes on.

What says the most, at least to me, is a video line in the sand that Mozilla put up on Firefox's 10th anniversary. It's called "Choose Independent". Here's the script:

Who owns the Internet?

The answer is no one.

The answer is everyone.

Which is why thousands of volunteers around the globe give their time and talent.

To create an Internet experience that's owned by everyone.

And doesn't own you.

Where your information isn't being

Resources

Celebrating 10 Years of Firefox:

<https://blog.mozilla.org/blog/2014/11/10/celebrating-10-years-of-firefox-2>

Firefox Developer Edition: <https://www.mozilla.org/en-US/firefox/developer>

Tantek Çelik: <http://tantek.com>

Tantek Çelik, “Why We Need the IndieWeb”:

<https://www.youtube.com/watch?v=HNmKO7Gr4TE&noredirect=1>

“Verizon Wireless Injects Identifiers to Track Mobile Customers’ Online Activities”:

<http://thehackernews.com/2014/10/verizon-wireless-injects-identifiers-to.html>

“Data Monitoring Saves Some People Money on Car Insurance, But Others Will Pay More”:

<http://www.forbes.com/sites/adamtanner/2013/08/14/data-monitoring-saves-some-people-money-on-car-insurance-but-some-will-pay-more>

“Digital Ad Fraud Is Rampant. Here’s Why So Little Has Been Done About It”:

<http://adage.com/article/digital/online-ad-fraud/292285>

“Lying and Hiding in the Name of Privacy”:

<http://customercommons.org/2013/05/08/lying-and-hiding-in-the-name-of-privacy>

“2014 Report: Adblocking Goes Mainstream”:

<http://blog.pagefair.com/2014/adblocking-report>

The Mozilla Manifesto: <https://www.mozilla.org/en-US/about/manifesto>

Mozilla Foundation Financing:

http://en.wikipedia.org/wiki/Mozilla_Foundation#Financing

Tor Partnering with Mozilla: <https://blog.torproject.org/blog/partnering-mozilla>

The Tor Project: <https://blog.torproject.org/blog>

Center for Democracy & Technology: <https://cdt.org>

“Introducing Polaris Privacy Initiative to Accelerate User-focused Privacy Online”:

<http://blog.mozilla.org/privacy/2014/11/10/introducing-polaris-privacy-initiative-to-accelerate-user-focused-privacy-online>

“Tracking Protection in Firefox”:

<http://monica-at-mozilla.blogspot.com.au/2014/11/tracking-protection-in-firefox.html>

Firefox OS for Phones: http://en.wikipedia.org/wiki/Firefox_OS

Firefox Interest Dashboard: <https://addons.mozilla.org/en-US/firefox/addon/firefox-interest-dashboard/?src=cb-dl-recentlyadded>

“Firefox: Choose Independent” Video:

<https://www.youtube.com/watch?v=LtOGa5M8AuU>

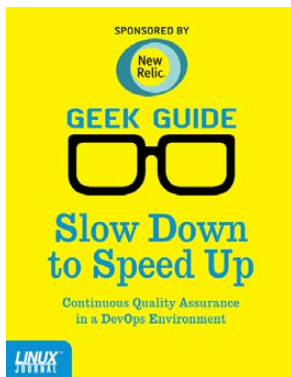
NEW!

Linux Journal eBook Series

GEEK GUIDES

FREE
Download
NOW!

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



By Bill Childers

DevOps is one of the newest and largest movements in Information Technology in the past few years. The name DevOps is a portmanteau of “Development” and “Operations” and is meant to denote a fusion of these two functions in a company. Whether or not your business actually does combine the two functions, the lessons and tools learned from the DevOps movement and attitude can be applied throughout the entire Information Technology space. This eBook focuses on one of the key attributes of the DevOps movement: Quality Assurance. At any point, you should be able to release your product, code or configuration—so long as you continue keeping your deliverables in a deployable state. This is done by “slowing down” to include a Quality Assurance step at each point in your workflow. The sooner you catch an error or trouble condition and fix it, the faster you can get back on track. This will lower the amount of rework required and keep your team’s momentum going in a forward direction, enabling your group to move on to new projects and challenges.

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



By Mike Diehl

This eBook presents a compelling argument as to why you should consider re-architecting your enterprise toward a private cloud. It outlines some of the design considerations that you need to be aware of before implementing your own private cloud, and it describes using the DevCloud installer in order to install OpenStack on an Ubuntu 14 server. Finally, this eBook will familiarize you with the features and day-to-day operations of an OpenStack-based private cloud architecture, all for less than \$10K!

DOWNLOAD NOW AT: <http://linuxjournal.com/geekguides>