

Getting Started
with Nextcloud

Review: the
Librem 13v2

A Look at GDPR's
Massive Impact

LINUX JOURNAL

Since 1994: The

ux community

PRIVACY

HOW TO
PROTECT YOUR DATA

EFFECTIVE
PRIVACY PLUGINS

GIVE YOUR SERVERS
SOME PRIVACY WITH
TOR HIDDEN SERVICES

INTERVIEW:
PRIVATE INTERNET ACCESS
GOES OPEN SOURCE

ISSUE 286 | MAY 2018

www.linuxjournal.com

82 *DEEP DIVE:* **PRIVACY**

83 Data Privacy: Why It Matters and How to Protect Yourself

by Petros Koutoupis

When it comes to privacy on the internet, the safest approach is to cut your Ethernet cable or power down your device. In reality though, most people can't actually do that and remain productive. This article provides an overview of the situation, steps you can take to mitigate risks and finishes with a tutorial on setting up a virtual private network.

106 Privacy Plugins

by Kyle Rankin

Protect yourself from privacy-defeating ad trackers and malicious JavaScript with these privacy-protecting plugins.

113 Tor Hidden Services

by Kyle Rankin

Why should clients get all the privacy? Give your servers some privacy too!



118 Facebook Compartmentalization

by Kyle Rankin

I don't always use Facebook, but when I do, it's over a compartmentalized browser over Tor.

121 The Fight for Control: Andrew Lee on Open-Sourcing PIA

by Doc Searls

When I learned that our sister company, Private Internet Access (PIA) was opening its source code, I immediately wanted to know the backstory, especially since privacy is the theme of this month's issue. So I contacted Andrew Lee, who founded PIA, and an interview ensued.

6 From the Editor—Doc Searls

Privacy Is Still Personal

10 Letters

UPFRONT

15 Product Review: GitStorage

by Petros Koutoupis

19 Readers' Choice Awards**23 FOSS Project Spotlight: Sawmill, the Data Processing Project**

by Daniel Berman

29 FOSS Project Spotlight: CloudMapper, an AWS Visualization Tool

by Scott Piper

34 Caption This: May Winner**35 Visualizing Molecules with EasyChem**

by Joey Bernard

40 Is It Linux or GNU/Linux?

by Christine Hall

44 News Briefs

COLUMNS

46 Reuven M. Lerner's At the Forge

Examining Data Using Pandas

56 Shawn Powers' The Open-Source Classroom

Review: the Librem 13v2

67 Zack Brown's diff -u

What's New in Kernel Development

74 Dave Taylor's Work the Shell

Generating Good Passwords

170 Glyn Moody's Open Sauce

The GDPR Takes Open Source to the Next Level

ARTICLES

128 **Programming in Color with ncurses**

by *Jim Hall*

Jim demonstrates color manipulation with curses by adding colors to his terminal adventure game.

142 **FOSS as a Part of a Corporate Sustainability Plan**

by *VM (aka Vicky) Brasseur*

Free and open-source software is a critical part of your company's supply chain. Here's why and how you can include it in your corporate sustainability plan.

151 **Nextcloud 13: How to Get Started and Why You Should**

by *Marco Fioretti*

Nextcloud could be the first step toward replacing proprietary services like Dropbox and Skype.

AT YOUR SERVICE

SUBSCRIPTIONS: *Linux Journal* is available as a digital magazine, in PDF, EPUB and MOBI formats. Renewing your subscription, changing your email address for issue delivery, paying your invoice, viewing your account details or other subscription inquiries can be done instantly online:

<http://www.linuxjournal.com/subs>. Email us at subs@linuxjournal.com or reach us via postal mail at *Linux Journal*, 9597 Jones Rd #331, Houston, TX 77065 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 different 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*, 9597 Jones Rd #331, Houston, TX 77065 USA. Letters may be edited for space and clarity.

SPONSORSHIP: We take digital privacy and digital responsibility seriously.

We've wiped off all old advertising from *Linux Journal* and are starting with a clean slate. Ads we feature will no longer be of the spying kind you find on most sites, generally called "adtech". The one form of advertising we have brought back is sponsorship. That's where advertisers support *Linux Journal* because they like what we do and want to reach our readers in general. At their best, ads in a publication and on a site like *Linux Journal* provide useful information as well as financial support. There is symbiosis there. For further information, email: sponsorship@linuxjournal.com or call +1-281-944-5188.

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 online: <http://www.linuxjournal.com/author>.

NEWSLETTERS: Receive late-breaking news, technical tips and tricks, an inside look at upcoming issues and links to in-depth stories featured on <http://www.linuxjournal.com>. Subscribe for free today: <http://www.linuxjournal.com/enewsletters>.

LINUX JOURNAL

EDITOR IN CHIEF: Doc Searls, doc@linuxjournal.com

EXECUTIVE EDITOR: Jill Franklin, jill@linuxjournal.com

TECH EDITOR: Kyle Rankin, lj@greenfly.net

ASSOCIATE EDITOR: Shawn Powers, shawn@linuxjournal.com

CONTRIBUTING EDITOR: Petros Koutoupis, petros@linux.com

CONTRIBUTING EDITOR: Zack Brown, zacharyb@gmail.com

SENIOR COLUMNIST: Reuven Lerner, reuven@lerner.co.il

SENIOR COLUMNIST: Dave Taylor, taylor@linuxjournal.com

PUBLISHER: Carlie Fairchild, publisher@linuxjournal.com

ASSOCIATE PUBLISHER: Mark Irgang, mark@linuxjournal.com

DIRECTOR OF DIGITAL EXPERIENCE:

Katherine Druckman, webmistress@linuxjournal.com

GRAPHIC DESIGNER: Garrick Antikajian, garrick@linuxjournal.com

ACCOUNTANT: Candy Beauchamp, acct@linuxjournal.com

COMMUNITY ADVISORY BOARD

John Abreau, Boston Linux & UNIX Group; John Alexander, Shropshire Linux User Group; Robert Belnap, Classic Hackers UGA Users Group; Aaron Chantrell, Bellingham Linux Users Group; Lawrence D’Oliveiro, Waikato Linux Users Group; Chris Ebenezer, Silicon Corridor Linux User Group; David Egts, Akron Linux Users Group; Michael Fox, Peterborough Linux User Group; Braddock Gaskill, San Gabriel Valley Linux Users’ Group; Roy Lindauer, Reno Linux Users Group; Scott Murphy, Ottawa Canada Linux Users Group; Andrew Pam, Linux Users of Victoria; Bob Proulx, Northern Colorado Linux User’s Group; Ian Sacklow, Capital District Linux Users Group; Ron Singh, Kitchener-Waterloo Linux User Group; Jeff Smith, Kitchener-Waterloo Linux User Group; Matt Smith, North Bay Linux Users’ Group; James Snyder, Kent Linux User Group; Paul Tansom, Portsmouth and South East Hampshire Linux User Group; Gary Turner, Dayton Linux Users Group; Sam Williams, Rock River Linux Users Group; Stephen Worley, Linux Users’ Group at North Carolina State University; Lukas Yoder, Linux Users Group at Georgia Tech

Linux Journal is published by, and is a registered trade name of,
Linux Journal, LLC. 4643 S. Ulster St. Ste 1120 Denver, CO 80237

SUBSCRIPTIONS

E-MAIL: subs@linuxjournal.com

URL: www.linuxjournal.com/subscribe

Mail: 9597 Jones Rd, #331, Houston, TX 77065

SPONSORSHIPS

E-MAIL: sponsorship@linuxjournal.com

Contact: Publisher Carlie Fairchild

Phone: +1-281-944-5188

LINUX is a registered trademark of Linus Torvalds.



privateinternetaccess[®]
always use protection[®]

Private Internet Access is a proud sponsor of *Linux Journal*.



**Join a
community
with a deep
appreciation
for open-source
philosophies,
digital
freedoms
and privacy.**

**Subscribe to
Linux Journal
Digital Edition
for only \$2.88 an issue.**

**SUBSCRIBE
TODAY!**

Privacy Is Still Personal

We solved privacy in the natural world with clothing, shelter, manners and laws. So far in the digital world, we have invisibility cloaks and the GDPR. The fastest way to get the rest of what we need is to recognize that privacy isn't a grace of platforms or governments.

By Doc Searls

In the physical world, privacy isn't controversial. In the digital world, it is.

The difference is that we've had thousands of years to work out privacy in the physical world, and about 20 in the digital one. So it should help to cut ourselves a little slack while we come up with the tech, plus the manners and laws to go with it—in that order. (Even though the gun has been jumped in some cases.)

To calibrate a starting perspective, it might help to start with what Yuval Noah Harari says in his book *Sapiens: A Brief History of Humankind*:

Judicial systems are rooted in common legal myths. Two lawyers who have never met can nevertheless combine efforts to defend a complete stranger because they both believe in



Doc Searls is a veteran journalist, author and part-time academic who spent more than two decades elsewhere on the *Linux Journal* masthead before becoming Editor in Chief when the magazine was reborn in January 2018. His two books are *The Cluetrain Manifesto*, which he co-wrote for Basic Books in 2000 and updated in 2010, and *The Intention Economy: When Customers Take Charge*, which he wrote for Harvard Business Review Press in 2012. On the academic front, Doc runs ProjectVRM, hosted at Harvard's Berkman Klein Center for Internet and Society, where he served as a fellow from 2006–2010. He was also a visiting scholar at NYU's graduate school of journalism from 2012–2014, and he has been a fellow at UC Santa Barbara's Center for Information Technology and Society since 2006, studying the internet as a form of infrastructure.

FROM THE EDITOR

the existence of laws, justice, human rights—and the money paid out in fees. Yet none of these things exists outside the stories that people invent and tell one another. There are no gods in the universe, no nations, no money, no human rights, no laws, and no justice outside the common imagination of human beings.

And yet this common imagination is what gives us civilization. We are civil to one another because of all the imaginings we share. And technologies are what make many of those imaginings possible. Those come first. Without the technologies making privacy possible, we would have none of the common manners and civic laws respecting it.

First among those technologies is clothing.

Nature didn't give us clothing. We had to make it from animal skins and woven fabrics. One purpose, of course, was to protect us from cold and things that might hurt us. But another was to conceal what today we politely call our "privates", plus other body parts we'd rather not show.

Second among those technologies was shelter. With shelter we built and marked personal spaces, and valved access and exposure to those spaces with doors, windows and shades.

How we use clothing and shelter to afford ourselves privacy differs between cultures and settings, but is well understood by everyone within both.

With clothing and shelter, we also can signal to others what personal spaces it is okay and not okay to visit, and under what conditions. The ability to send, receive and respect those signals, and to agree about what they mean, are essential for creating order within a civilization, and laws as well.

As of today, we have very little clothing and shelter in the digital world.

Yes, we do have ways of remaining hidden or anonymous (for example, with crypto and Tor), and selectively revealing facts about ourselves (for example with PKI: public key infrastructure). And services have grown up around those developments, such as VPNs.

FROM THE EDITOR

(Disclosure: *Linux Journal*'s sister company is Private Internet Access, a VPN. See my interview in this issue with Andrew Lee, founder of PIA, about the company's decision to open source its code.) We also have prophylaxis against tracking online, thanks to browser extensions and add-ons, including ad blockers that also stop tracking.

As clothing goes, this is something like having invisibility cloaks and bug spray before we get shirts, pants and underwear. But hey, they work, and they're a start.

We need more, but what? Look for answers elsewhere in this issue. In the meantime, however, don't assume that privacy is a grace of companies' (especially platforms') privacy policies. Here are three things worth knowing about those:

1. They can be changed whenever the company pleases.
2. They are not an agreement between you and the company.
3. They are theirs, not yours.

Alas, nearly all conversation about privacy in governments and enterprises assumes that your privacy is mostly their concern.

Here's how I framed an approach to solving privacy three years ago here, in a column titled "["Privacy Is Personal"](#):

So the real privacy challenge is a simple one. We need clothing with zippers and buttons, walls with doors and locks, windows with shutters and shades—that work the same for each and all of us, to give us agency and scale.

Giants aren't going to do it for us. Nor are governments. Both can be responsive and supportive, but they can't be in charge, or that will only make us worse victims than we are already. Privacy for each of us is a personal problem online, and it has to be solved at the personal level. The only corporate or "social" clothing and shelter online are the equivalents of prison garb and barracks.

FROM THE EDITOR

What would our clothing and shelter be, specifically? A few come to mind:

- Ways to encrypt and selectively share personal data easily with other parties we have reason to trust.
- Ways to know the purposes to which shared data is used.
- Ways to assert terms and policies and obtain agreement with them.
- Ways to assert and maintain sovereign identities for ourselves and manage our many personal identifiers—and to operate anonymously by default with those who don't yet know us. (Yes, administrative identifiers are requirements of civilization, but they are not who we really are, and we all know that.)
- Ways to know and protect ourselves from unwelcome intrusion in our personal spaces.

All these things need to be as casual and easily understood as clothing and shelter are in the physical world today. They can't work only for wizards. Privacy is for muggles too. Without agency and scale for muggles, the Net will remain the Land of Giants, who regard us all as serfs by default.

Now that we have support from the GDPR and other privacy laws popping up around the world, we can start working our way down that punch list. ■

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

LETTERS

Re: Getting Started with ncurses

I really enjoyed Jim Hall’s “Getting Started with ncurses” article in the March 2018 issue. When publishing articles that include code and compile commands (which I like to try out myself), please include the code or links to the code required for a successful build. The code for getrandom_int.c and getrandom_int.h were left out. I could guess at the code from the description, but including the code would make for a much more complete article with fully working code to try.

—Ken Lee

Jim Hall replies: I’m glad you liked the article. Sorry I didn’t include the getrandom_int.c source code. The `getrandom_int()` function is a simple wrapper to the Linux `getrandom()` system call, guaranteed to return a positive integer value:

```
#include <sys/random.h>

int getrandom_int(void)
{
    int rand;

    getrandom(&rand, sizeof(int), GRND_RANDOM);

    if (rand < 0) {
        return (rand * -1);
    } else {
        return (rand);
    }
}
```

And the getrandom_int.h header file simply declares the `getrandom_int()` function:

```
int getrandom_int(void);
```

Bravo! Standing-O

Regarding Doc Searls' inaugural From the Editor (in the March 2018 issue), about all I can add for now is: Bravo! and a standing ovation.

The rebirth of *LJ* this year, after (just like Mr Twain) “the report of my death was an exaggeration”, has been excellent news in a world often full of disappointments, and this first, newly reborn issue is a so welcome proof of this resurrection.

In January, upon hearing the good news, I immediately renewed my subscription and continue to look for ways to lend my active support to *LJ*'s mission, and I urge other readers to do the same. In a world full of lying politicians, self-interested C-suite execs, adtech and other evil business-models, *LJ* speaks truth to our common freedoms, rights and interests. We need it—and need to support it—now more than ever.

Thanks so much! Congratulations to Doc, Carlie and the entire staff/crew of this great magazine! Looking forward to future issues, and a long run for *LJv2*!

—Lorin

Doc Searls replies: Thanks, Lorin. We appreciate your support enormously.

In addition to those you name, I'd also like to thank Jill Franklin for her tireless work putting the magazine together for all these many years—and for her infinite patience with habitually late writers such as myself. Katherine Druckman too, for her work at keeping our website up and running for so long on a shoestring that was about one molecule thick.

We're working hard to improve and expand Linux Journal and welcome all the support we can get.

March 2018 From the Editor

First I want to say congratulations on the new launch of *Linux Journal*. I've been a

subscriber from day one.

I found Doc's March 2018 From the Editor column informative and interesting. I'd like to make a suggestion that might help your exposure of the subject and your effort at an open-source project to reverse the on-line advertising model.

Leo Laporte and the TWiT network of shows have three shows that I think should receive copies of this editorial: This Week In Law (TWIL), This Week in Google (TWIG) and This Week in Tech (TWiT).

Just my two cents.

I look forward to many more years of *Linux Journal* and your new effort in fixing online advertising,

—William Main

Resurrection

Congratulations on being reborn. I just received my second “new” issue, and as before, I read Doc Searls first thing, only now I don't have to swim all the way to back. After reading his column, I read through all the letters, and it occurred to me that without the restrictions of paper, you can publish as much as you want to. I found that the conversations within the Letters section were as compelling as anything in the magazine. That's not a knock on the rest of the magazine; it's just an observation. With that said, welcome back!

—Ron Smith

April 2018 Issue

Good to see that *LJ* has returned, and that the April 2018 issue is as good as ever. I've been reading it since 1994 when I helped to start **ManLUG**, which is still going in the present day. Since that time I've written for *Linux Magazine*, *Linux User and Developer* and *Linux Format*. It's been fun.

Here's hoping we see many more issues of *LJ* in the future.

—Richard Ibootson

Ansible

Many thanks to Shawn Powers for his articles about Ansible. I seem to recall a series not too long ago on the same thing, but these hit at just the right time for me. As an Ansible newbie, I have run across an irritation that seems to contradict something in Shawn's January 5 article on the website (see "[Ansible: the Automation Framework That Thinks Like a Sysadmin](#)"). The most attractive way to escalate privileges, as Shawn writes (option #4), is to allow passwordless sudo on specific programs. After figuring out the best way to do that, I discovered that Ansible doesn't handle it well. It's explained [here](#). Disappointing, but it makes sense. Unfortunately, that leaves me with entering passwords manually or letting my account on target machines become functionally equivalent to root. That would take away the security advantage of a separate account, and makes it much easier for me to make horrible mistakes. Are there better (more secure and yet still convenient) ways to handle this?

By the way, I'm so thankful that *LJ* is back. I even extended my subscription (that 99-issue deal from long ago) in support. Looking forward to what you have in store!

—Jesse Jacobsen

From Social Media

Wayne McDermott: I just re-subscribed after a gap of some years. The new format is great, and I like the idea of the Deep Dives. There seems to be less of the old-man sysadmin stuff as well, plus no politics! Good work.

U @urisharf: "...adtech does damage to a brand every time it places that brand's ad next to fake news or on a crappy publisher's website." Kudos to @linuxjournal, whitelisted in my adblocker.

Roberto Carraro @robcar1972: Oh, I love the new *LJ*!

LETTERS

Shawn Powers @shawnp0wers: Can we PLEASE refer to this as the zombie issue? At least internally? :D

Josh Wheeler @mantlepro: Replying to @BryanLunduke @linuxjournal @Microsoft Microsoft's involvement in open source probably has more to do with strategic advantage vs. freedom of the end user from unjust software tyranny. Their involvement in GNU+Linux is counter-intuitive at best. We need more copyleft advocates who care about preserving users' freedoms

Stormy Roy @royking3: Replying to @dsearls @linuxjournal and 3 others The current policy environment has definitely incentivized the growth of adtech. While the ideas of #CustomerTech or VRM have been around for decades, why do you think it's yet to go mainstream? It's not a "technology" problem.

Steve Ketelsen @podfish: Just cracked open my first issue of the newly launched @linuxjournal. Skimming though the PDF, it really feels like the old flavor is back, and it's jam packed: 181 pages of great content! Welcome back, friends. ■

SEND LJ A LETTER *We'd love to hear your feedback on the magazine and specific articles. Please write us [here](#) or send email to ljeditor@linuxjournal.com.*

PHOTOS *Send your Linux-related photos to ljeditor@linuxjournal.com, and we'll publish the best ones here.*

UPFRONT



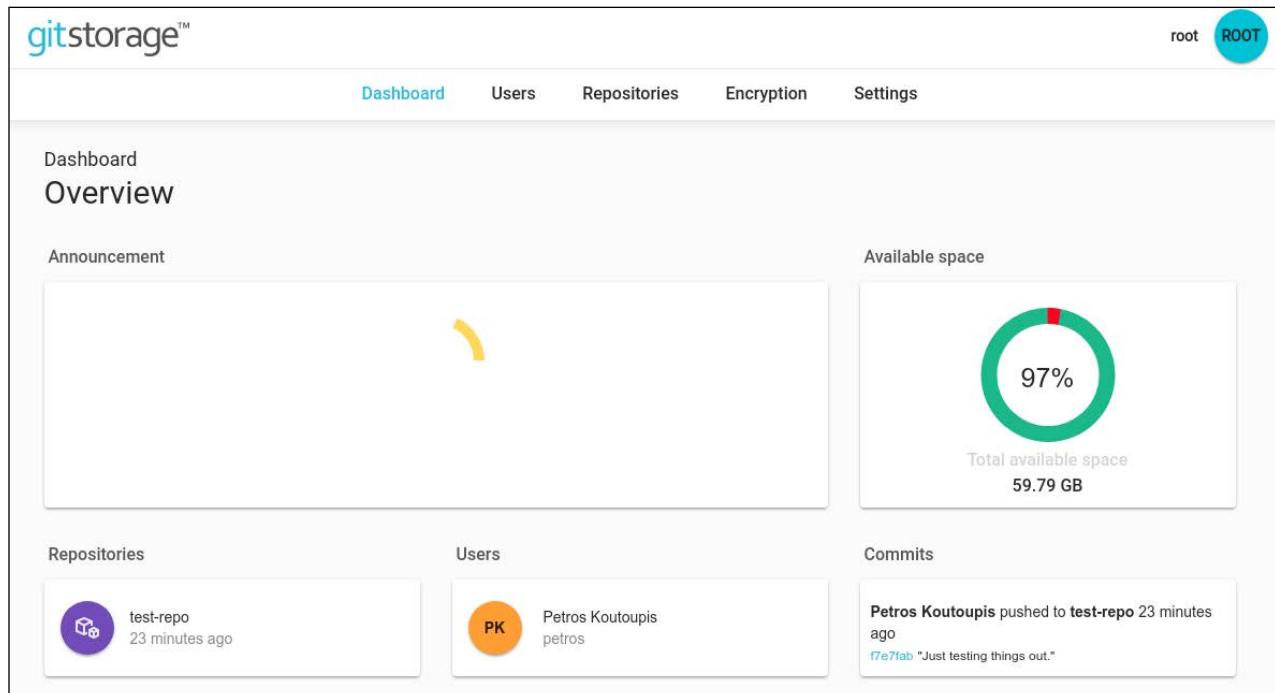
Product Review: GitStorage

Petros reviews the GitStorage server appliance, which emphasizes data privacy and security.

By profession, I'm a software developer. Aside from a preferred editor, what matters most to a developer is the use of a Source Code Manager (SCM). So, when a new product comes along featuring my favorite SCM, Git, I had no choice but to spend some time using it.

GitStorage develops and distributes a Git server appliance of the same name with an emphasis on data privacy and security. The company produces two flavors, the key differences being the following:

- Price: \$399 vs. \$499.
- Local storage capacity: 16GB vs. 64GB.
- Number of users: 10 vs. unlimited.
- Color: pink vs. blue.

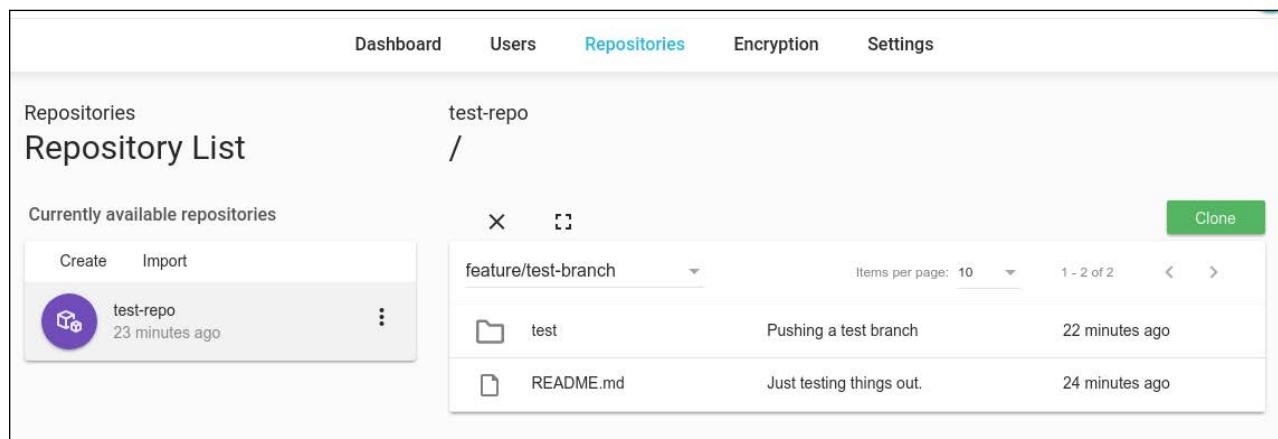


The GitStorage Dashboard

When I received the product, it was packaged well with everything needed to hit the ground running: the device, a micro-USB-connected power plug, mounting screws and a basic set of instructions for connecting to the local device over Ethernet. All you need to do is connect it to your router or switch, apply power to the device, and within a minute, you'll be able to access a web-based front end to the device's assigned IP address from within your preferred web browser.

On initial connection, you're greeted with a setup and configuration menu. This setup allows you to configure the administrator account, generate CA certificates for your web browser, configure SMTP email relays so you even can connect the device to an existing Dropbox account for remote synchronization—all of which can be configured at a later date.

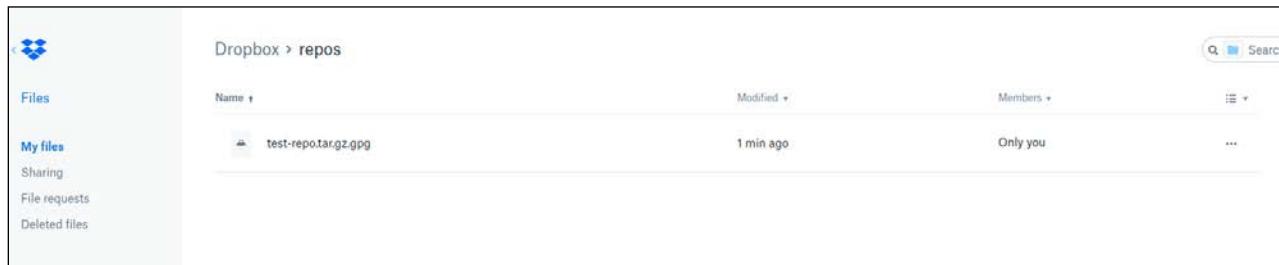
When the configuration is complete, you're redirected to the dashboard of a very intuitive user interface. And because this is a single-purpose appliance, the features and functionality presented in the interface are simple and easy to understand. You can dynamically create or remove users, create new or remove existing code repositories, and assign the appropriate users to those repositories. When those same users log in to the web interface, they will be limited only to the repositories to which they are assigned.



The screenshot shows the UPFRONT web interface with the 'Repositories' tab selected in the top navigation bar. The main content area is titled 'Repository List' and displays a table of available repositories. The table has two columns: 'Repositories' and 'test-repo'. The first row shows 'test-repo' with a slash indicating it's a root repository. Below this, there's a section titled 'Currently available repositories' with 'Create' and 'Import' buttons. A card for 'test-repo' shows it was created 23 minutes ago. To the right of the table is a detailed view of the 'test-repo' branch, showing a list of commits. The commits are:

Commit	Message	Time
feature/test-branch	Pushing a test branch	22 minutes ago
test	Just testing things out.	24 minutes ago
README.md		

A View of Repositories



A Synchronized Archive File to Dropbox

Again, the device is predominately focused on securing your data and limiting access to those who are granted it. Data at rest is always encrypted—that is, it's local to the device, even when synchronized to the cloud. For instance, if Dropbox is configured as your back-end remote archive location, a tarball that is GPG-signed will be synchronized to it. Even if people were to access that Dropbox archive, they will need a password to decrypt the file and then untar the tarball.

Now, how does GitStorage compare to existing Git solutions? It simplifies the task of deploying and maintaining your very own Git servers. You don't need to be a Git expert to get this handy device up and running and start providing developers the comfort and security of maintaining their code using the industry's most popular SCM. Out of the box, it is easy to configure and just works. Also, if very sensitive source code is pushed to this device, rest assured that your data at rest always will be encrypted in both local and remote locations.

What environments would benefit from this? I see this as a perfect solution for hobbyists and small-to-medium-sized software development firms. As long as you don't push too many binary files, the larger of the two devices (64GB) definitely can last you some time. And even if it doesn't, you're also able to add more GitStorage devices to your network.

—Petros Koutoupis

Readers' Choice Awards

This month the categories are Best Content Management System, Best Desktop Environment and Best Programming Language. Note that all the contenders listed were nominated by readers via Twitter. Be sure to check LinuxJournal.com for new polls each week and vote for your favorites!



BEST CONTENT MANAGEMENT SYSTEM

- **WordPress: 42%**

- Drupal: 23%
- Other: 15%
- Joomla: 10%
- Concrete5: 4%
- Grav: 3%
- ModX: 3%

Unless you've been living under a rock, you most certainly have heard of WordPress, one of the most popular blogging platforms around that also happens to be 100% open source. WordPress [powers 27% of the web](#) from personal to corporate to even government sites ([Whitehouse.gov](#) for one).

In a 2008 interview, *Linux Journal*'s Katherine Druckman asked WordPress founder Matt Mullenweg, “You frequently have reiterated your commitment to open-source ideals and GPL licensing. How has this commitment factored into the development of your company, Automattic? How do you use open-source technology to achieve your goals?”

Mullenweg responded:

When I set out to create Automattic, it was an interesting dilemma—in our society, it seemed the best way to have an impact on the world was working within a for-profit framework, but at the same time, I'd seen multiple examples of “open-source companies” suffocating the communities they grew from.

I came across an interesting hack though—by keeping WordPress.org a separate entity from Automattic and basing our business entirely on GPL code, you create a balance that aligns the fiduciary responsibilities of the corporation with the interests of the community at large. In the long term—10, 20 years from now—it still will be in the best interest of Automattic to support the broader community as much as possible, because its own business succeeds when they do.

I didn't want WordPress to be a one-company project, so by separating out the nonprofit and for-profit sides and making some explicit decisions about businesses Automattic would never enter, we created a lot of room for other companies to embrace, support and build on top of WordPress. Hopefully, we also set a good example of how to contribute back to the community.

It was the best way I could think of to ensure that the principles I believe in would endure beyond my personal involvement or control of either organization. (But I still look both ways when crossing the street.)

Congratulations to WordPress for being *Linux Journal's* Readers' Choice.

BEST DESKTOP ENVIRONMENT

- **KDE: 35%**
- GNOME: 20%
- Xfce: 15%
- Cinnamon: 11%
- MATE: 7%
- Other: 7%
- Unity: 3%
- LXQt: 1%

Thanks to its stability, performance, feature set and a loyal following, the K Desktop Environment (KDE) won Best Desktop Environment in this year's *Linux Journal* Readers' Choice Awards.

Linux Journal reader Larry Coombes says his vote was for KDE for a mass of reasons, including:

- 1) Totally configurable, which is ESSENTIAL for new users who want their Linux desktop configured to emulate the exact rendering of Windows (XP or 7) that they have worked on forever. Also, essential for when the new user turns to me and says, "Can Linux do

XYZ?? Can I make it so...”, with KDE, the answer is always “Yes”, which is much better than explaining limitations of a desktop environment and trying to train someone, because then you hit “But I could always do this in Windows...”.

2) I have tried other desktop environments, and none of them come close to KDE’s keyboard-shortcut support, which is ESSENTIAL to my business. I teach young children, and I am jumping from app to app, file to file in chaos. With KDE, I can minimize touchpad operation and use a laptop as a proper, portable device. I have around 50 keyboard shortcuts to move my work forward—and to be able to use in taxis and trains!

BEST PROGRAMMING LANGUAGE

- **Python: 31%**

- C: 20%
- C++: 14%
- Other: 9%
- Java: 8%
- Perl: 7%
- JavaScript: 4%
- PHP: 3%
- Ruby: 3%

Python wins Best Programming Language again this year in *Linux Journal*'s annual Readers' Choice Awards. It's easy to use, powerful and versatile with a really large and active community. Having that supportive community ensures that developers of all skill levels easily can find the support and documentation they require, which feeds Python's popularity. It certainly helps that Python has something like a corporate sponsor. Python is recognized as an official language at Google, running on many of its internal systems and showing up in many Google APIs. In fact, Google's developer website offers free [Python classes, videos and exercises](#).

FOSS Project Spotlight: Sawmill, the Data Processing Project

Introducing Sawmill, an open-source Java library for enriching, transforming and filtering JSON documents.

If you're into centralized logging, you are probably familiar with the ELK Stack: Elasticsearch, Logstash and Kibana. Just in case you're not, ELK (or Elastic Stack, as it's being renamed these days) is a package of three open-source components, each responsible for a different task or stage in a data pipeline.

Logstash is responsible for aggregating the data from your different data sources and processing it before sending it off for indexing and storage in Elasticsearch. This is a key role. How you process your log data directly impacts your analysis work. If your logs are not structured correctly and you have not configured Logstash correctly, your logs will not be parsed in a way that enables you to query and visualize them in Kibana.

[Logz.io](#) used to rely heavily on Logstash for ingesting data from our customers, running multiple Logstash instances at any given time. However, we began to experience some pain points that ultimately led us down the path to the project that is the subject of this article: Sawmill.

Explaining the Motivation

Over time, and as our data pipelines became more complex and heavy, we began to encounter serious performance issues. Our Logstash configuration files became extremely complicated, which resulted in extremely long startup times. Processing also was taking too long, especially in the case of long log messages and in cases where there



was a mismatch between the configuration and the actual log message.

The above points resulted in serious stability issues, with Logstash coming to a halt or sometimes crashing. The worst thing about it was that troubleshooting was a huge challenge. We lacked visibility and felt a growing need for a way to monitor key performance metrics.

There were additional issues we encountered, such as dynamic configuration reload and the ability to apply business logic, but suffice it to say, Logstash was simply not cutting it for us.

Introducing Sawmill

Before diving into Sawmill, it's important to point out that Logstash has developed since the time we began working on this project, with new features that help deal with some of the pain points described above.

So, what is Sawmill?

Sawmill is an open-source Java library for enriching, transforming and filtering JSON documents.

For Logstash users, the best way to understand Sawmill is as a replacement of the filter section in the Logstash configuration file. Unlike Logstash, Sawmill

does not have any inputs or outputs to read and write data. It is responsible only for data transformation.

Using Sawmill pipelines, you can use your groks, geoip, user-agent resolving, add or remove fields/tags and more, in a descriptive manner, using configuration files or builders, in a simple DSL, allowing you to change transformations dynamically.

Sawmill Key Features

Here's a list of the key features and processing capabilities that Sawmill supports:

- Written in Java, Sawmill is thread-safe and efficient, and uses caches where needed.
- Sawmill can be configured in HOCON or JSON.
- Sawmill allows you to configure a timeout for long processing using a configurable threshold.
- Sawmill generates metrics for successful, failed, expired and dropped executions, and a metric for processing exceeding a defined threshold. All metrics are available per pipeline and processor.
- 25+ processors, including grok, geoip, user-agent, date, drop, key-value, json, math and more.
- Nine logical conditions, including the basics as well as field-exists, has-value, match-regex and math-compare.

Using Sawmill

Here is a basic example illustrating how to use Sawmill:

```
Doc doc = new Doc(myLog);
PipelineExecutor pipelineExecutor = new PipelineExecutor();
pipelineExecutor.execute(pipeline, doc);
```

As you can see, there are a few entities in Sawmill:

- *Doc* — essentially a Map representing a JSON.
- *Processor* — a single *document* logical transformation. Either grok-processor, key-value-processor, add-field and so on.
- *Pipeline* — specifies a series of processing steps using an ordered list of *processors*. Each *processor* transforms the *document* in some specific way. For example, a *pipeline* might have one *processor* that removes a field from the *document*, followed by another *processor* that renames a field.
- *PipelineExecutor* — executes the *processors* defined in the *pipeline* on a *document*. The *PipelineExecutor* is responsible for the execution flow—handling onFailure and onSuccess flows, stops on failure, exposes metrics of the execution and more.
- *PipelineExecutionTimeWatchdog* — responsible for warning on long processing time, interrupts and stops processing on timeout (not shown in the example above).

Sawmill Configuration

A Sawmill pipeline can get built from a **HOCON string** (Human-Optimized Config Object Notation).

Here is a simple configuration snippet, to get the feeling of it:

```
{  
  "steps": [{  
    "grok": {  
      "config": {  
        "field": "message",  
        "overwrite": ["message"]  
      }  
    }  
  },  
  {"patterns": ["%{COMBINEDAPACHELOG}+ %{GREEDYDATA:extra_fields}"]}  
}
```

```
    }
  }]
}
```

Which is equivalent to the following in HOCON:

```
steps: [{  
  grok.config: {  
    field : "message"  
    overwrite : ["message"]  
    patterns :  
      ["%{COMBINEDAPACHELOG}+ %{GREEDYDATA:extra_fields}"]  
  }  
}]
```

To understand how to use Sawmill, here's a simple example showing GeoIP resolution:

```
package io.logz.sawmill;  
  
import io.logz.sawmill.Doc;  
import io.logz.sawmill.ExecutionResult;  
import io.logz.sawmill.Pipeline;  
import io.logz.sawmill.PipelineExecutor;  
  
import static io.logz.sawmill.utils.DocUtils.createDoc;  
  
public class SawmillTesting {  
  
  public static void main(String[] args) {  
  
    Pipeline pipeline = new Pipeline.Factory().create(  
      "{ steps :[{\n" +  
      "    geoIp: {\n" +
```

```
"      config: {\n" +
"        sourceField: \"ip\"\n" +
"        targetField: \"geoip\"\n" +
"        tagsOnSuccess: [\"geo-ip\"]\n" +
"      }\n" +
"}]\n" +
"}");  
  
Doc doc = createDoc("message", "testing geoip resolving",
↳ "ip", "172.217.11.174");
ExecutionResult executionResult = new
PipelineExecutor().execute(pipeline, doc);
  
  
if (executionResult.isSucceeded()) {
    System.out.println("Success! result
↳ is:"+doc.toString());
}
}  
}
```

End Results

We've been using Sawmill successfully in our ingest pipelines for more than a year now, processing the huge amounts of log data shipped to us by our users.

We know Sawmill is still missing some key features, and we are looking forward to getting contributions from the community. We also realize that at the end of the day, Sawmill was developed for our specific needs and might not be relevant for your use case. Still, we'd love to get your feedback.

—*Daniel Berman*

FOSS Project Spotlight: CloudMapper, an AWS Visualization Tool

Duo Security has released CloudMapper, an open-source tool for visualizing Amazon Web Services (AWS) cloud environments.

When working with AWS, it's common to have a number of separate accounts run by different teams for different projects. Gaining an understanding of how those accounts are configured is best accomplished by visually displaying the resources of the account and how those resources can communicate. This complements a traditional asset inventory.

Duo built CloudMapper to generate interactive network diagrams of AWS accounts and released it as open source on [Github](#).

See a demo [here](#).

Using CloudMapper, you can quickly answer a number of questions, such as:

- Which resources are publicly exposed?
- What resources can communicate internally with which other resources?
- Do you have a robust architecture in the event of an availability zone failure?
- How many regions is this account using? How “big” is this account? How complex is it?

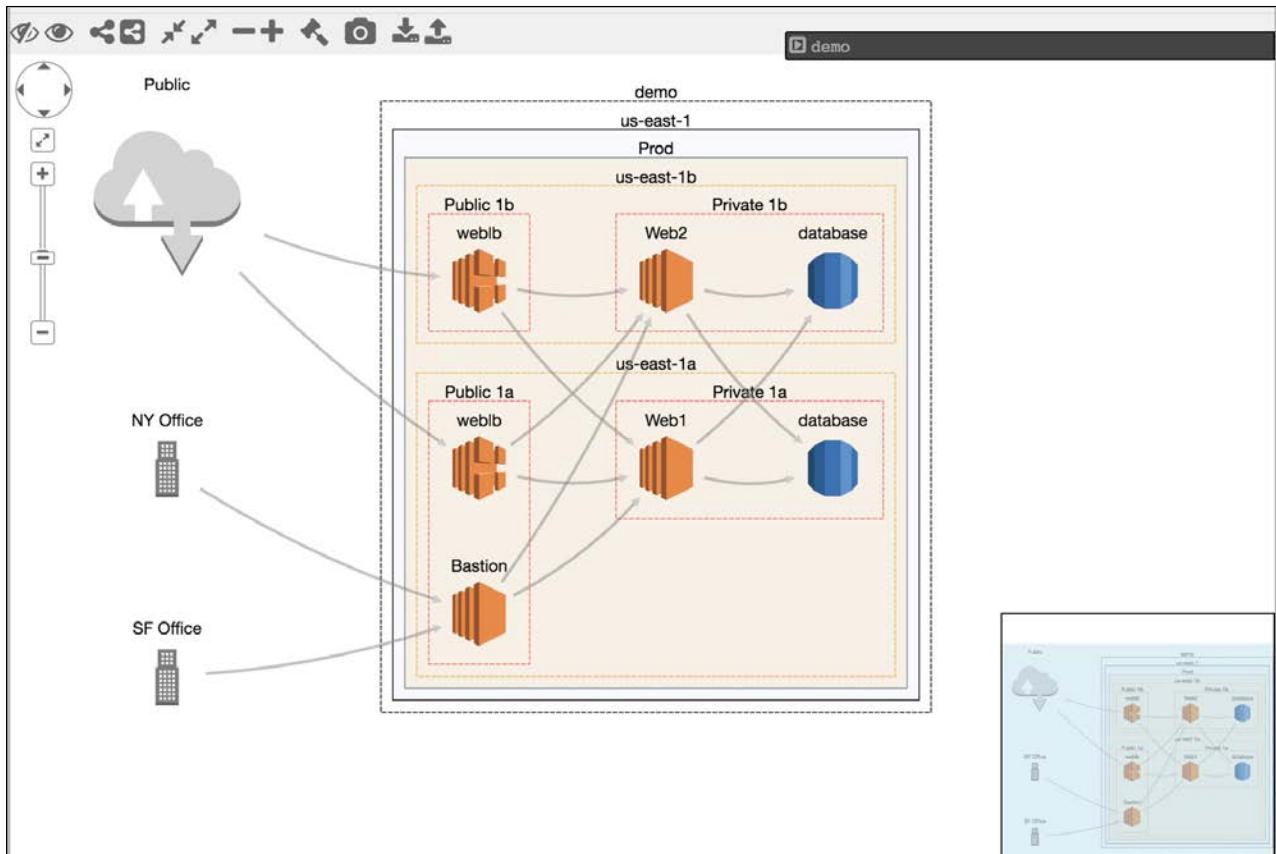


Figure 1. Screenshot of CloudMapper Visualizing a Demo Account

CloudMapper allows engineers to double-check their understanding of what they've built, quickly understand other environments and present that information to other stakeholders.

How It Works

There are three steps to getting up and running with CloudMapper:

1. Collect information about an AWS account via a shell script that uses the [AWS CLI](#).
2. Convert that data into a format usable by the web browser.
3. Run a simple web server to view the collected data in your browser.

The first step of collecting information only requires the privileges to describe and list information about an account. This can be done with the AWS **SecurityAudit** policy. If you don't have direct access to the account, someone who does can run this script and send you the bundle of files it creates.

The second step of converting these cached files into something for the web browser

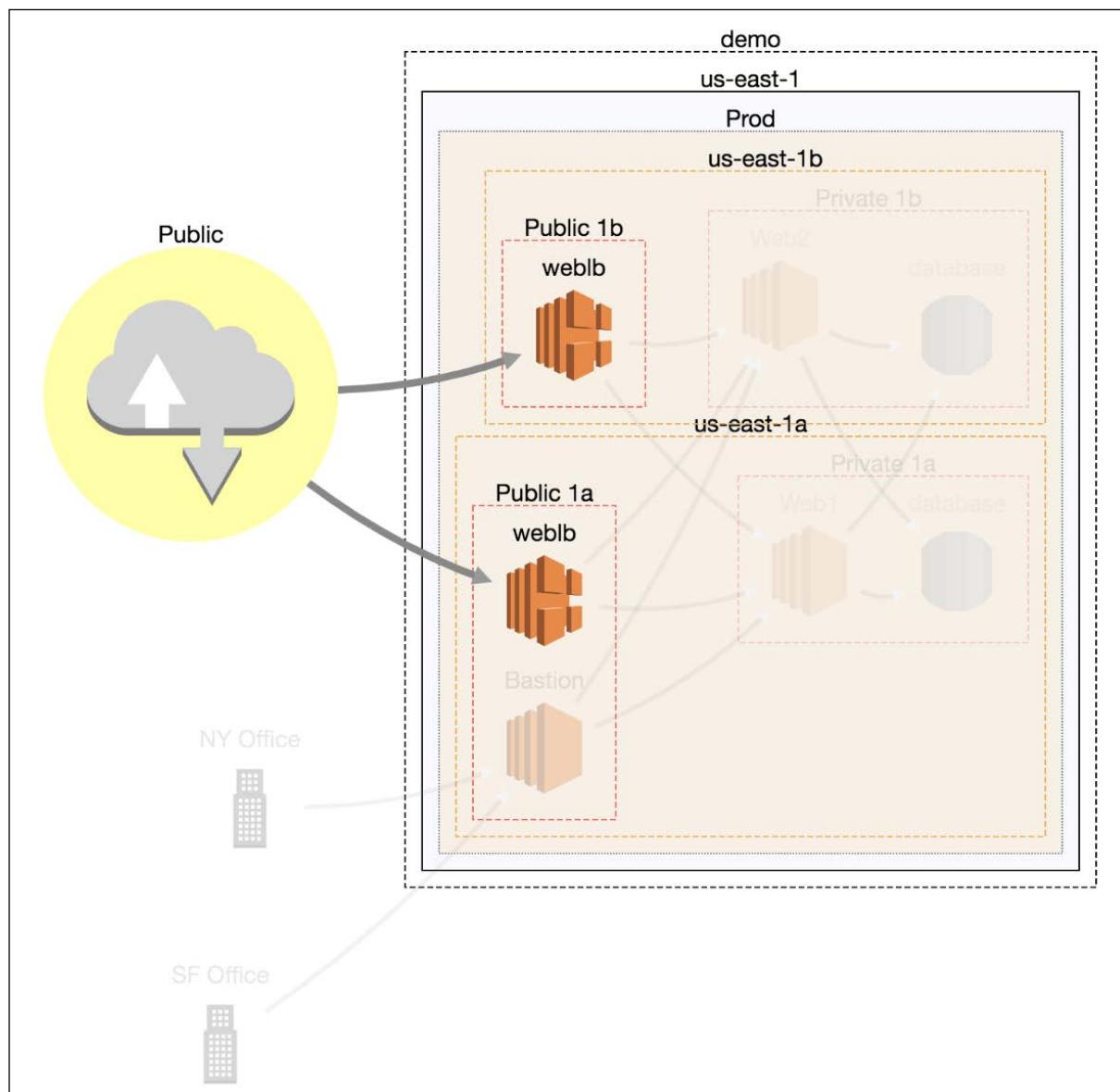


Figure 2. Highlighted Neighbors

display is where most of the logic is. This is where the Security Groups are analyzed to determine what network pathways exist, and parent/child relationships are created between nodes, such as EC2 instances, and compound node structures, such as subnets, availability zones, VPCs, regions and accounts.

The final step of visualizing the data in the browser makes heavy use of [cytoscape.js](#) to perform the graph layout and allow interaction with the resources. Historically, this problem would have been solved with [graphviz](#), but that solution is more suited toward generating static images as output. Cytoscape originally was created to visualize molecular interaction networks, but it has been found to be well suited for a

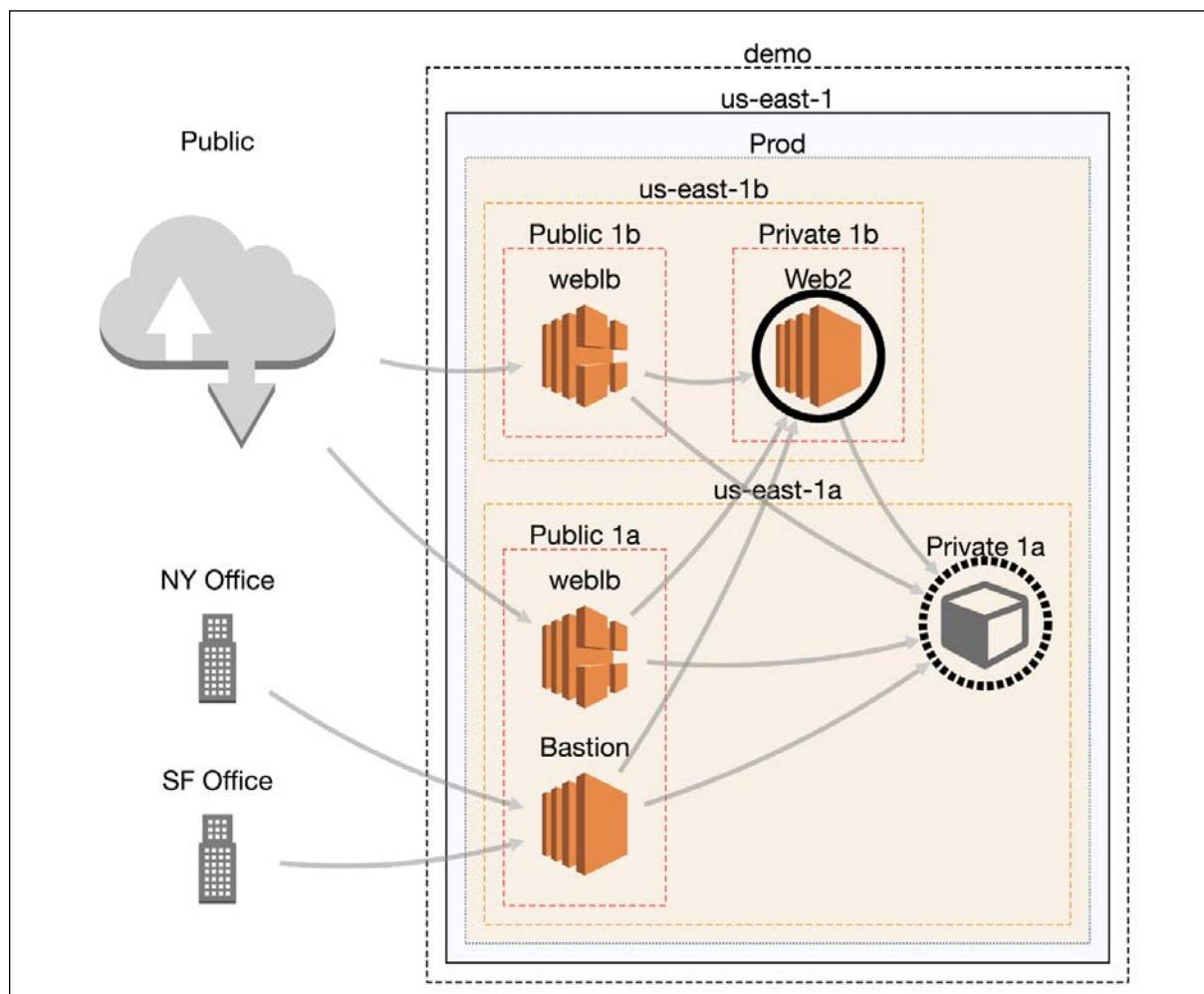


Figure 3. Compressed Node

variety of other network visualizations.

By using cytoscape.js and presenting the diagram to the user in a web browser, you can click on nodes to get more information about them, move them around, delete them and more. This is only for visualizing the data, so any actions you take will not impact your actual AWS environments.

You can zoom in and out, pan, save a high-definition image, or save and load the layout after you've moved nodes around. You can click on a compound node, such as a VPC, and compress it to a single node in order to simplify the visualization. You easily can find and select neighbors, siblings, children or parent nodes. You can click on edges to get details about the Security Groups that are allowing that communication to happen.

Improving the Layout

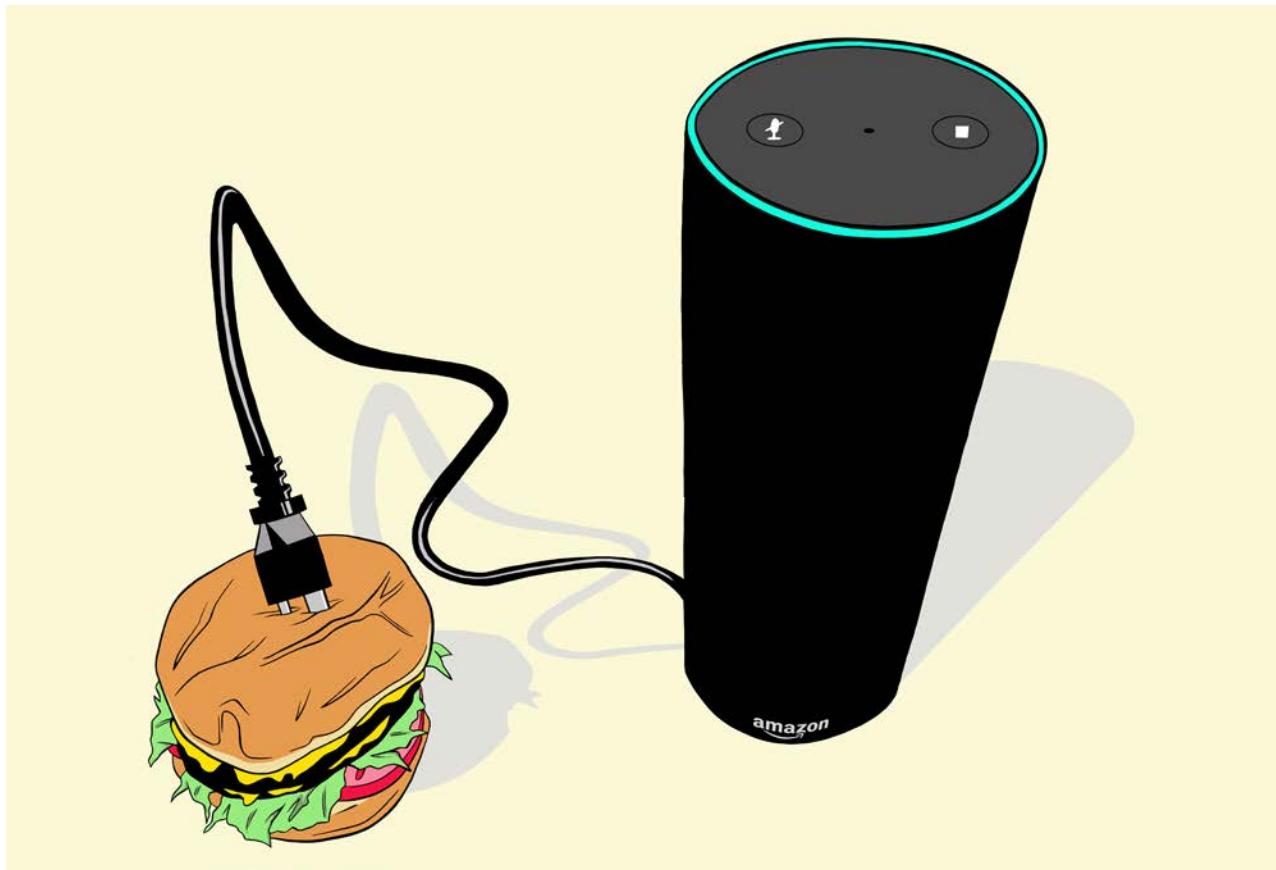
Visualizing large networks is a hard problem. CloudMapper uses the CoSE (Compound Spring Embedder) layout for Cytoscape.js that was developed by the i-Vis Lab in Bilkent University, which is regarded as one of the best algorithms for laying out graphs with compound nodes. However, any graph with a large number of nodes and edges is inherently complicated. To manage this problem, CloudMapper has a number of filtering options that can be used when preparing the data for visualization.

Options to reduce the amount of data displayed include:

- Show only specified regions.
- Ignore internal edges if you want to see only what resources are exposed publicly.
- Aggregate similar EC2 instances to a single node based on a tag name.

Even with these techniques and the advanced layout algorithm used, still be prepared to spend some time rearranging the nodes.

—Scott Piper, AWS Security Consultant



Caption This: May Winner

Winner: *Is this what my cardiologist means by I need an echo?*

—Tom Dison, twitter.com/fretinator

Second Place: *USBurger* —Greg Charnock, twitter.com/gregcharnock7

Third Place: “*Alexa, where’s the beef?*” —Jack, via comment on
<https://www.linuxjournal.com>

Each month, we provide a cartoon in need of a caption—check <https://www.linuxjournal.com> for the next one. You submit your caption in the comments on the site or via Twitter, we choose three finalists, and readers vote for their favorite. See the June issue for the next winner.

Visualizing Molecules with EasyChem

Chemistry is one of the heavy hitters in computational science. This has been true since the beginning, and it's no less true today. Because of this, several software packages specifically target this user group. Most of these software packages focus on calculating things within chemistry, like bond energies or protein folding structures. But, once you've done the science portion, you need to be able to communicate your results, usually in the form of papers published in journals. And, part of the information you'll need to disseminate is imagery of the molecules from your work. And, that's where EasyChem, this article's subject, comes into play.

EasyChem helps generate publication-quality images of molecular structures. It should be available in the package management repositories for most distributions. In Debian-based distributions, you can install it with the following command:

```
sudo apt-get installed easychem
```

Once it's installed, you can start it either from your GUI's menu system or from the command prompt. When it first starts, you get a blank canvas within which to start your project.

One of the first things you'll want to check is whether the option to have helpful messages is turned on. You can check this by clicking Options→Learning messages. With this selected, you'll get helpful information in the bottom bar of the EasyChem window.

Let's start with a simple molecule like benzene. Benzene is a ring of six carbon atoms, with every other bond a double bond. You can create this structure by using the options at the bottom of the draw window. Making sure that the "Add bonds" option is selected, select the "Simple" bond from the drop-down of "Bond type". If you now place the

UPFRONT

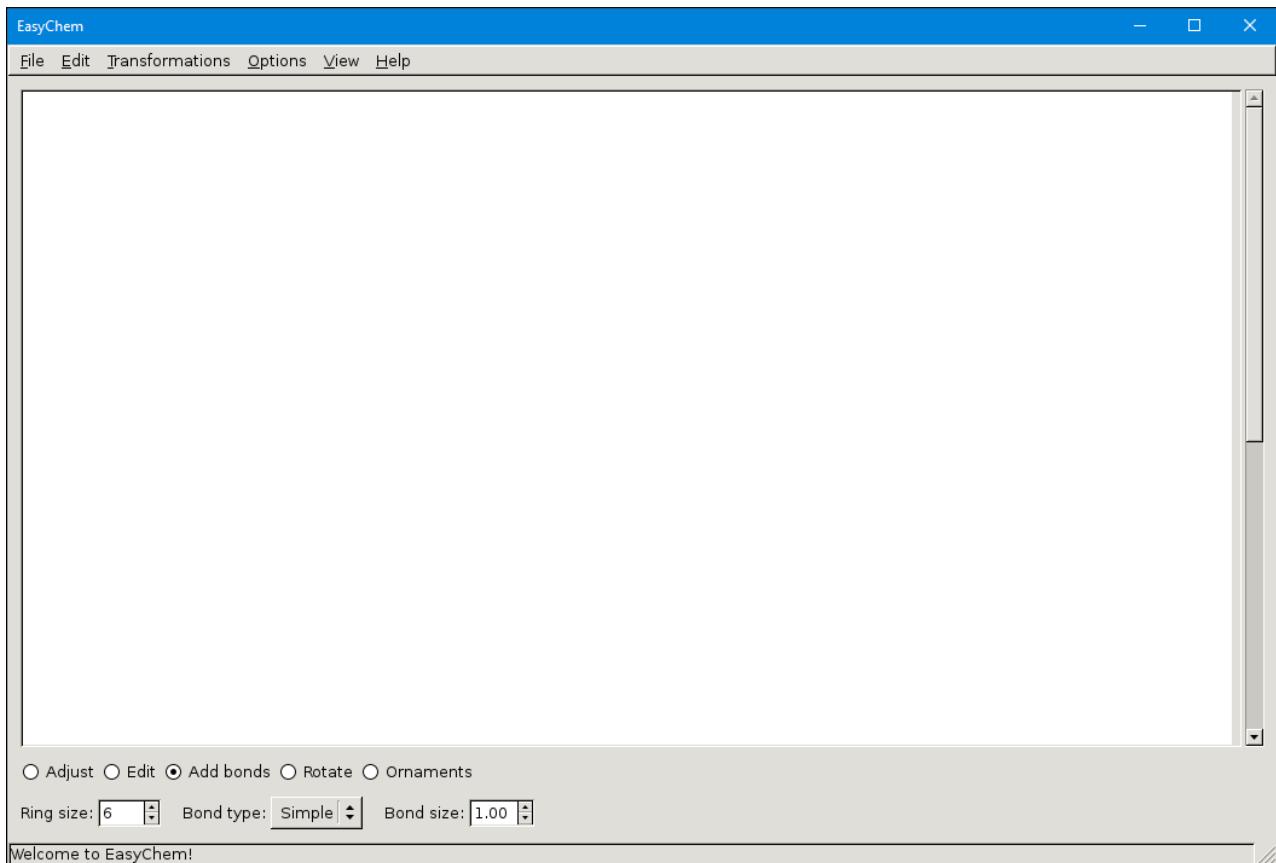


Figure 1. You get a blank workspace when you first start EasyChem.

mouse pointer somewhere in the window and click and drag, you'll get a single bond drawn. To get a ring, you need to hold down the Ctrl key, and then click and drag. This will draw a ring structure for you.

You can set the number of atoms to use in the ring with the “Ring size” option in the bottom left of the window. The default is six, which is what you'll want for your benzene ring.

To get the alternating bond types, select the “Edit” option at the bottom, and then you'll be able to select individual bonds and change their types. When you select one of the bonds, you'll see a new pop-up window where you can change the details, such as the type of bond, along with the color and the relative width if it is a multiple bond.

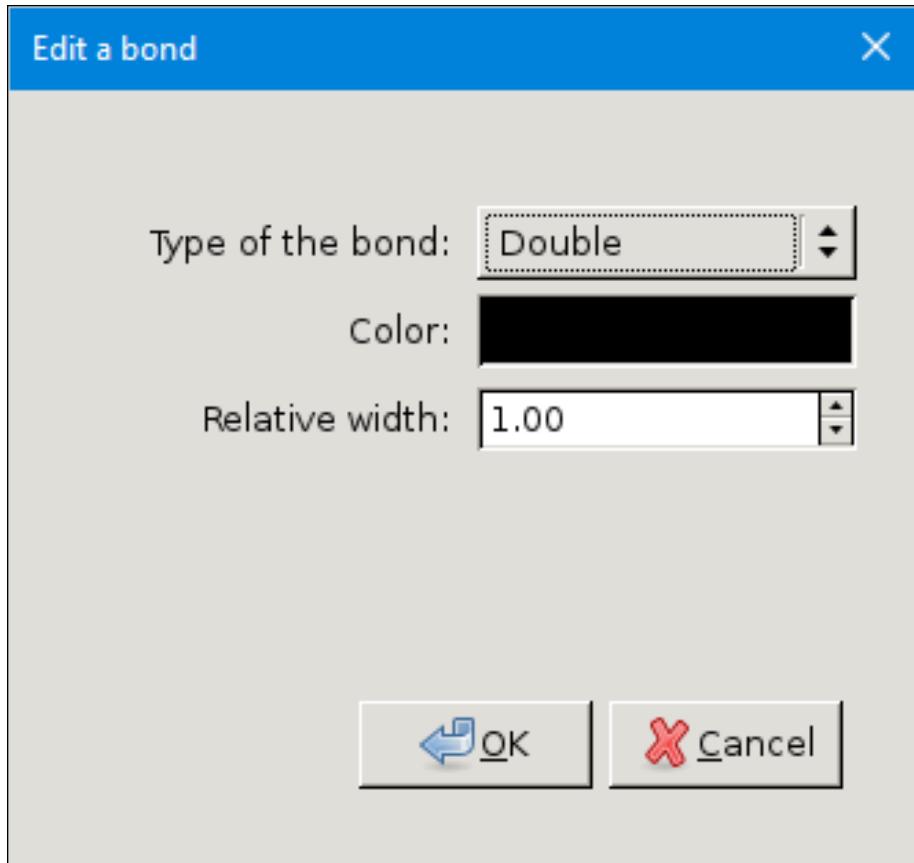


Figure 2. EasyChem lets you change several options of individual bonds.

Once you have some of your diagram constructed, you'll likely want to make alterations to the overall look. When you select the first option at the bottom of the window, labeled "Adjust", you can select one or more objects and adjust their position in the diagram. If you have a collection of objects you need to move, you can select them by creating a box around them with your mouse. They'll then change color to show they've been selected and you can move them around. If you select a single bond instead, the rest of the molecule will be adjusted when you move that one single bond.

You can rotate selections of your diagram with the Rotate option. In the benzene ring example, you can't rotate individual bonds. It only makes sense to rotate the complete ring.

You also can add ornamentation to the ending of the bonds. These are items like non-bonding pairs, electronic gaps or radicals. If you are creating larger additions to a

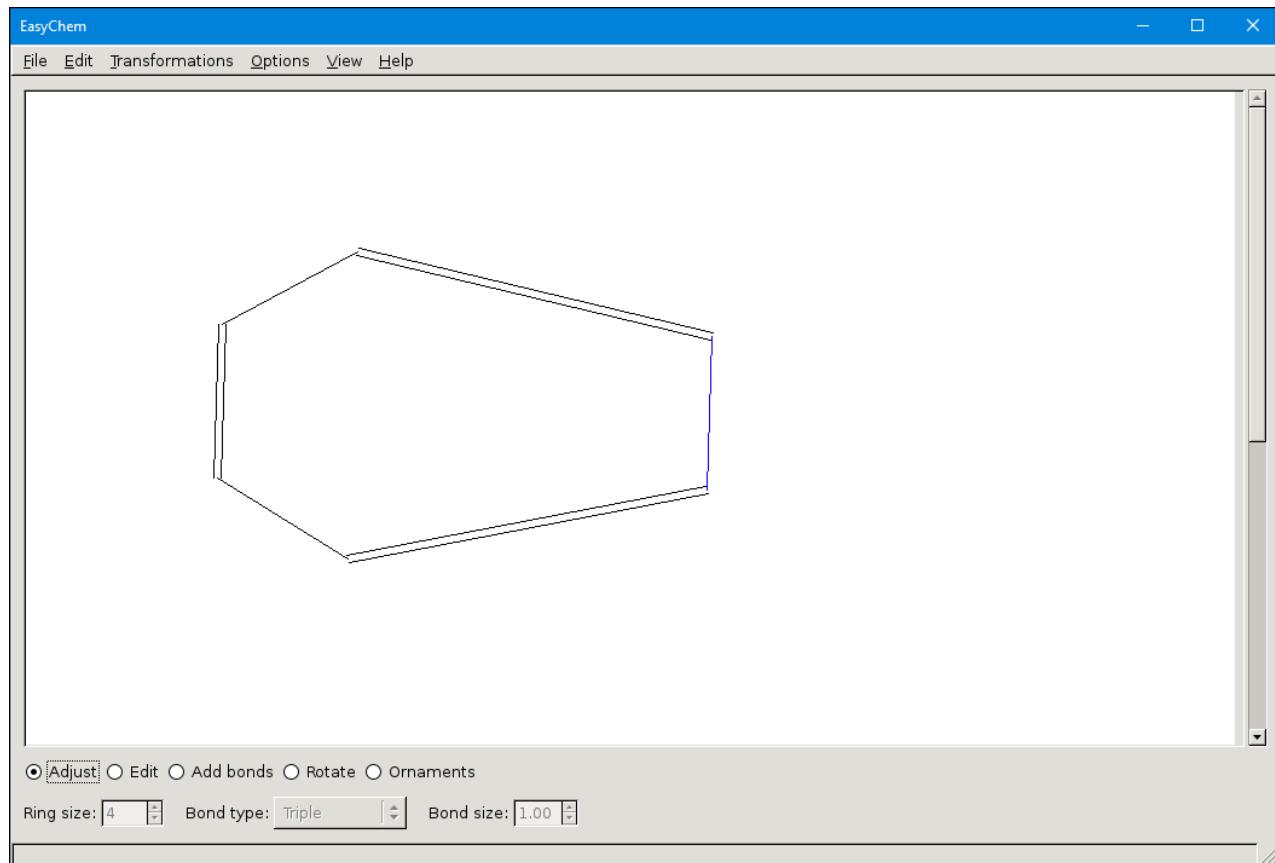


Figure 3. You can move elements of your diagram around, either individually or as a group.

molecule, you'll probably want to build it up from simpler pieces. For example, if you want to add an OH group to the benzene ring, you would add a single bond to one of the vertices. Next, at the end of this new bond, you would add an "Atom/Group" bond type, and then you can add a label to it.

The bond type list provides other decorative options. For example, you could add a circle to the center of the benzene ring, or you could add arrows or arcs to indicate transitions.

You can save your images using the native file format. When you click File→Save, a dialog window pops up asking you to save your work to a file with the ".ech" ending. This way you can save your intermediate steps and be able to re-create any output files. When you finish your molecule, you'll want to export it so you can use it in one

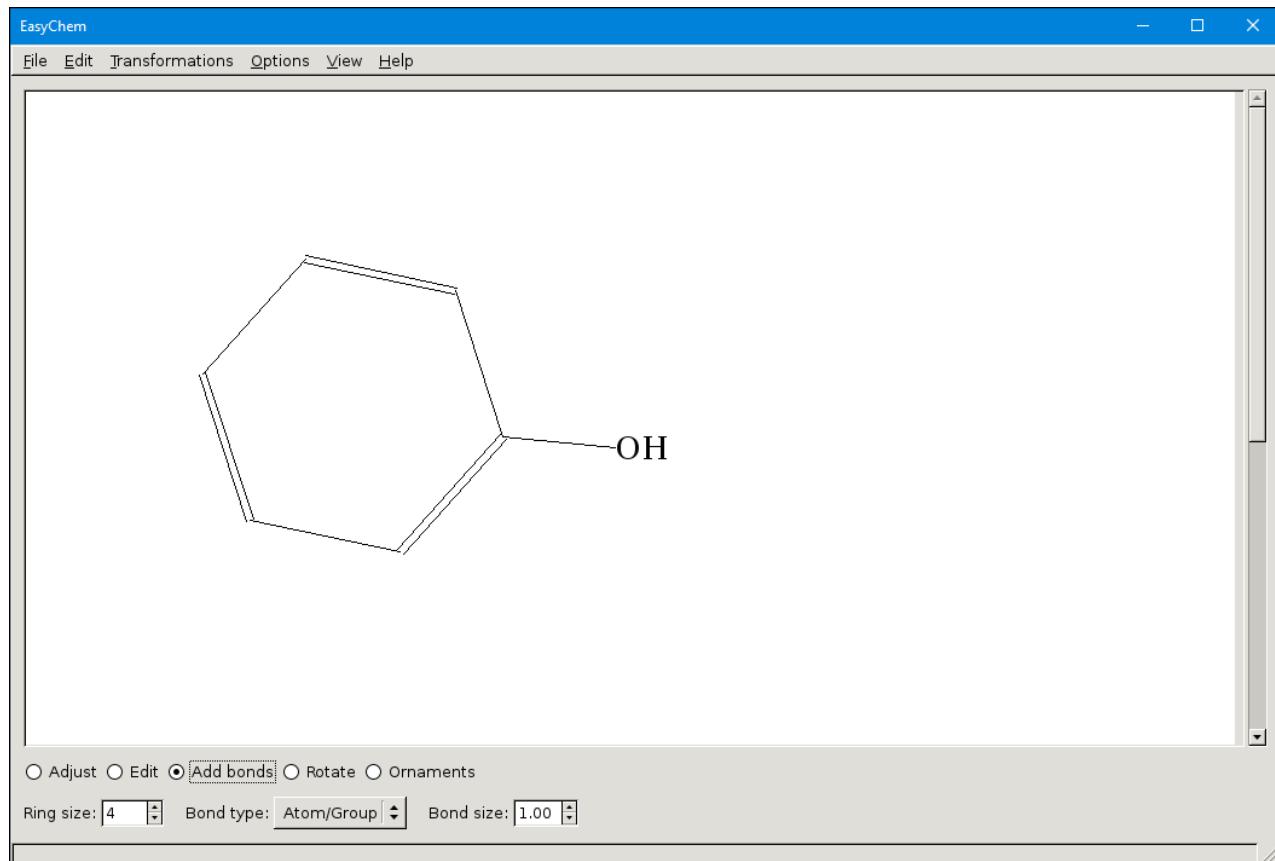
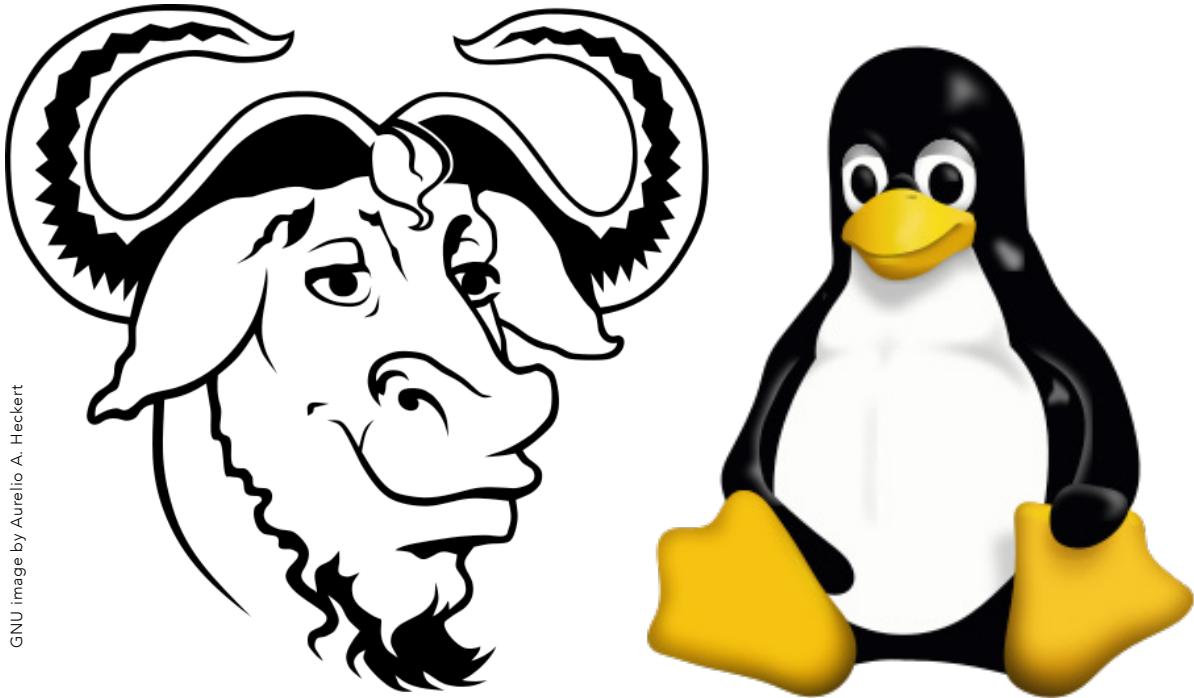


Figure 4. You can use simpler structures to build up more complicated objects.

of your documents. EasyChem provides several options for exporting your image into encapsulated PostScript files. These files have the “.eps” filename ending. If you’re writing your paper in LaTeX, this probably is your best option, as it provides the best quality. It also has the ability to save in xfig format. You then can import it into any other software that understands xfig. Both of these options are text-based file formats. They’re essentially instructions to a printing device. The final option is to export it to PDF format. Although more people may be experienced with PDF files, they’re essentially binary in nature. This means that they aren’t as easily incorporated into other documents, so you’ll most likely want to accept the default EPS file format.

—*Joey Bernard*



Is It Linux or GNU/Linux?

After putting this question to the experts, the conclusion is that no matter what you call it, it's still Linux at its core.

Should the Linux operating system be called “Linux” or “GNU/Linux”? These days, asking that question might get as many blank stares returned as asking, “Is it live or is it Memorex?”

Some may remember that the Linux naming convention was a controversy that raged from the late 1990s until about the end of the first decade of the 21st century. Back then, if you called it “Linux”, the GNU/Linux crowd was sure to start a flame war with

accusations that the GNU Project wasn't being given due credit for its contribution to the OS. And if you called it "GNU/Linux", accusations were made about political correctness, although operating systems are pretty much apolitical by nature as far as I can tell.

The brouhaha got started in the mid-1990s when Richard Stallman, among other things the founder of the Free Software Movement who penned the General Public License, began insisting on using the term "GNU/Linux" in recognition of the importance of the GNU Project to the OS. GNU was started by Stallman as an effort to build a free-in-every-way operating system based on the still-not-ready-for-prime-time Hurd microkernel.

According to this take, Linux was merely the kernel, and GNU software was the sauce that made Linux work.

Noting that the issue seems to have died down in recent years, and mindful of Shakespeare's observation on roses, names and smells, I wondered if anyone really cares anymore what Linux is called. So, I put the issue to a number of movers and shakers in Linux and open-source circles by asking the simple question, "Is it GNU/Linux or just plain Linux?"

"This has been one of the more ridiculous debates in the FOSS realm, far outdistancing the Emacs-vi rift", said Larry Cafiero, a longtime Linux advocate and FOSS writer who pulls publicity duties at the Southern California Linux Expo. "It's akin to the Chevrolet-Chevy moniker. Technically the car produced by GM is a Chevrolet, but rarely does anyone trot out all three syllables. It's a Chevy. Same with the shorthand for GNU/Linux being Linux. The shorthand version—the Chevy version—is Linux. If you insist in calling it a Chevrolet, it's GNU/Linux."

Next up was Steven J. Vaughan Nichols, who's "been covering Unix since before Linux was a grad student". He didn't mince any words.

"Enough already", he said. "RMS tried, and failed, to create an operating system: Hurd. He and the Free Software Foundation's endless attempts to plaster his GNU name to the work of Linus Torvalds and the other Linux kernel developers is disingenuous and an insult to their work. RMS gets credit for EMACS, GPL, and GCC. Linux? No."

To be fair, the use of GNU-related monikers didn't start with Stallman. An early distribution, Yggdrasil, used the term "Linux/GNU/X" in 1992, and shortly thereafter the terms "GNU/Linux" and "GNU+Linux" began showing up in Usenet and mailing-list discussions. Debian, which early on was sponsored by the Free Software Foundation, starting using the term "GNU/Linux" in 1994, which it continues to use to this day. Stallman began publicly advocating its use in 1996.

But Stallman's advocacy always put a bad taste in some people's mouths.

"For me it's always, always, always, always Linux," said Alan Zeichick, an analyst at Camden Associates who frequently speaks, consults and writes about open-source projects for the enterprise. "One hundred percent. Never GNU/Linux. I follow industry norms."

Well, somebody has to defend orthodoxy.

Gaël Duval, founder of the once uber-popular Mandrake/Mandriva distro who's now developing eelo, a privacy-respecting Android clone, pointed out that insisting on GNU/Linux might open the door wider than originally intended. "I understand people who support the idea to call it GNU/Linux", he said. "On the other hand, I do not see why in this case we wouldn't use "GNU/X11/KDE/Gnome/Whatever/Linux" for desktop systems, because graphical environments and apps are very significant in such systems.

"Personally, I'm comfortable with both Linux and GNU/Linux", he added, "but I use simply Linux, because adding complexity in communication and marketing is generally not efficient."

Richi Jennings, an independent industry analyst who pens a weekly security column on TechBeacon, expressed a similar sentiment. "Look, it's totally fair to give the GNU project its due", he said. "On the other hand, if that fairness needs to be expressed in a naming convention, why stop at GNU? Why not also recognize BSD, XINU, PBM, OpenSSL, Samba and countless other FLOSS projects that need to be included to form a workable distro?

"The bottom line is that 'Linux' is what the vast majority of people call it. So that's what it should be called, because that's how language works."

Self-professed “ace Linux guru” and Linux writer Carla Schroder said, “I’ve never called it GNU/Linux. GNU coreutils, tar, make, gcc, wget, bash and so on are still fundamental tools for a lot of Linux users. Certain people can’t let any Linux discussion pass without insisting that ‘Linux’ is only the kernel. Linux distros include a majority of non-GNU software, and I’m fine with ‘Linux’ as an umbrella term for the whole works. It’s simple and it’s widely recognized.”

Tallying the votes, it looks as if the “ayes” have it, and you can call Linux what you want. If anybody gives you any grief, tell them what Schroder told me: “Arguing is fun, but I suggest that contributing financially or in other ways to GNU/Linux/FOSS projects is more helpful.”

Or, we could argue about whether it’s FOSS or FLOSS.

Christine Hall has been a journalist since 1971. In 2001, she began writing a weekly consumer computer column, and she started covering Linux and FOSS in 2002 after making the switch to GNU/Linux. When not writing about tech, she can be found watching Netflix or anything else she can find that’s not housecleaning. Follow her on Twitter: @BrideOfLinux.

News Briefs

Visit LinuxJournal.com for
daily news briefs.

- The [Zenroom](#) project, a brand-new crypto-language virtual machine, has reached version 0.5.0. Zenroom's goal is "improving people's awareness of how their data is processed by algorithms, as well facilitate the work of developers to create and publish algorithms that can be used both client and server side." In addition, it "has no external dependencies, is smaller than 1MB, runs in less than 64KiB memory and is ready for experimental use on many target platforms: desktop, embedded, mobile, cloud and browsers." The program is free software and is licensed under the GNU LGPL v3. Its main use case is "distributed computing of untrusted code where advanced cryptographic functions are required".
- Feral Interactive released GameMode, an open-source tool that helps Linux users get the best performance out of their games. According to the press release, "GameMode instructs your CPU to automatically run in Performance Mode when playing games." *Rise of the Tomb Raider*, which was released in April 2018, will be the first release to integrate this tool. GameMode is available now via [GitHub](#).
- Linux kernel developer, free software activist and Google engineer Matthew Garrett discovered that Symantec is using a Linux distro based on the QCA Software Development Kit (QSDK) project: "This is a GPLv2-licensed, open-source platform built around the Linux-based OpenWrt Wi-Fi router operating system" (if true, this means Symantic needs to share the Norton Core Router's code). So, Garrett [tweeted](#) "Hi @NortonOnline the Norton Core is clearly running Linux and the license requires you to distribute the kernel source code so where can I get it?" (Source: [ZDNet](#).)
- The Linux Foundation [announced](#) the launch of the [LF Deep Learning Foundation](#), "an umbrella organization focused on driving open source innovation in artificial intelligence, machine learning and deep learning",

with a goal of making those technologies available to data scientists and developers. In addition, the Linux Foundation also debuted the [Acumos AI project](#), an “open source framework that makes it easy to build, share, and deploy AI apps”.

- The EFF has questions and advice for Google regarding the company’s work on “Project Maven”, which is “a U.S. Department of Defense (DoD) initiative to deploy machine learning for military purposes”. Read the [“Google Should Not Help the U.S. Military Build Unaccountable AI Systems”](#) post by Peter Eckersley and Cindy Cohn for more information.
- Richard Stallman writes [“A radical proposal to keep personal data safe”](#) in *The Guardian*: “The surveillance imposed on us today is worse than in the Soviet Union. We need laws to stop this data being collected in the first place.”
- The Qubes security-oriented OS [has released version 4.0](#). Major changes in this version include “fully virtualized VMs for enhanced security”, “a powerful new VM volume manager that makes it easy to keep VMs on external drives”, “more secure backups with scrypt for stronger key derivation and enforced encryption” and “rewritten command-line tools with new options”. See the [release notes](#) for more information, and download Qubes [here](#).
- Purism announces that its Librem laptop orders are now shipping within a week—in other words, on average, the company now can fulfill orders within five business days. See the [Purism blog](#) for more information on this milestone.

Examining Data Using Pandas

You don't need to be a data scientist to use Pandas for some basic analysis.

By Reuven M. Lerner

Traditionally, people who program in Python use the data types that come with the language, such as integers, strings, lists, tuples and dictionaries. Sure, you can create objects in Python, but those objects typically are built out of those fundamental data structures.

If you're a data scientist working with Pandas though, most of your time is spent with NumPy. NumPy might feel like a Python data structure, but it acts differently in many ways. That's not just because all of its operations work via vectors, but also because the underlying data is actually a C-style array. This makes NumPy extremely fast and efficient, consuming far less memory for a given array of numbers than traditional Python objects would do.

The thing is, NumPy is designed to be fast, but it's also a bit low level for some people. To get more functionality and a more flexible interface, many people use Pandas, a Python package that provides two basic wrappers around NumPy arrays: one-dimensional Series objects and two-dimensional Data Frame objects.



Reuven M. Lerner teaches Python, data science and Git to companies around the world. His free, weekly “better developers” email list reaches thousands of developers each week; subscribe [here](#). Reuven lives with his wife and children in Modi'in, Israel.

I often describe Pandas as “Excel within Python”, in that you can perform all sorts of calculations as well as sort data, search through it and plot it.

For all of these reasons, it’s no surprise that Pandas is a darling of the data science community. But here’s the thing: you don’t need to be a data scientist to enjoy Pandas. It has a lot of excellent functionality that’s good for Python developers who otherwise would spend their time wrestling with lists, tuples and dictionaries.

So in this article, I describe some basic analysis that everyone can do with Pandas, regardless of whether you’re a data scientist. If you ever work with CSV files (and you probably do), I definitely recommend thinking about using Pandas to open, read, analyze and even write to them. And although I don’t cover it in this article, Pandas handles JSON and Excel very well too.

Creating Data Frames

Although it’s possible to create a data frame from scratch using Python data structures or NumPy arrays, it’s more common in my experience to do so from a file. Fortunately, Pandas can load data from a variety of file formats.

Before you can do anything with Pandas, you have to load it. In a Jupyter notebook, do:

```
%pylab inline  
import pandas as pd
```

For example, Python comes with a **csv** module that knows how to handle files in CSV (comma-separated value) format. But, then you need to iterate over the file and do something with each of those lines/rows. I often find it easier to use Pandas to work with such files. For example, here’s a CSV file:

```
a,b,c,d  
e,f,g,h  
"i,j",k,l,m  
n,o,p,q
```

You can turn this into a data frame with:

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

Now you can view the contents of the data frame in Jupyter with:

```
df
```

The thing is, there are all sorts of CSV files you probably don't even think of as CSV files. For example, consider the Linux /etc/passwd file. It's not really a CSV file, but if you think about it for a moment, you'll realize that the format is the same. Each record is on a single line, and the fields are separated with ":" characters. So in this case, you'll need to use the **sep** parameter to indicate the separator:

```
filename = '/etc/passwd'
df = pd.read_csv(filename, sep=':')
df.head()
```

The command **df.head()**, much like the UNIX **head** utility, shows the first few rows of the data frame. I often use this method to inspect the data and make sure it came through okay.

In this particular case, the data came through just fine, but the first line (the root user's record!) was interpreted as header rows. Fortunately, you can fix that with another parameter:

```
df = pd.read_csv(filename, sep=':', header=None)
df.head()
```

If you pass the **header** parameter, you're telling Pandas which row of the file should be considered the headers. But if you pass **None** as a value, you're telling Pandas that none of the rows is the header. That's fine, but then you're going to get integers (starting with 0) as your column names. I'd rather use real names, so to specify

them, do:

```
df = pd.read_csv(filename, sep=':', header=None,
                  names=['username', 'password', 'uid',
                         'gid', 'name', 'homedir', 'shell'])
df.head()
```

The thing is, do you really need the **password** column? Given that on modern computers, it'll always contain "x", I think that you can leave it out. So, you can say:

```
df = pd.read_csv(filename, sep=':', header=None,
                  usecols=[0,2,3,4,5,6],
                  names=['username', 'uid', 'gid',
                         'name', 'homedir', 'shell'])
df.head()
```

The **usecols** parameter indicates which columns you want to read from the file.

Cars Towed in Chicago

Now, why would you want to use Pandas on your /etc/passwd file? You probably don't, but as you can imagine, if it works on that file, it'll work on other files too.

For example, many cities now are making data available to the general public. Chicago publishes much of its data online, so let's take a look and see, for example, what you can find out about what cars were towed in the last 90 days in Chicago.

You can go to the Chicago data portal at <https://data.cityofchicago.org/browse>. I clicked "towed vehicles"→"export"→"CSV" to get a CSV file. Then I took the downloaded file and imported it into Pandas with:

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

Nah, I'm just kidding. You think that I have the time and energy to download the file and

then load it? One of my favorite features of Pandas is the fact that most methods that work with files also can work with URLs. So, I can say:

```
url = 'https://data.cityofchicago.org/api/views/ygr5-vcbg/
       ↪rows.csv?accessType=DOWNLOAD'
df = pd.read_csv(url)
```

(Note that the URL might well have changed by the time you read this, or it might be keyed to my session information. If not, all the better, from my perspective!)

I then take a look at the data frame and discover that it has headers, that the separator is a comma and that it worked just fine. The only adjustment I'm going to make is to parse the "tow date" column as a date, so I can play with that a bit. I also indicate that because the date is in US format, the day comes first:

```
df = pd.read_csv(url, parse_dates=['Tow Date'], dayfirst=False)
df.head()
```

You'll notice that the first column now appears a bit differently, in year-month-day format. That shows there is a timestamp. You also can see that if you run the `df.info` method, which tells you about the data frame itself:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5560 entries, 0 to 5559
Data columns (total 10 columns):
Tow Date           5560 non-null datetime64[ns]
Make               5537 non-null object
Style              5538 non-null object
Model              509 non-null object
Color              5536 non-null object
Plate              4811 non-null object
State              5392 non-null object
Towed to Address  5560 non-null object
```

AT THE FORGE

```
Tow Facility Phone      5559 non-null object
Inventory Number        5560 non-null int64
dtypes: datetime64[ns](1), int64(1), object(8)
memory usage: 434.5+ KB
```

As you can see, the “Tow Date” column is now being stored as a `datetime64`, a 64-bit field containing a date and time.

So, now that you have this data, what can you do with it? One of my favorite methods is `value_counts`, which tells how many times a particular value appears in a column. So, you can say the following to find out how many vehicles were towed from each state:

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

Now, that’s going to be a long list, so it’s probably better to limit it by using `head` on the series you get back from `value_counts`:

```
df['State'].value_counts().head(10)
```

Now you’ll see which ten states have the most towed vehicles in Chicago in the last 90 days:

IL	4948
IN	148
TX	48
WI	48
MN	28
IA	27
MI	19
GA	14
FL	14
TN	13

AT THE FORGE

Not surprisingly, most of the vehicles towed in Chicago were from Illinois, with the second-highest number from Indiana, which is nearby. Less expected (to me, at least) was the large number of towed vehicles from Texas, which is not exactly nearby.

To make use of the fact that the timestamp is now a true `datetime` object, you can find out the most common dates on which vehicles were towed:

2018-03-03	215
2018-03-04	195
2018-02-24	165
2018-03-25	148
2018-03-26	140
2018-03-24	135
2018-03-15	126
2018-03-21	122
2018-02-03	120
2018-03-22	117

As you can see, there's a great deal of variation. Maybe that variation is due to the day of the week? In order to find out, you can use the `dt` proxy object Pandas provides for `datetime` columns. You can use that to extract information from the `datetime` column and then analyze it. For example, you can say:

```
df['Tow Date'].dt.dayofweek
```

This retrieves the day of the week for each of the tow dates. Then you can use `value_counts` to summarize how many vehicles were towed on each day of the week:

```
df['Tow Date'].dt.dayofweek.value_counts()
```

The results are as follows:

5 1143

```
4    831
3    820
6    798
2    794
0    640
1    534
```

The above indicates that Chicago towed far more cars on Fridays than on other days. This might always be true, or it might just be true for this 90-day sample.

You also can check to see what brand of cars are towed most often or whether there's a correlation between the color and the chances of being towed.

What if you want to know, week by week, how many cars were towed? For that, you can take advantage of a great Pandas feature, in which you can set the data frame's index to be a timestamp column:

```
df.set_index('Tow Date', inplace=True)
```

Now, instead of accessing rows by an integer index, you'll use a timestamp. But the point is not to access things that way. Rather, it's a **resample**, similar to a **GROUP BY** query in SQL. Resampling can be done in a variety of ways; here's asking for it to be on a one-week basis:

```
df.resample('1W')
```

By itself, this does nothing. But if you then count the number of rows for each week, you get much more interesting output. For every column, you get the number of entries per week. The numbers differ, because many columns have missing (**NaN**) information, which isn't included in the count. That's fine; you can just use the **Make** column, which seems to be filled in for every towed vehicle:

```
df.resample('1W').count()['Make']
```

Now you should get a very nice report, showing how many cars were towed each week:

```
Tow Date
2017-12-31    103
2018-01-07    321
2018-01-14    209
2018-01-21    241
2018-01-28    250
2018-02-04    399
2018-02-11    248
2018-02-18    328
2018-02-25    587
2018-03-04    862
2018-03-11    495
2018-03-18    601
2018-03-25    754
2018-04-01    139
```

```
Freq: W-SUN, Name: Make, dtype: int64
```

Humans see patterns more easily with graphics than with tables of numbers, so you might want to plot this, also using Pandas:

```
df.resample('1W').count()['Make'].plot()
```

That makes a line plot showing that the number of towed vehicles shot up at the start of March. Why? You could look at temperatures and a calendar to start to guess, but the fact that you can download, inspect and graph this information so easily is a good starting point.

Conclusion

Pandas is more than merely a tool for data scientists to do numeric analysis. The fact that it can read (and write) formats such as CSV, Excel and JSON make it

my primary tool for working with those formats. Add to that the fact that it can summarize data, including based on timestamps, and you probably can see why Pandas is so popular.

Resources

The home page for Pandas is [here](#), and the Jupyter notebook, which I mentioned in this article, is [here](#).

Finally, the list of “datetime components”, which you can access via the `dt` object on timestamp columns in Pandas, is [here](#). ■

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

Review: the Librem 13v2

The Librem 13—“the first 13-inch ultraportable designed to protect your digital life”—ticks all the boxes, but is it as good in real life as it is on paper?

I don’t think we’re supposed to call portable computers “laptops” anymore. There’s something about them getting too hot to use safely on your lap, so now they’re officially called “notebooks” instead. I must be a thrill-seeker though, because I’m writing this review with the Librem 13v2 directly on my lap. I’m wearing pants, but apart from that, I’m risking it all for the collective. The first thing I noticed about the Librem 13? The company refers to it as a laptop. Way to be brave, Purism!

Why the Librem?

I have always been a fan of companies who sell laptops (er, notebooks) pre-installed with Linux, and I’ve been considering buying a Purism laptop for years. When our very own Kyle Rankin started working for the company, I figured a company smart enough to hire Kyle deserved my business, so I ordered the Librem 13 (Figure 1). And when I ordered it, I discovered I could pay with Bitcoin, which made me even happier!

There are other reasons to choose Purism computers too. The company is extremely focused on privacy, and it goes so far as to have hardware switches that turn off the webcam and WiFi/



Shawn Powers is Associate Editor here at *Linux Journal*, and has been around Linux since the beginning. He has a passion for open source, and he loves to teach. He also drinks too much coffee, which often shows in his writing.

THE OPEN-SOURCE CLASSROOM

Figure 1. The 13" Librem 13v2 is the perfect size for taking on the road (photo from [Purism](#))

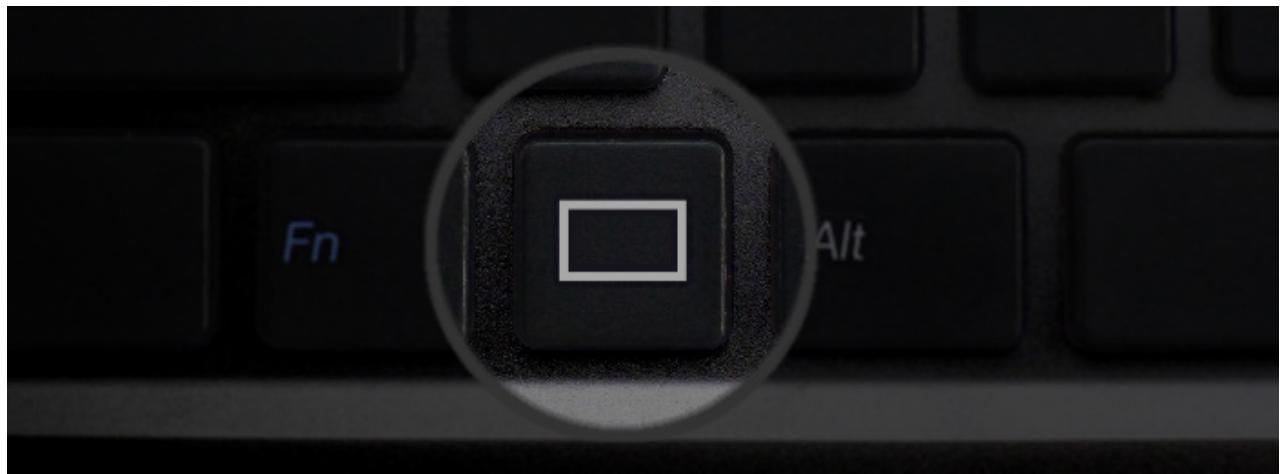


Figure 2. No Windows key here! This beats a sticker-covered Windows logo any day (photo from [Purism](#)).

Bluetooth radios. And because they're designed for open-source operating systems, there's no "Windows" key; instead there's a meta key with a big white rectangle on it, which is called the Purism Key (Figure 2). On top of all those things, the computer itself is rumored to be extremely well built, with all the bells and whistles usually available only on high-end top-tier brands.

My Test Unit

Normally when I review a product, I get whatever standard model the company sends around to reviewers. Since this was going to be my actual daily driver, I ordered what I wanted on it. That meant the following:

- i7-6500U processor, which was standard and not upgradable, and doesn't need to be!
- 16GB DDR4 RAM (default is 4GB).
- 500GB M.2 NVMe (default is 120GB SATA SSD).
- Intel HD 520 graphics (standard, not upgradable).
- 1080p matte IPS display.
- 720p 1-megapixel webcam.
- Elantech multitouch trackpad.
- Backlit keyboard.

The ports and connectors on the laptops are plentiful and well laid out. Figure 3 shows an "all sides" image from the Purism website. There are ample USB ports, full-size HDMI, and the power connector is on the side, which is my preference on laptops. In this configuration, the laptop cost slightly more than \$2000.



Figure 3. There are lots of ports, but not in awkward places (photo from [Purism](#)).

The Physical Stuff and Things

The Case The shell of the Librem 13 is anodized aluminum with a black matte texture. The screen's exterior is perfectly plain, without any logos or markings. It might seem like that would feel generic or overly bland, but it's surprisingly elegant. Plus, if you're the sort of person who likes to put stickers on the lid, the Librem 13 is a blank canvas. The underside is nearly as spartan with the company name and little else. It has a sturdy hinge, and it doesn't feel "cheap" in any way. It's hard not to compare an aluminum case to a MacBook, so I'll say the Librem 13 feels less "chunky" but almost as solid.

The Screen Once open, the screen has a matte finish, which is easy to see and doesn't

have the annoying reflection so prevalent on laptops that have a glossy finish. I'm sure there's a benefit to a glossy screen, but whatever it might be, the annoying glare nullifies the benefit for me. The Librem 13's screen is bright, has a sufficient 1080p resolution, and it's pleasant to stare at for hours. A few years back, I'd be frustrated with the limitation of a 1080p (1920x1080) resolution, but as my eyes get older, I actually prefer this pixel density on a laptop. With a higher-res screen, it's hard to read the letters without jacking up the font size, eliminating the benefit of the extra pixels!

The Keyboard I'm a writer. I'm not quite as old-school as Kyle Rankin with his mechanical PS/2 keyboard, but I am very picky when it comes to what sort of keys are on my laptop. Back in the days of netbooks, I thought a 93%-sized keyboard would be perfectly acceptable for lengthy writing. I was horribly wrong. I didn't realize a person could get cramps in their hands, but after an hour of typing, I could barely pick my nose much less type at speed.

The Librem 13's keyboard is awesome. I won't say it's the best keyboard I've ever used, but as far as laptops go, it's right near the top of the pile. Like most (good) laptops, the Librem 13 has Chicklet style keys, but the subtleties of click pressure, key travel, springiness factor and the like are very adequate. The Librem 13v2 has a new feature, in that the keys are backlit (Figure 4). Like most geeks, I'm a touch typist, but in a dark room, it's still incredibly nice to have the backlight. Honestly, I'm not sure why I appreciate the backlight so much, but I've tried both on and off, and I really hate when the keyboard is completely dark. That might just be a personal preference, but having the choice means everyone is happy.

The Trackpad The Librem 13 has a huge (Figure 5), glorious trackpad. Since Apple is known for having quality hardware, it's only natural to compare the Librem 13 to the Macbook Pro (again). For more than a decade, Apple has dominated the trackpad scene. Using a combination of incredible hardware and silky smooth software, the Apple trackpad has been the gold standard. Even if you hate Apple, it's impossible to deny its trackpads have been better than any other—until recently. The Librem 13v2 has a trackpad that is 100% as nice as MacBook trackpads. It is large, supports “click anywhere” and has multipoint support with gestures. What does all that mean? The things that have



Figure 4. I don't notice the keyboard after hours of typing, which is what you want in a keyboard (photo from [Purism](#)).

made Apple King of Trackpad Land are available not only on another company's hardware, but also with Linux. My favorite combination is two-finger scrolling with two-finger clicking for "right-click". The trackpad is solid, stable and just works. I'd buy the Librem 13 for the trackpad alone, but that's just a throwaway feature on the website.

The Power Adapter It might seem like a silly thing to point out, but the Librem 13 uses a standard 19-volt power adapter with a 5.5mm/2.5mm barrel connector. Why is that significant? Because I accidentally threw my power supply away with the box, and I was worried I'd have to special-order a new one. Thankfully, the dozen or so power supplies I have in my office from netbooks, NUCs and so on fit the Librem 13 perfectly. Although I don't recommend throwing your power supply away, it's nice to know replacements are easy to find online and probably in the back of your tech junk drawer.

Hardware Switches I'm not as security-minded as perhaps I should be. I'm definitely not as security-minded as many *Linux Journal* readers. I like that the Librem 13 has physical switches that disconnect the webcam and WiFi/Bluetooth.



Figure 5. This trackpad is incredible. It's worth buying the laptop for this feature alone (photo from [Purism](#)).

For many of my peers, the hardware switches are the single biggest selling point. There's not much to say other than that they work. They physically switch right to left as opposed to a toggle, and it's clear when the physical connection to the devices have been turned off (Figure 6). With the Librem 13, there's no need for electrical tape over the webcam. Plus, using your computer while at DEFCON isn't like wearing a meat belt at the dog pound. Until nanobots become mainstream, it's hard to beat the privacy of a physical switch.

I worried a bit about how the operating systems would handle hardware being physically disconnected. I thought perhaps you'd need special drivers or custom

software to handle the disconnect/reconnect. I'm happy to report all the distributions I've tried have handled the process flawlessly. Some give a pop-up about devices being connected, and some quietly handle it. There aren't any reboots required, however, which was a concern I had.

Audio/Video I don't usually watch videos on my laptop, but like most people, I will show others around me funny YouTube videos. The audio on the Librem 13 is sufficiently loud and clear. The video subsystem (I mention more about that later) plays video just fine, even full screen. There is also an HDMI port that works like an HDMI connection should. Modern Linux distributions are really good at handling external displays, but every time I plug in a projector and it just works, my heart sings!

PureOS

The Librem 13 comes with Purism's "PureOS" installed out of the box. The OS is Debian-based, which I'm most comfortable using. PureOS uses its own repository, hosted and maintained by Purism. One of the main reasons PureOS exists is so that Purism can make sure there is no closed-source code or proprietary drivers installed on its computers. Although the distro includes tons of packages, the really impressive



Figure 6. It's not possible to accidentally turn these switches on or off, which is awesome (photo from [Purism](#)).

thing is how well the laptop works without any proprietary code. The “purity” of the distribution is comforting, but the standout feature is how well Purism chose the hardware. Anyone who has used Linux laptops knows there’s usually a compromise regarding proprietary drivers and wrappers in order to take full advantage of the system. Not so with the Librem 13 and PureOS. Everything works, and works well.

PureOS works well, but the most impressive aspect of it is what it does *while* it’s working. The pre-installed hard drive walks you through encryption on the first boot. The Firefox-based browser (called “Purebrowser”) uses HTTPS Everywhere, defaults to DuckDuckGo as the search engine, and if that’s not sufficient for your privacy needs, it includes the Tor browser as well. The biggest highlight for me was that since Purebrowser is based on Firefox, the browsing experience wasn’t lacking. It didn’t “feel” like I was running a specialized browser to protect my identity, which makes doing actual work a lot easier.

Other Distributions

Although I appreciate PureOS, I also wanted to try other options. Not only was I curious, but honestly, I’m stuck in my ways, and I prefer Ubuntu MATE as my desktop interface. The good news is that although I’m not certain the drivers are completely open source, I am sure that Ubuntu installs and works very well. There are a few glitches, but nothing serious and nothing specific to Ubuntu (more on those later).

I tried a handful of other distributions, and they all worked equally well. That makes sense, since the hardware is 100% Linux-compatible. There was an issue with most distributions, which isn’t the fault of the Librem 13. Since my system has the M.2 NVMe as opposed to a SATA SSD, most installers have a difficult time determining where to install the bootloader. Frustratingly, several versions of the Ubuntu installer don’t let the manual selection of the correct partition to be chosen either. The workaround seems to be setting up hard drive partitions manually, which allows the bootloader partition to be selected. (For the record, it’s /dev/nvme0n1.) Again, this isn’t Purism’s fault; rather, it’s the Linux community getting up to speed with NVMe drives and EFI boot systems.

Quirks

There are a few oddities with a freshly installed Librem 13. Most of the quirks are ironed out if you use the default PureOS, but it's worth knowing about the issues in case you ever switch.

NVMe Thing As I mentioned, the bootloader problem with an NVMe system is frustrating enough that it's worth noting again in this list. It's not impossible to deal with, but it can be annoying.

Backslash Key The strangest quirk with the Librem 13 is the backslash key. It doesn't map to backslash. On every installation of Linux, when you try to type backslash, you get the “less than” symbol. Thankfully, fixing things like keyboard scancodes is simple in Linux, but it's so strange. I have no idea how the non-standard scancode slipped through QA, but nonetheless, it's something you'll need to deal with. There's a [detailed thread on the Purism forum](#) that makes fixing the problem simple and permanent.

Trackpad Stuff As I mentioned before, the trackpad on the Librem 13 is the nicest I've ever used on a non-Apple laptop. The oddities come with various distributions and their trackpad configuration software. If your distribution doesn't support the gestures and/or multipoint settings you expect, rest assured that the trackpad supports every feature you are likely to desire. If you can't find the configuration in your distro's setup utility, you might need to dig deeper.

The Experience and Summary

The Librem 13 is the fastest laptop I've ever used. Period. The system boots up from a cold start faster than most laptops wake from sleep. Seriously, it's insanely fast. I ran multiple VMs without any significant slowdowns, and I was able to run multiple video-intensive applications without thinking “laptops are so slow” or anything like that.

The only struggle I had was when I tried to use the laptop for live streaming to Facebook using OBS (Open Broadcast Studio). The live transcoding really taxed the CPU. It was able to keep up, but normally on high-end computers, it's easier to offload

the transcoding to a discrete video card. Unfortunately, there aren't any non-Intel video systems that work well without proprietary drivers. That means even though the laptop is as high-end as they get, the video system works well, but it can't compare to a system with a discrete NVIDIA video card.

Don't let the live streaming situation sour your view of the Librem 13 though. I had to try **really** hard to come up with something that the Librem 13 didn't chew through like the desktop replacement it is. And even with my live streaming situation, I was able to transcode the video using the absurdly fast i7 CPU. This computer is lightning fast, and it's easily the best laptop I've ever owned. More than anything, I'm glad this is a system I purchased and not a "review copy", so I don't have to send it back! ■

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

What's New in Kernel Development

By Zack Brown

Speeding Up Netfilter (by Avoiding Netfilter)

Imre Palik tried to speed up some of Linux's networking code but was met with stubborn opposition. Essentially, he wanted networking packets to bypass the **netfilter** code unless absolutely necessary. Netfilter, he said, was designed for flexibility at the expense of speed. According to his tests, bypassing it could speed up the system by as much as 15%.

Netfilter is a piece of infrastructure that gives users a tremendous amount of power and flexibility in processing and restricting networking traffic. Imre's idea was that if the user didn't want to filter network packets, the netfilter code shouldn't even be traversed. He therefore wanted to let users disable netfilter for any given firewall that didn't need it.

There was some initial interest and also some questions about how he'd calculated his 15% speed increase. **Florian Westphal** tried to reason out where the speedup might have come from. But **David S. Miller** put his foot down, saying that any speedup estimates were just guesses until they were properly analyzed via **perf**.



Zack Brown is a tech journalist at *Linux Journal* and *Linux Magazine*, and is a former author of the “Kernel Traffic” weekly newsletter and the “Learn Plover” stenographic typing tutorials. He first installed Slackware Linux in 1993 on his 386 with 8 megs of RAM and had his mind permanently blown by the Open Source community. He is the inventor of the *Crumble* pure strategy board game, which you can make yourself with a few pieces of cardboard. He also enjoys writing fiction, attempting animation, reforming Labanotation, designing and sewing his own clothes, learning French and spending time with friends'n'family.

David absolutely refused to apply networking patches without a more reliable indication that they would improve the situation.

Imre explained his testing methods and asserted that they seemed sound to him. But **Pablo Neira Ayuso** felt that Imre's approach was too haphazard. He said there needed to be a more generic way to do that sort of testing.

David was completely unsatisfied by Imre's tests. Instead of trying to work around netfilter, even in cases where there were no actual filters configured, he said, the proper solution was to speed up netfilter so it wouldn't be necessary to bypass it. David said, "We need to find a clean and generic way to make the netfilter hooks as cheap as possible when netfilter rules are not in use."

David Woodhouse, on the other hand, felt that a 15% speedup was a 15% speedup, and we shouldn't look a gift horse in the mouth.

But, David M stood firm. The netfilter hooks were the fundamental issue, he said, and "I definitely would rather see the fundamental issue addressed rather than poking at it randomly with knobs for this case and that."

David W and others started hunting around for ways to satisfy David M without actually recoding the netfilter hooks. David W suggested having the hooks disable themselves automatically if they detected that they wouldn't be useful.

Ultimately there was no conclusion to the thread, although it seems clear that for the moment, Imre's code is dead in the water. The problem with that is that 15% really is 15%, and speedups are good even if they're not perfect. It's conceivable that no one will come up with a good way to fix netfilter hooks, and that Imre's patch will receive better testing and more meaningful performance numbers. At that point, it's possible even David M would say okay.

Read-Only Memory

Igor Stoppa posted a patch to allow **kernel memory pools** to be made read-only. Memory pools are a standard way to group memory allocations in Linux so their time

cost is more predictable. With Igor's patch, once a memory pool was made read-only, it could not be made read-write again. This would secure the data for good and against attackers. Of course, you could free the memory and destroy the pool. But short of that, the data would stay read-only.

There was not much controversy about this patch. **Kees Cook** felt that **XFS** would work well with the feature. And, having an actual user would help Igor clarify the usage and nail down the API.

This apparently had come up at a recent conference, and **Dave Chinner** was ready for Igor's patch. He remarked, "we have a fair amount of static data in XFS that we set up at mount time and it never gets modified after that. I'm not so worried about VFS level objects (that's a much more complex issue) but there is a lot of low hanging fruit in the XFS structures we could convert to write-once structures."

Igor said this was exactly the kind of thing he'd had in mind.

A bunch of folks started talking about terminology and use cases, and speculated on further abilities. No one had any negative comment, and everyone was excited to get going with it.

The thing about a patch like this is that people can use the feature or not. It helps them with security, or it costs them nothing. It adds an ability but adds no complexity to the code. Unless something weird happens, I'd expect this patch to go into the kernel as soon as the API stabilizes.

Working around Intel Hardware Flaws

Efforts to work around serious hardware flaws in **Intel** chips are ongoing. **Nadav Amit** posted a patch to improve compatibility mode with respect to Intel's **Meltdown** flaw. Compatibility mode is when the system emulates an older CPU in order to provide a runtime environment that supports an older piece of software that relies on the features of that CPU. The thing to be avoided is to emulate massive security holes created by hardware flaws in that older chip as well.

diff -u

In this case, Linux is already protected from Meltdown by use of **PTI** (page table isolation), a patch that went into Linux 4.15 and that was subsequently backported all over the place. However, like the **BKL** (big kernel lock) in the old days, PTI is a heavy-weight solution, with a big impact on system speed. Any chance to disable it without reintroducing security holes is a chance worth exploring.

Nadav's patch was an attempt to do this. The goal was "to disable PTI selectively as long as x86-32 processes are running and to enable global pages throughout this time."

One problem that Nadav acknowledged was that since so many developers were actively working on anti-Meltdown and anti-**Spectre** patches, there was plenty of opportunity for one patch to step all over what another was trying to do. As a result, he said, "the patches are marked as an RFC since they (specifically the last one) do not coexist with Dave Hansen's enabling of global pages, and might have conflicts with Joerg's work on 32-bit."

Andrew Cooper remarked, chillingly:

Being 32bit is itself sufficient protection against Meltdown (as long as there is nothing interesting of the kernel's mapped below the 4G boundary). However, a 32bit compatibility process may try to attack with Spectre/SP2 to redirect speculation back into userspace, at which point (if successful) the pipeline will be speculating in 64bit mode, and Meltdown is back on the table. SMEP will block this attack vector, irrespective of other SP2 defenses the kernel may employ, but a fully SP2-defended kernel doesn't require SMEP to be safe in this case.

And Dave, nearby, remarked, "regardless of Meltdown/Spectre, SMEP is valuable. It's valuable to everything, compatibility-mode or not."

SMEP (Supervisor Mode Execution Protection) is a hardware mode, whereby the OS can set a register on compatible CPUs to prevent userspace code from running. Only code that already has root permissions can run when SMEP is activated.

Andy Lutomirski said that he didn't like Nadav's patch because he said it drew a

distinction between “compatibility mode” tasks and “non-compatibility mode” tasks. Andy said no such distinction should be made, especially since it’s not really clear how to make that distinction, and because the ramifications of getting it wrong might be to expose significant security holes.

Andy felt that a better solution would be to enable and disable 32-bit mode and 64-bit mode explicitly as needed, rather than guessing at what might or might not be compatibility mode.

The drawback to this approach, Andy said, was that old software would need to be upgraded to take advantage of it, whereas with Nadav’s approach, the judgment would be made automatically and would not require old code to be updated.

Linus Torvalds was not optimistic about any of these ideas. He said, “I just feel this all is a nightmare. I can see how you would want to think that compatibility mode doesn’t need PTI, but at the same time it feels like a really risky move to do this.” He added, “I’m not seeing how you keep user mode from going from compatibility mode to L mode with just a far jump.”

In other words, the whole patch, and any alternative, may just simply be a bad idea.

Nadav replied that with his patch, he tried to cover every conceivable case where someone might try to break out of compatibility mode and to re-enable PTI protections if that were to happen. Though he did acknowledge, “There is one corner case I did not cover (LAR) and Andy felt this scheme is too complicated. Unfortunately, I don’t have a better scheme in mind.”

Linus remarked:

Sure, I can see it working, but it’s some really shady stuff, and now the scheduler needs to save/restore/check one more subtle bit.

And if you get it wrong, things will happily work, except you’ve now defeated PTI. But you’ll never notice, because you won’t be testing for it, and the only people who will

are the black hats.

This is exactly the “security depends on it being in sync” thing that makes me go “eww” about the whole model. Get one thing wrong, and you’ll blow all the PTI code out of the water.

So now you tried to optimize one small case that most people won’t use, but the downside is that you may make all our PTI work (and all the overhead for all the _normal_ cases) pointless.

And Andy also remarked, “There’s also the fact that, if this stuff goes in, we’ll be encouraging people to deploy 32-bit binaries. Then they’ll buy Meltdown-fixed CPUs (or AMD CPUs!) and they may well continue running 32-bit binaries. Sigh. I’m not totally a fan of this.”

The whole thread ended inconclusively, with Nadav unsure whether folks wanted a new version of his patch.

The bottom line seems to be that Linux has currently protected itself from Intel’s hardware flaws, but at a cost of perhaps 5% to 30% efficiency (the real numbers depend on how you use your system). And although it will be complex and painful, there is a very strong incentive to improve efficiency by adding subtler and more complicated workarounds that avoid the heavy-handed approach of the PTI patch. Ultimately, Linux will certainly develop a smooth, near-optimal approach to Meltdown and Spectre, and probably do away with PTI entirely, just as it did away with the BKL in the past. Until then, we’re in for some very ugly and controversial patches.

Cleaning Up the VSF

Dongsu Park posted a patch in collaboration with **Eric W. Biederman**, and originally inspired by **Seth Forshee**, to make an odd adjustment to the filesystem code. Specifically, they wanted any user with the capability **CAP_CHOWN** over a filesystem’s superblock, to be able to **chown** (change the owner) of files within that filesystem.

Apparently, this would become an issue only when running a virtual system (that is, a container) on top of a running Linux system and if the underlying filesystem had files with

user IDs or group IDs that didn't map to anything in the current user namespace within the container. Before writing such files to disk, you'd have to run **chown** on those files to tell them to which owner to map. Otherwise, writing such files to disk without a good uid or gid mapping would corrupt those fields in the filesystem.

A couple technical comments were made about the patch, but **Miklos Szeredi** expressed confusion about why the problem solved by the patch might ever be triggered. If you can't **chown** the file to be owned by the user doing the writing, he remarked, how can you write the file in order to produce the corruption? To which Eric replied that the patch wasn't actually intended to be a fix for any real problem. No one was in danger of hitting this particular problem.

The patch, he explained, was part of a larger strategy of shoring up the virtual file system (VFS) and making sure it handled all generic cases correctly—whether or not those cases could occur in real life. The goal was to draw a clear distinction between problems showing up in real-world filesystems and problems showing up at the lower VFS level. This way, when bug reports came in, it would be more straightforward to associate them with particular filesystems, rather than trying to debug them in the VFS.

He said, “In this case the generic concern is what happens when the uid is read from the filesystem and it gets mapped to INVALID_UID and then the inode for that file is written back. That is a trap for the unwary filesystem implementation and not a case that I think anyone will actually care about.”

So essentially, it was not even a housekeeping patch, but instead a patch to make housekeeping itself easier.

Note: if you're mentioned in this article and have a response, please send the text to ljeditor@linuxjournal.com, and we'll run it in the next Letters section and post it on the website as an addendum to the original article. ■

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

Generating Good Passwords



Dave works on a new method for generating secure passwords with the help of 1Password.

By **Dave Taylor**

A while back I shared a script concept that would let you enter a proposed password for an account and evaluate whether it was very good (well, maybe “secure” would be a better word to describe the set of tests to ensure that the proposed password included uppercase, lowercase, a digit and a punctuation symbol to make it more unguessable).

Since then, however, I’ve really been trying personally to move beyond mnemonic passwords of any sort to those that look more like gobbledegook. You know what I mean—passwords like fRz3li,4qDP? that turn out to be essentially random and, therefore, impossible to crack using any sort of dictionary attack.

Aiding me with this is the terrific password manager 1Password. You can learn more about it [here](#), but the key feature I’m using is a combination of having it securely store my passwords for hundreds of websites and having a simple and straightforward password generator feature (Figure 1).

If I’m working on the command line, however, why pop out to the program to get a good password? Instead, a script can do the same thing, particularly if I again tap into the useful `$RANDOM`

Dave Taylor has been hacking shell scripts on Unix and Linux systems for a really long time. He’s the author of *Learning Unix for Mac OS X* and *Wicked Cool Shell Scripts*. He can be found on Twitter as @DaveTaylor and you can reach him through his tech Q&A site [Ask Dave Taylor](#).

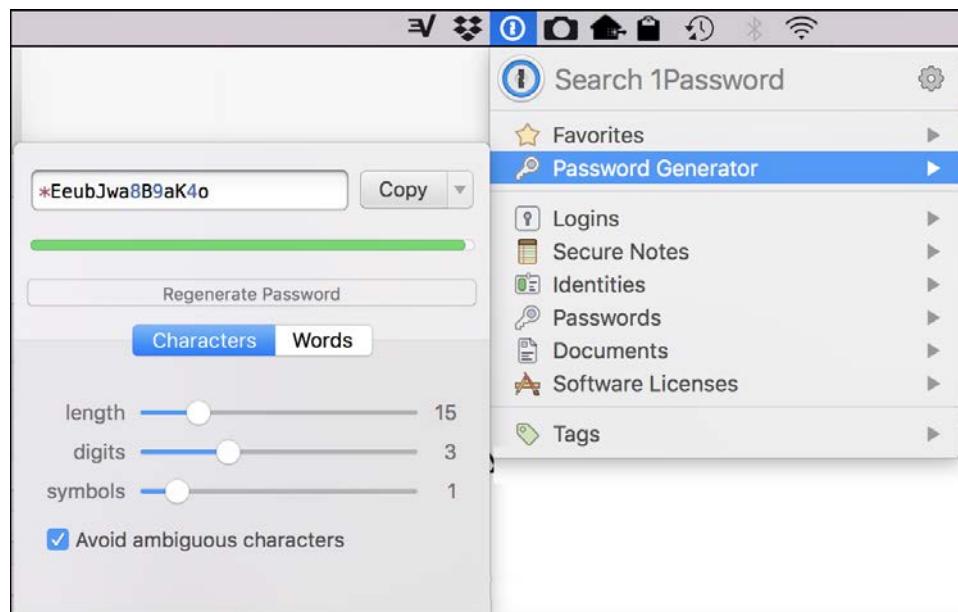


Figure 1. 1Password Password Generation System

shortcut for generating random numbers.

Generating Secure Passwords

The easiest way to fulfill this task is to have a general-purpose approach to generating a random element from a specific set of possibilities. So, a random uppercase letter might be generated like this:

```
uppers="ABCDEFGHIJKLMNOPQRSTUVWXYZ"  
  
letter=${uppers:$(( $RANDOM % 26 )):1}
```

The basic notational convention used here is the super handy Bash shell variable slicing syntax of:

```
 ${variable:startpoint:charcount}
```

To get the first character only of a variable, for example, you can simply reference it as:

```
 ${variable:1:1}
```

WORK THE SHELL

That's easy enough. Instead of a fixed reference number, however, I'm using `$(($RANDOM % 26))` as a way to generate a value between 0-25 that's different each time.

Add strings that contain all the major character classes you seek and you've got a good start:

```
lowers="abcdefghijklmnopqrstuvwxyz"  
digits="0123456789"  
punct="()./?;:[{}]|=+-_*^%$#@!~" # skip quotes
```

To get even fancier, there's another notation `#{#variable}` that returns the number of characters in a variable, so the following shows that there are 24 characters in that particular string:

```
$ echo ${#punct}  
24
```

To be maximally flexible, every reference can include that too, allowing me to add or delete specific punctuation marks at will:

```
 ${punct:$(( $RANDOM % ${#punct} )):1}
```

It's starting to look a bit like my cat ran over the keyboard, but that's why I add all the spaces here. Many script writers tend to eschew all those spaces and use shorter variable names, but in my opinion, something like this is definitely harder to read and understand:

```
 ${p:$(( $RANDOM % ${#p} )):1}
```

In fact, it reminds me of an old programming language called APL where it was generally accepted that it was easier to rewrite code than understand what someone else had done within a program. Yikes.

This solves the challenge of producing a random character in a specific charset. The next piece of the script builds a sequence of these random characters to create a string of the

WORK THE SHELL

desired length and complexity.

Add lowercase and a constrained set of punctuation and some rules on how many of each you want, and you can make some pretty complicated passwords. To start, let's just focus on a random sequence of n uppercase letters.

That's easily done:

```
while [ ${#password} -lt $length ] ; do
    letter=${uppers:$(( $RANDOM % ${#uppers} )):1}
    password="$password$letter"
done
```

Remember that the `#{#var}` notation produces the length of the current value of that variable, so this is an easy way to build up the `$password` variable until it's equal to the target length as specified in `$length`.

Here's a quick test run or two:

```
$ sh makepw.sh
password generated = HDBYPMVETY
password generated = EQKIQRCCTZT
password generated = DNCJMMXNHM
```

Looks great! Now the bigger challenge is to pick randomly from a set of choices. There are a couple ways to do it, but let's use a `case` statement, like this:

```
while [ ${#password} -lt $length ] ; do
    case $(( $RANDOM % 4 )) in
        0 ) letter=${uppers:$(( $RANDOM % ${#uppers} )):1} ;;
        1 ) letter=${lowers:$(( $RANDOM % ${#lowers} )):1} ;;
        2 ) letter=${punct:$(( $RANDOM % ${#punct} )):1} ;;
        3 ) letter=${digits:$(( $RANDOM % ${#digits} )):1} ;;
```

WORK THE SHELL

```
esac
password="${password}${letter}"
done
```

Since you're basically weighing upper, lower, digits and punctuation the same, it's not a huge surprise that the resultant passwords are rather punctuation-heavy:

```
$ sh makepw.sh
password generated = 8t&4n=&b(B
password generated = 5=B]9?CEqQ
password generated = |10|*;%&A;
```

These are all great passwords, impossible to guess algorithmically (and, yeah, hard to remember too, but that's an inevitable side effect of this kind of password algorithm).

But let's say that you'd rather have it be biased toward letters and digits than punctuation, because it's so much easier to type. That can be done by simply expanding the random number choice and assigning more than one value to those options you want to have appear more frequently, like this:

```
while [ ${#password} -lt $length ] ; do
    case $(( $RANDOM % 7 )) in
        0|1 ) letter=${uppers:$(( $RANDOM % ${#uppers} )):1} ;;
        2|3 ) letter=${lowers:$(( $RANDOM % ${#lowers} )):1} ;;
        4|5 ) letter=${punct:$(( $RANDOM % ${#punct} )):1} ;;
        6 ) letter=${digits:$(( $RANDOM % ${#digits} )):1} ;;
    esac
    password+="$password$letter"
done
```

This works better, and the results are a bit less like a cat running across your keyboard:

```
$ sh makepw.sh
```

```
password generated = /rt?7D8QxR
password generated = us&*gpyB*-_
password generated = rB}?2:)eJM
password generated = PC34j0D_}2
```

Next time, maybe I'll switch things around and let the user specify desired length and probability of punctuation being added to the password produced. Stay secure until then.

Revisiting Last Month's Script: Randomly Switching Upper and Lowercase

[Last time](#), I talked about what's known informally as l33t-speak, a series of letter and letter-pair substitutions that marks the jargon of the hacker elite (or some subset of hacker elite, because I'm pretty sure that real computer security experts don't need to substitute vowels with digits to sound cool and hip).

Still, it was an interesting exercise as a shell-scripting problem, because it's surprisingly simply to adapt a set of conversion rules into a sequence of commands. I sidestepped one piece of it, however, and that's what I want to poke around with: changing uppercase and lowercase letters somewhat randomly.

This is where "Linux Journal" might become "LiNUx jOurNAl", for example. Why? Uhm, because it's a puzzle to solve. Jeez, you ask such goofy questions of me!

Breaking Down a Line Letter by Letter

The first and perhaps most difficult task is to take a line of input and break it down letter by letter so each can be analyzed and randomly transliterated. There are lots of ways to accomplish this in Linux (of course), but I'm going to use the built-in Bash substring variable reference sequence. It looks like this:

```
 ${variable:index:length}
```

So to get just the ninth character of variable `input`, for example, I could use `${input:9:1}`. Bash also has another handy variable reference that produces the

length of the value of a particular variable: `$[#variable]`. Put the two together, and here's the basic initialization and loop:

```
input="$*"
length="${#input}"

while [ $charindex -lt $length ]
do
    char="${input:$charindex:1}"
    # conversion occurs here
    newstring="${newstring}${char}"
    charindex=$(( $charindex + 1 ))
done
```

Keep in mind that `charindex` is initialized to 0, and `newstring` is initialized to "", so you can see how this quickly steps through every character, adding it to `newstring`. "Conversion occurs here" is not very exciting, but that's the placeholder you need.

Lower, Meet Upper, and Vice Versa

Last time I also showed a quick and easy way to choose a number 1–10 randomly, so you can sometimes have something happen and other times not happen. In this command:

```
doit=$(( $RANDOM % 10 ))      # random virtual coin flip
```

Let's say there's only a 30% chance that an uppercase letter will convert to lowercase, but a 50% chance that a lowercase letter will become uppercase. How do you code that? To start, let's get the basic tests:

```
if [ -z "$(echo "$char" | sed -E 's/[:lower:]/'' )" ]
then
    # it's a lowercase character
elif [ -z "$(echo "$char" | sed -E 's/[:upper:]/'' )" ]
then
```

WORK THE SHELL

```
# it's uppercase  
fi
```

This is a classic shell-script trick: to ascertain if a character is a member of a class, replace it with null, then test to see if the resultant string is null (the `-Z` test).

The last bit's easy. Generate the random number, then if it's below the threshold, transliterate the `char`; otherwise, do nothing. Thus:

```
if [ -z "$(echo \"$char\" | sed -E 's/[[[:lower:]]//')" ]  
then  
    # lowercase. 50% chance we'll change it  
    if [ $doit -lt 5 ] ; then  
        char=$(echo $char | tr '[[[:lower:]]]' '[[[:upper:]]]'  
    fi  
    elif [ -z "$(echo \"$char\" | sed -E 's/[[[:upper:]]//')" ]  
    then  
        # uppercase. 30% chance we'll change it  
        if [ $doit -lt 3 ] ; then  
            char=$(echo $char | tr '[[[:upper:]]]' '[[[:lower:]]]'  
        fi  
    fi
```

Put it all together and you have this Frankenstein's monster of a script:

```
$ sh changecase.sh Linux Journal is a great read.  
LiNuX JoURNal is a GrEaT ReAd.  
$ !!  
LiNuX journAl IS a gREat rEAd  
$
```

Now you're ready for writing some ransom notes, it appears! ■

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

DEEP DIVE

PRIVACY

Data Privacy: Why It Matters and How to Protect Yourself

When it comes to privacy on the internet, the safest approach is to cut your Ethernet cable or power down your device. But, because you can't really do that and remain somewhat productive, you need other options. This article provides a general overview of the situation, steps you can take to mitigate risks and finishes with a tutorial on setting up a virtual private network.

By Petros Koutoupis

Sometimes when you're not too careful, you increase your risk of exposing more information than you should, and often to the wrong recipients—Facebook is a prime example. The company providing the social-media product of the same name has been under scrutiny recently and for good reason. The point wasn't that Facebook directly committed the atrocity, but more that a company linked to the previous US presidential election was able to access and inappropriately store a large trove of user data from the social-media site. This data then was used to target specific individuals. How did it happen though? And what does that mean for Facebook (and other social-media) users?

In the case of Facebook, a data analysis firm called Cambridge Analytica was given permission by the social-media site to collect user data from a downloaded

**DEEP
DIVE**



application. This data included users' locations, friends and even the content the users "liked". The application supposedly was developed to act as a personality test, although the data it mined from users was used for so much more and in what can be considered not-so-legal methods.

At a high level, what does this all mean? Users allowed a third party to access their data without fully comprehending the implications. That data, in turn, was sold to other agencies or campaigns, where it was used to target those same users and their peer networks. Through ignorance, it becomes increasingly easy to "share" data and do so without fully understanding the consequences.

Getting to the Root of the Problem

For some, deleting your social-media account may not be an option. Think about it. By deleting your Facebook account, for example, you may essentially be deleting the platform that your family and friends choose to share some of the greatest events in their lives. And although I continue to throw Facebook in the spotlight, it isn't the real problem. Facebook merely is taking advantage of a system with zero to no regulations on how user privacy should be handled. Honestly, we, as a society, are making up these rules as we go along.

Recent advancements in this space have pushed for additional protections for web users with an extra emphasis on privacy. Take the General Data Protection Regulation (GDPR), for example. Established by the European Union (EU), the GDPR is a law directly affecting data protection and privacy for all individuals within the EU. It also addresses the export or use of said personal data outside the EU, forcing other regions and countries to redefine and, in some cases, reimplement their services or offerings. This is most likely the reason why you may be seeing updated privacy policies spamming your inboxes.

Now, what exactly does GDPR enforce? Again, the primary objective of GDPR is to give EU citizens back control of their personal data. The compliance deadline is set for May 25, 2018. For individuals, the GDPR ensures that basic identity (name, address and so on), locations, IP addresses, cookie data, health/genetic data, race/ethnic data,

sexual orientation and political opinions are always protected. And once the official deadline hits, it initially will affect companies with a presence in an EU country or offering services to individuals living in EU countries. Aside from limiting the control a company would have over your private data, the GDPR also places the burden on the same company to be more upfront and honest with any sort of data breach that could have resulted in the same data from being inappropriately accessed.

Although recent headlines have placed more focus around social-media sites, they are not the only entities collecting data about you. The very same concepts of data collection and sharing even apply to the applications installed on your mobile devices. Home assistants, such as Google Home or Amazon Alexa, constantly are listening. The companies behind these devices or applications stand by their claims that it's all intended to enrich your experiences, but to date, nothing prevents them from misusing that data—that is, unless other regions and countries follow in the same footsteps as the EU.

Even if those companies harvesting your data don't ever misuse it, there is still the risk of a security breach (a far too common occurrence) placing that same (and what could be considered private) information about you into the wrong hands. This potentially could lead to far more disastrous results, including identity theft.

Where to Start?

Knowing where to begin is often the most difficult task. Obviously, use strong passwords that have a mix of uppercase and lowercase characters, numbers and punctuation, and don't use the same password for every online account you have. If an application offers two-factor authentication, use it.

The next step involves reviewing the settings of all your social-media accounts. Limit the information in your user profile, and also limit the information you decide to share in your posts, both public and private. Even if you mark a post as private, that doesn't mean no one else will re-share it to a more public audience. Even if you believe you're in the clear, that data eventually could leak out. So, the next time you decide to post about legalizing marijuana or "like" a post about something extremely political or

polarizing, that decision potentially could impact you when applying for a new job or litigating a case in court—that is, anything requiring background checks.

The information you do decide to share in your user profile or in your posts doesn't stop with background checks. It also can be used to give unwanted intruders access to various non-related accounts. For instance, the name of your first pet, high school or the place you met your spouse easily can be mined from your social-media accounts, and those things often are used as security questions for vital banking and e-commerce accounts.

Social-Media-Connected Applications

Nowadays, it's easy to log in to a new application using your social-media accounts. In some cases, you're coerced or tricked into connecting your account with those applications. Most, if not all, social-media platforms provide a summary of all the applications your account is logged in to. Using Facebook as an example, navigate to your Settings page, and click the Apps Settings page. There you will find such a list (Figure 1).

As you can see in Figure 1, I'm currently connected to a few accounts, including the Linux

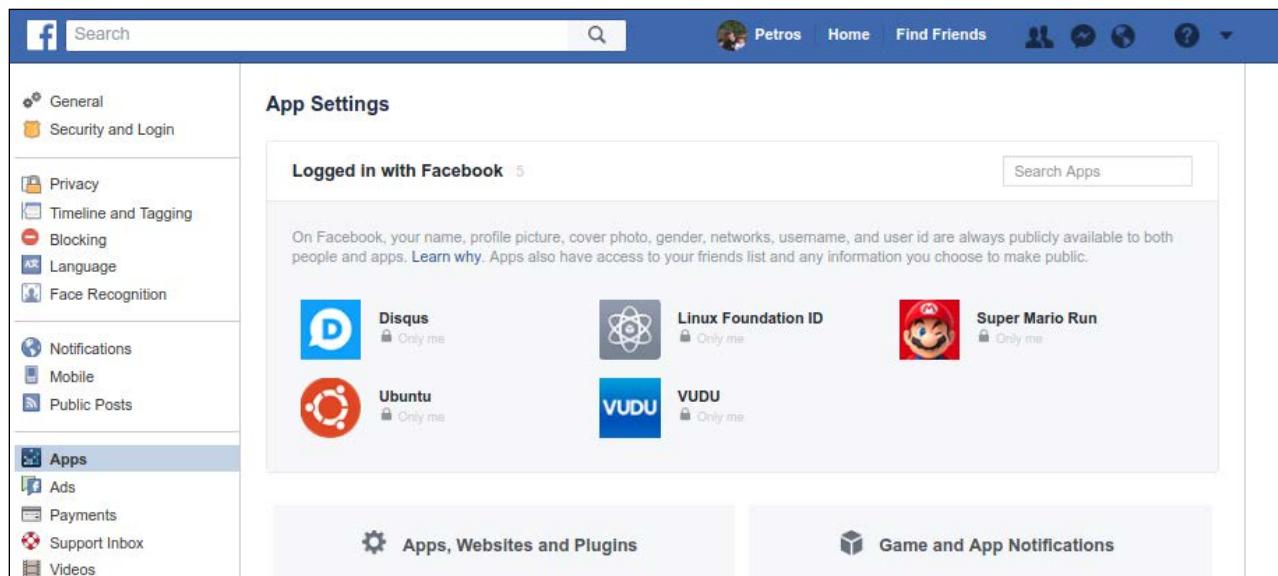


Figure 1. The Facebook Application Settings Page

Foundation and Super Mario Run. These applications have direct access to my account information, my timeline, my contacts and more.

Applications such as these don't automatically authenticate you with your social-media account. You need to authorize the application specifically to authenticate you by that account. And even though you may have agreed to it at some point, be sure to visit these sections of your assorted accounts routinely and review what's there.

So, the next time you are redirected from that social-media site and decide to take that personality quiz or survey to determine what kind of person you are attracted to or even what you would look like as the opposite sex, think twice about logging in using your account. By doing so, you're essentially agreeing to give that platform access to everything stored in your account.

This is essentially how firms like Cambridge Analytica obtain user information. You never can be too sure of how that information will be used or misused.

Tracking-Based Advertisements

The quickest way for search engines and social-media platforms to make an easy dollar is to track your interests and specifically target related advertisements to you. How often do you search for something on Google or peruse through Facebook or Twitter feeds and find advertisements of a product or service you were looking into the other day? These platforms keep track of your location, your search history and your general activities. Sounds scary, huh? In some cases, your network of friends even may see the products or services from your searches.

To avoid such targeting, you need to rethink how you search the internet. Instead of Google, opt for something like DuckDuckGo. With online stores like Amazon, keep your wish lists and registries private or share them with a select few individuals. In the case of social media, you probably should update your Advertisement Preferences.

In some cases, you can completely disable targeted advertisements based on your search or activity history. You also can limit what your network of peers can see.

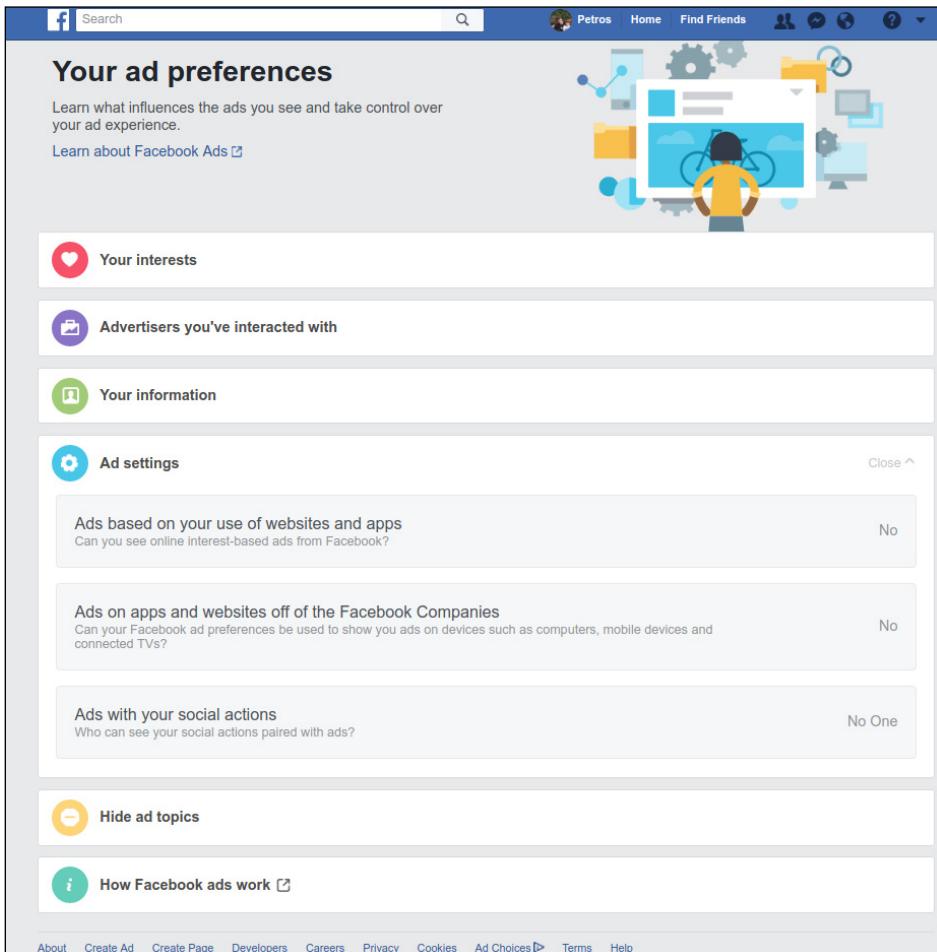


Figure 2. The Facebook Advertisement Preferences Page

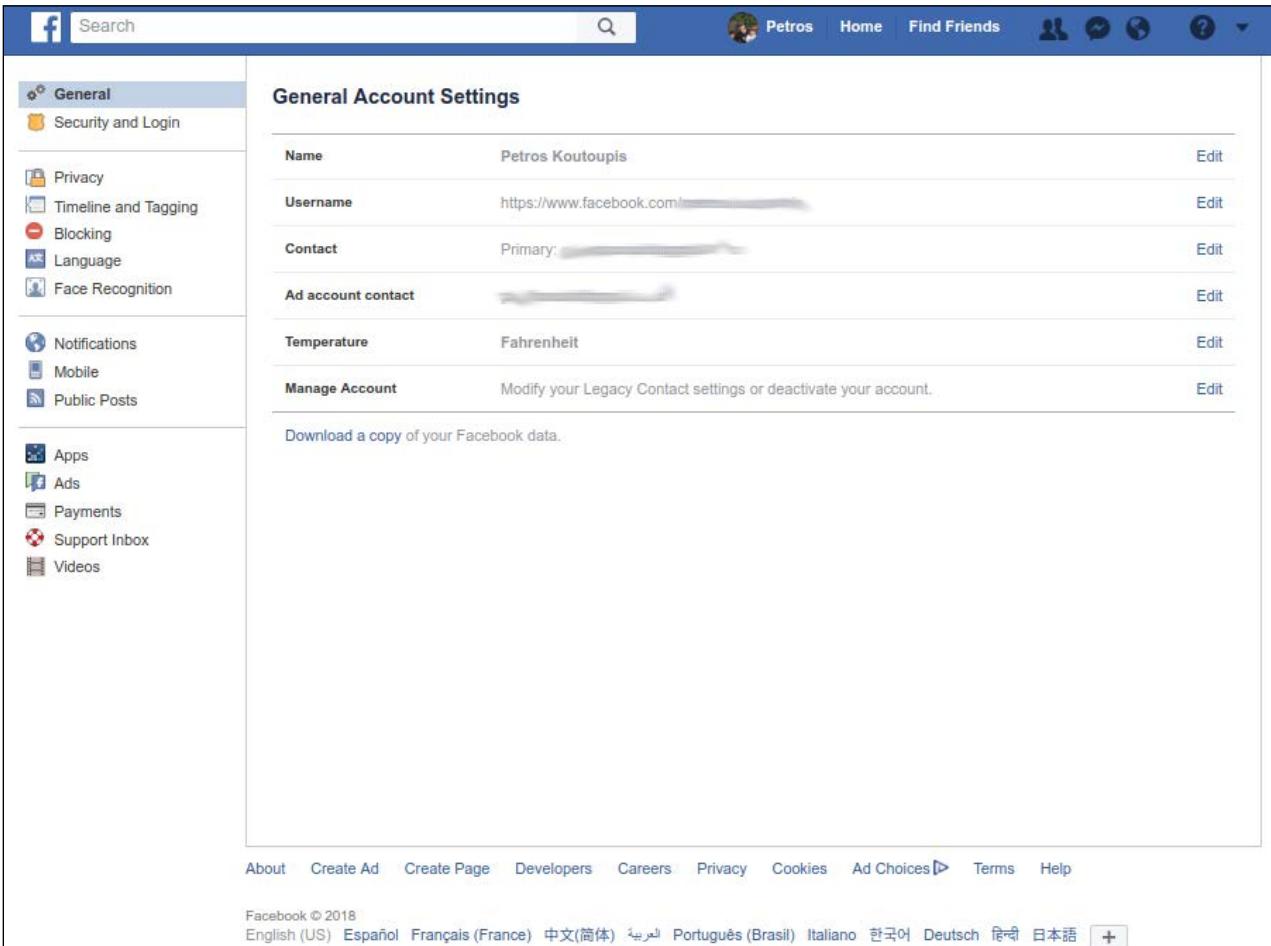
Understanding What's at Risk

People often don't think about privacy matters until something affects them directly. A good way to understand what personal data you risk is to request that same data from a service provider. It is this exact data that the service provider may sell to third parties or even allow external applications to access.

You can request this data from Facebook, via the General Account Settings page. At the very bottom of your general account details, there's a link appropriately labeled "Download a copy of your Facebook data".

It takes a few minutes to collect everything and compress it into a single .zip file, but when complete, you'll receive an email with a direct link to retrieve this archive.

DEEP DIVE



The screenshot shows the Facebook General Account Settings page. On the left, there's a sidebar with various settings categories like General, Privacy, Notifications, and Apps. The main content area is titled "General Account Settings" and contains fields for Name (Petros Koutoupis), Username (https://www.facebook.com/...), Contact (Primary: ...), Ad account contact (...), Temperature (Fahrenheit), and Manage Account (Modify your Legacy Contact settings or deactivate your account). Below these fields is a link to download a copy of your Facebook data. At the bottom of the page, there are links for About, Create Ad, Create Page, Developers, Careers, Privacy, Cookies, Ad Choices, Terms, Help, and language options (English (US), Español, Français (France), 中文(简体), العربية, Português (Brasil), Italiano, 한국어, Deutsch, हिन्दी, 日本語, +).

Figure 3. The Facebook General Account Settings Page

When extracted, you'll see a nicely organized collection of all your assorted activities:

```
$ ls -l
total 24
drwxrwxr-x 2 petros petros 4096 Mar 24 07:11 html
-rw-r--r-- 1 petros petros 6403 Mar 24 07:01 index.htm
drwxrwxr-x 8 petros petros 4096 Mar 24 07:11 messages
drwxrwxr-x 6 petros petros 4096 Mar 24 07:11 photos
drwxrwxr-x 3 petros petros 4096 Mar 24 07:11 videos
```

DEEP DIVE

Open the file index.html at the root of the directory (Figure 4).

Everything, and I mean *everything*, you have done with this account is stored and never deleted. All of your Friends history, including blocked and removed individuals is preserved. Every photograph and video uploaded and every private message sent via Messenger is forever retained. Every advertisement you clicked (and in turn every advertiser that has your contact information) and possibly even more is recorded. I don't even know 90% of the advertisers appearing on my list, nor have I ever agreed to share information with them. I also can tell that a lot of them aren't even from this country. For



Petros Koutoupis

Profile	https://www.facebook.com/██████████
Registration Date	Monday, July 6, 2015 at 11:10am CDT
Emails	██████████@facebook.com(Previous Email)
Address	
Neighborhood	
Phones	██████████
Photos	Birthday May 29, █████
Videos	Gender Male
Friends	Other names Πέτρος Κουτούπης (Birth Name) Peter (Nickname)
Messages	
Pokes	
Events	Current City Chicago, Illinois
Security	Relationship Status Married to █████ Koutoupis since February 13, 2009
Ads	Family
Applications	
Education	 Electronic Engineering Technology
Workplace	Linux Journal January 1, 2018 - contributing editor I write stuff and it gets published. IBM August 3, 2015 - Senior Platform Architect IBM Cloud Object Storage (Cleversafe) Key responsibilities include: * Performing magic tricks. * Telling bad jokes.

Figure 4.
The Facebook
Archive
Profile Page

instance, why does a German division of eBay have my information through Facebook when I use the United States version of eBay and always have since at least 1999? Why does Sally Beauty care about who I am? Last time I checked, I don't buy anything through that company (I am not real big into cosmetics or hair-care products).

It's even been reported that when Facebook is installed on your mobile device, it can and will log all details pertaining to your phone calls and text messages (names, phone numbers, duration of call and so on).

Mobile Devices

I've already spent a great deal of time focusing on social media, but data privacy doesn't end there. Another area of concern is around mobile computing. It doesn't matter which mobile operating system you are running (Android or iOS) or which mobile hardware you are using (Samsung, Huawei, Apple and so on). The end result is the same. Several mobile applications, when installed, are allowed unrestricted access to more data than necessary.

With this in mind, I went to the Google Play store and looked up the popular Snapchat app. Figure 5 shows a summary of everything Snapchat needed to access.

A few of these categories make sense, but some of the others leave you wondering. For instance, why does Snapchat need to know about my "Device ID & call information" or my WiFi and Bluetooth connection information? Why does it need to access my SMS text messages? What do applications like Snapchat do with this collected data? Do they find ways to target specific products and features based on your history or do they sell it to third parties?

Mobile devices often come preinstalled with software or preconfigured to store or synchronize your personal data with a back-end cloud storage service. This software may be provided by your cellular service provider, the hardware product manufacturer or even by the operating system developer. Review those settings and disable anything that does not meet your standards. Even if you rely on Google to synchronize your photographs and videos to your Google Drive or Photos account, restrict which folders

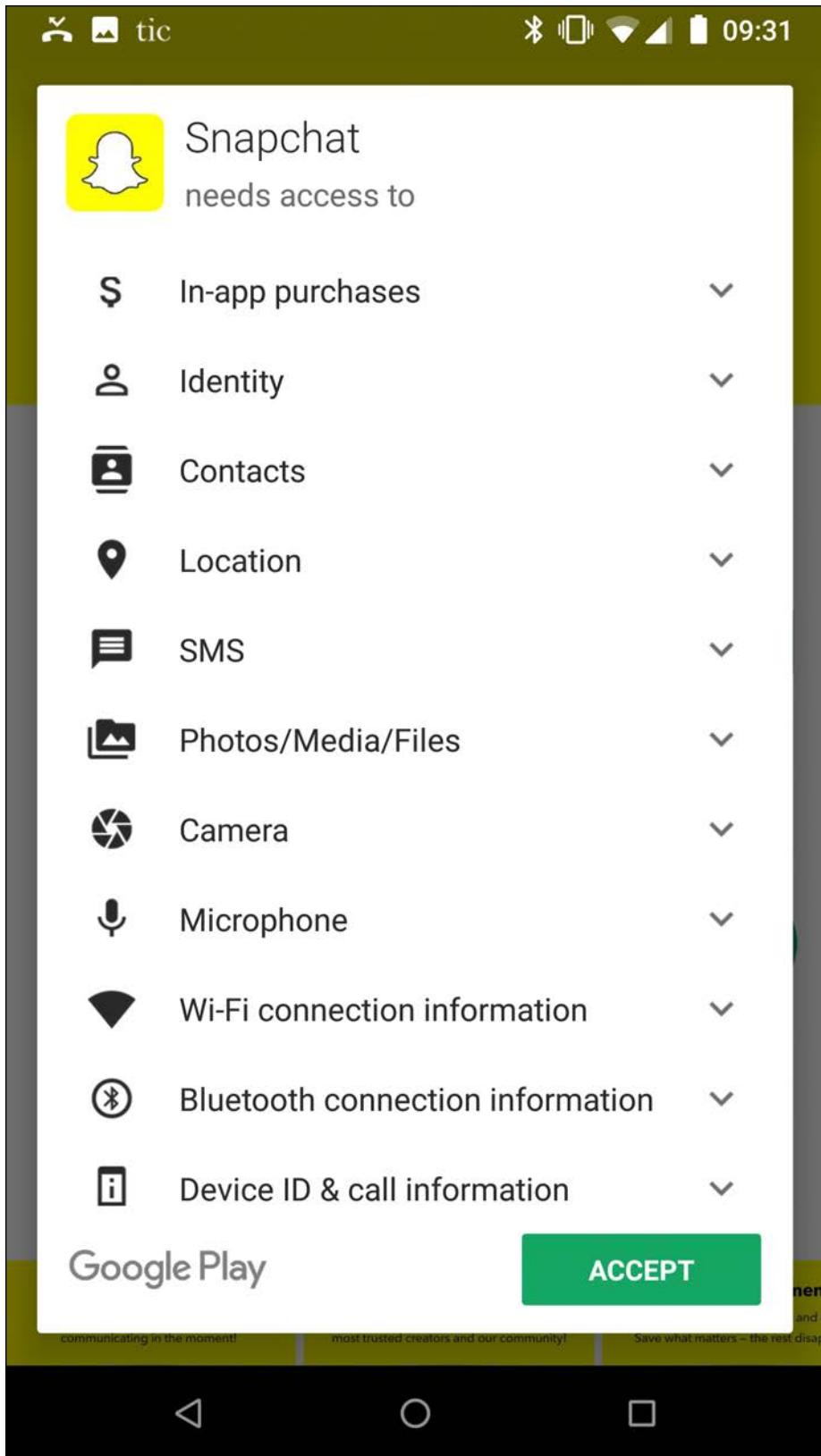


Figure 5. Access Requirements for a Popular Mobile Application

or subdirectories are synchronized.

Want to take this a step further? Think twice about enabling fingerprint authentication. Hide all notifications on your lock screen. Disable location tracking activity. Encrypt your phone.

What about Local Privacy?

There is more. You also need to consider local privacy. There is a certain recipe you always should follow.

Passwords Use good passwords. Do not enable auto-login, and also disable root logins for SSH. Enable your screen lock when the system is idle for a set time and when the screensaver takes over.

Encryption Encrypting your home directory or even the entire disk drive limits the possibility that unauthorized individuals will gain access to your personal data while physically handing the device. Most modern Linux distributions offer the option to enable this feature during the installation process. It's still possible to encrypt an existing system, but before you decide to undertake this often risky endeavor, make sure you first back up anything that's considered important.

Applications Review your installed applications. Keep things to a minimum. If you don't use it, uninstall it. This can help you in at least three ways:

1. It will reduce overall clutter and free up storage space on your local hard drive.
2. You'll be less at risk of hosting a piece of software with bugs or security vulnerabilities, reducing the potential of your system being compromised in some form or another.
3. There is less of a chance that the same software application is collecting data that it shouldn't be collecting in the first place.

System Updates Keep your operating system updated at all times. Major Linux

distributions are constantly pushing updates to existing packages that address both software defects and security vulnerabilities.

HTTP vs. HTTPS Establish secure connections when browsing the internet. Pay attention when transferring data. Is it done over HTTP or HTTPS? (The latter is the secured method.) The last thing you need is for your login credentials to be transferred as plain text to an unsecured website. That's why so many service providers are securing your platforms for HTTPS, where all requests made are encrypted.

Web Browsers Use the right web browser. Too many web browsers are less concerned with your privacy and more concerned with your experience. Take the time to review your browsing requirements, and if you need to run in private or “incognito” mode or just adopt a new browser more focused on privacy, take the proper steps to that.

Add-ons While on the topic of web browsers, review the list of whatever add-ons are installed and configured. If an add-on is not in use or sounds suspicious, it may be safe to disable or remove it completely.

Script Blockers Script blockers (NoScript, AdBlock and so on) can help by preventing scripts embedded on websites from tracking you. A bit of warning: these same programs can and may even ruin your experiences with a large number of websites visited.

Port Security Review and refine your firewall rules. Make sure you drop anything coming in that shouldn't be intruding in the first place. This may even be a perfect opportunity to discover what local services are listening on external ports (via `netstat -lt`). If you find that these services aren't necessary, turn them off.

Securing Connections

Every device connecting over a larger network is associated with a unique address. The device obtains this unique address from the networking router or gateway to which it connects. This address commonly is referred to as that device's IP (internet protocol) address. This IP address is visible to any website and any server you visit. You'll always be identified by this address while using this device.

It's through this same address that you'll find advertisements posted on various websites and in various search engines rendered in the native language of the country in which you live. Even if you navigate to a foreign website, this method of targeting ensures that the advertisements posted on that site cater to your current location.

Relying on IP addresses also allows some websites or services to restrict access to visitors from specific countries. The specific range of the address will point to your exact country on the world map.

Virtual Private Network

An easy way to avoid this method of tracking is to rely on the use of Virtual Private Networks (VPNs). It is impossible to hide your IP address directly. You wouldn't be able to access the internet without it. You can, however, pretend you're using a different IP address, and this is where the VPN helps.

A VPN extends a private network across a public network by enabling its users to send/receive data across the public network as if their device were connected directly to the private network. There exists hundreds of VPN providers worldwide. Choosing the right one can be challenging, but providers offer their own set of features and limitations, which should help shrink that list of potential candidates.

Let's say you don't want to go with a VPN provider but instead want to configure your own VPN server. Maybe that VPN server is located somewhere in a data center and nowhere near your personal computing device. For this example, I'm using Ubuntu Server 16.04 to install OpenVPN and configure it as a server. Again, this server can be hosted from anywhere: in a virtual machine in another state or province or even in the cloud (such as AWS EC2). If you do host it on a cloud instance, be sure you set that instance's security group to accept incoming/outgoing UDP packets on port 1194 (your VPN port).

The Server Log in to your already running server and make sure that all local packages are updated:

```
$ sudo apt-get update && sudo apt-get upgrade
```

Install the **openvpn** and **easy-rsa** packages:

```
$ sudo apt-get install openvpn easy-rsa
```

Create a directory to set up your Certificate Authority (CA) certificates. OpenVPN will use these certificates to encrypt traffic between the server and client. After you create the directory, change into it:

```
$ make-cadir ~/openvpn-ca  
$ cd ~/openvpn-ca/
```

Open the vars file for editing and locate the section that contains the following parameters:

```
# These are the default values for fields  
# which will be placed in the certificate.  
# Don't leave any of these fields blank.  
export KEY_COUNTRY="US"  
export KEY_PROVINCE="CA"  
export KEY_CITY="SanFrancisco"  
export KEY_ORG="Fort-Funston"  
export KEY_EMAIL="me@myhost.mydomain"  
export KEY_OU="MyOrganizationalUnit"  
  
# X509 Subject Field  
export KEY_NAME="EasyRSA"
```

Modify the fields accordingly, and for the **KEY_NAME**, let's define something a bit more generic like “server” Here's an example:

```
# These are the default values for fields  
# which will be placed in the certificate.  
# Don't leave any of these fields blank.
```

```
export KEY_COUNTRY="US"
export KEY_PROVINCE="IL"
export KEY_CITY="Chicago"
export KEY_ORG="Linux Journal"
export KEY_EMAIL="localadmin@example.com"
export KEY_OU="Community"

# X509 Subject Field
export KEY_NAME="server"
```

Export the variables:

```
$ source vars
NOTE: If you run ./clean-all, I will be doing a rm -rf
      ↵on /home/ubuntu/openvpn-ca/keys
```

Clean your environment of old keys:

```
$ ./clean-all
```

Build a new private root key, choosing the default options for every field:

```
$ ./build-ca
```

Next build a new private server key, also choosing the default options for every field (when prompted to input a challenge password, you won't input anything for this current example):

```
$ ./build-key-server server
```

Toward the end, you'll be asked to sign and commit the certificate. Type "y" for yes:

```
Certificate is to be certified until Mar 29 22:27:51 2028
```

```
↳GMT (3650 days)
Sign the certificate? [y/n]:y
```

```
1 out of 1 certificate requests certified, commit? [y/n]y
Write out database with 1 new entries
Data Base Updated
```

You'll also generate strong Diffie-Hellman keys:

```
$ ./build-dh
```

To help strengthen the server's TLS integrity verification, generate an HMAC signature:

```
$ openvpn --genkey --secret keys/ta.key
```

So, you've finished generating all the appropriate server keys, but now you need to generate a client key for your personal machine to connect to the server. To simplify this task, create this client key from the same server where you generated the server's keys.

If you're not in there already, change into the same ~/openvpn-ca directory and source the same **vars** file from earlier:

```
$ cd ~/openvpn-ca
$ source vars
```

Generate the client certificate and key pair, again choosing the default options, and for the purpose of this example, avoid setting a challenge password:

```
$ ./build-key client-example
```

As with the server certificate/key, again, toward the end, you'll be asked to sign and commit the certificate. Type "y" for yes:

```
Certificate is to be certified until Mar 29 22:32:37 2028
```

```
↳GMT (3650 days)
```

```
Sign the certificate? [y/n]:y
```

```
1 out of 1 certificate requests certified, commit? [y/n]y
```

```
Write out database with 1 new entries
```

```
Data Base Updated
```

Change into the keys subdirectory and copy the keys you generated earlier over to the /etc/openvpn directory:

```
$ cd keys/  
$ sudo cp ca.crt server.crt server.key ta.key  
↳dh2048.pem /etc/openvpn/
```

Extract the OpenVPN sample server configuration file to the /etc/openvpn directory:

```
$ gunzip -c /usr/share/doc/openvpn/examples/sample-config-files/  
↳server.conf.gz |sudo tee /etc/openvpn/server.conf
```

Let's use this template as a starting point and apply whatever required modifications are necessary to run the VPN server application. Using an editor, open the /etc/openvpn/server.conf file. The fields you're most concerned about are listed below:

```
;tls-auth ta.key 0 # This file is secret
```

```
[ ... ]
```

```
;cipher BF-CBC      # Blowfish (default)  
;cipher AES-128-CBC # AES  
;cipher DES-EDE3-CBC # Triple-DES
```

```
[ ... ]
```

```
;user nobody
;group nogroup
```

Uncomment and add the following lines:

```
tls-auth ta.key 0 # This file is secret
key-direction 0
```

```
[ ... ]
```

```
;cipher BF-CBC      # Blowfish (default)
cipher AES-128-CBC # AES
auth SHA256
;cipher DES-EDE3-CBC # Triple-DES
```

```
[ ... ]
```

```
user nobody
group nogroup
```

You'll need to enable IPv4 packet forwarding via `sysctl`. Uncomment the field `net.ipv4.ip_forward=1` in /etc/sysctl.conf and reload the configuration file:

```
$ sudo sysctl -p
net.ipv4.ip_forward = 1
```

If you're running a firewall, UDP on port 1194 will need to be open at least to the public IP address of the client machine. Once you do this, start the server application:

```
$ sudo systemctl start openvpn@server
```

And if you wish, configure it to start automatically every time the system reboots:

```
$ sudo systemctl enable openvpn@server
```

Finally, create the client configuration file. This will be the file the client will use every time it needs to connect to the VPN server machine. To do this, create a staging directory, set its permissions accordingly and copy a client template file into it:

```
$ mkdir -p ~/client-configs/files
$ chmod 700 ~/client-configs/files/
$ cp /usr/share/doc/openvpn/examples/sample-config-files/client.conf
  ~/client-configs/base.conf
```

Open the ~/client-configs/base.conf file in an editor and locate the following areas:

```
remote my-server-1 1194
;remote my-server-2 1194
```

```
[ ... ]
```

```
ca ca.crt
cert client.crt
key client.key
```

```
[ ... ]
```

```
;cipher x
```

The variables should look something like this:

```
remote <public IP of server> 1194
;remote my-server-2 1194
```

```
[ ... ]
```

```
#ca ca.crt
#cert client.crt
#key client.key

[ ... ]

cipher AES-128-CBC
auth SHA256

key-direction 1

# script-security 2
# up /etc/openvpn/update-resolv-conf
# down /etc/openvpn/update-resolv-conf
```

The remote server IP will need to be adjusted to reflect the public IP address of the VPN server. Be sure to adjust the cipher while also adding the **auth** and the **key-direction** variables. Also append the commented **script-security** and **update-resolv-conf** lines. Now, generate the OVPN file:

```
cat ~/client-configs/base.conf \
<(echo -e '<ca>' ) \
~/openvpn-ca/keys/ca.crt \
<(echo -e '</ca>\n<cert>' ) \
~/openvpn-ca/keys/client-example.crt \
<(echo -e '</cert>\n<key>' ) \
~/openvpn-ca/keys/client-example.key \
<(echo -e '</key>\n<tls-auth>' ) \
~/openvpn-ca/keys/ta.key \
<(echo -e '</tls-auth>' ) \
> ~/client-configs/files/client-example.ovpn
```

You should see the newly created file located in the `~/client-configs/files` subdirectory:

```
$ ls ~/client-configs/files/  
client-example.ovpn
```

The Client Copy the OVPN file to the client machine (your personal computing device). In the example below, I'm connected to my client machine and using SCP, transferring the file to my home directory:

```
$ scp petros@openvpn-server:~/client-configs/files/  
client-example.ovpn ~/  
client-example.ovpn          100%   13KB  12.9KB/s  00:00
```

Install the OpenVPN package:

```
$ sudo apt-get install openvpn
```

If the /etc/openvpn/update-resolv-conf file exists, open your OVPN file (currently in your home directory) and uncomment the following lines:

```
script-security 2  
up /etc/openvpn/update-resolv-conf  
down /etc/openvpn/update-resolv-conf
```

Connect to the VPN server by pointing to the client configuration file:

```
$ sudo openvpn --config client-example.ovpn
```

While you're connected, you'll be browsing the internet with the public IP address used by the VPN server and one that was not assigned by your Internet Service Provider (ISP).

Summary

How does it feel to have your locations, purchasing habits, preferred reading content, search history (including health and illness), political views and more shared with an unknown number of recipients across this mysterious thing we call the internet? It

DEEP DIVE

probably doesn't feel very comforting. This might be information we typically would not want our closest family and friends to know, so why would we want strangers to know it instead? It is far too easy to be complacent and allow such personal data mining to take place. Retaining true anonymity while also enjoying the experiences of the modern web is definitely a challenge. Although, it isn't impossible.

I have highlighted some appropriate steps you can take to reduce your digital footprint on the internet, but this is far from a complete list. Don't stop here. Do your due diligence and give yourself the reassurance that when you're using modern technology, you're accessing it in a safe and secure manner. ■



Petros Koutoupis, *LJ* Contributing Editor, is currently a senior platform architect at IBM for its Cloud Object Storage division (formerly Cleversafe). He is also the creator and maintainer of the RapidDisk Project. Petros has worked in the data storage industry for well over a decade and has helped pioneer the many technologies unleashed in the wild today.

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

Privacy Plugins

Protect yourself from privacy-defeating ad trackers and malicious JavaScript with these privacy-protecting plugins.

By Kyle Rankin

Although your phone is probably the biggest threat to your privacy, your web browser is a close second. In the interest of providing you targeted ads, the web is littered with technology that attempts to track each site you go to via a combination of cookies and JavaScript snippets. These trackers aren't just a privacy threat, they are also a security threat. Because of how ubiquitous these ad networks are, attackers have figured out ways to infiltrate some of them and make them serve up even more malicious code.

The good news is that a series of privacy plugins work well with Firefox under Linux. They show up as part of the standard list of approved add-ons and will help protect you against these kinds of threats. Many different privacy plugins exist, but instead of covering them all, in this article, I highlight some of my personal favorites—the ones I install on all of my browsers. Although I discuss these plugins in the context of Firefox, many of them also are available for other Linux browsers. Because all of these plugins are standard Firefox add-ons, you can install them through your regular Firefox add-on search panel.

Privacy Badger

The EFF has done a lot of work recently to improve privacy and security for average users online, and its **Privacy Badger** plugin is the first one I want to cover here. The idea behind Privacy Badger is to apply some of the tools from different plugins like AdBlock Plus, Ghostery and others that inspect third-party JavaScript on a page. When that JavaScript comes from a known tracking network or attempts to install a tracking cookie on your computer, Privacy Badger steps in and blocks it.

If so many other plugins do something similar, why re-invent the wheel with Privacy



Figure 1. Privacy Badger (By Electronic Frontier Foundation <https://www.eff.org/sites/all/themes/badger/badger-stroke.png>, CC BY 3.0 us, <https://commons.wikimedia.org/w/index.php?curid=42161849>)

Badger? Well, the downside to many of the other tools is that they often require user intervention to tweak and tune. Although it's great for people who want to spend their time doing that, average users probably rather would spend their time actually browsing the web. Privacy Badger has focused on providing similar protection without requiring any special tweaking or tuning. As you browse the web, it keeps track of these different sites, and by observing their behavior, decides whether they are tracking you.

So once you install Privacy Badger, how do you know it's working? For starters, after the plugin is installed, you'll see a new icon of a cartoon badger head in your Firefox

task bar (Figure 1). Above that icon is a green, yellow or red box that contains a number. If Privacy Badger didn't have to block anything on a site, you'll see a nice green 0 in the box. On the other hand, if you see a yellow or red box, Privacy Badger flagged parts of the site, and if you click the icon, you'll see a list of the sites along with a rating. If a site is flagged as yellow, it means Privacy Badger thinks it might be trying to track you, but its cookies seem necessary for the functioning of the site so it has allowed them. On the other hand, if something is red, it means Privacy Badger has blocked it completely.

HTTPS Everywhere

Ad trackers are obvious threats to your privacy, but there also are threats that might be less obvious. One big way you can be tracked is through your use of unencrypted traffic over HTTP. When you visit a site with HTTP, someone sitting between you and the site can capture a copy of your traffic and forward it on to the site. This is a particular risk if you are using a public network, like at a coffee shop or a conference. The solution is to make sure you visit sites only over HTTPS. By using HTTPS, you not only encrypt all of the traffic between your browser and the site, the site also will authenticate itself to you with a certificate so you can be sure you are talking directly to it and not to an attacker in the middle.

Because browsers default to HTTP, practically speaking, it can be difficult to add “https://” in front of every URL you type, especially when you consider how rarely you might even enter a URL in your browser these days. Plus, many sites default to HTTP even if they support HTTPS. Fortunately, the EFF saves the day again with its **HTTPS Everywhere** plugin. The HTTPS Everywhere plugin changes the default behavior in the browser so that when you visit a site, it attempts to connect to HTTPS first. Only after an HTTPS connection fails will it fall back to HTTP.

In addition to favoring HTTPS over HTTP, you can click the S icon in the blue box on your taskbar to see extra HTTPS Everywhere settings. In particular, you can check a box (which is off by default) that will go a step further and block all unencrypted traffic to a site (Figure 2). You also can set preferences that apply only to particular sites.

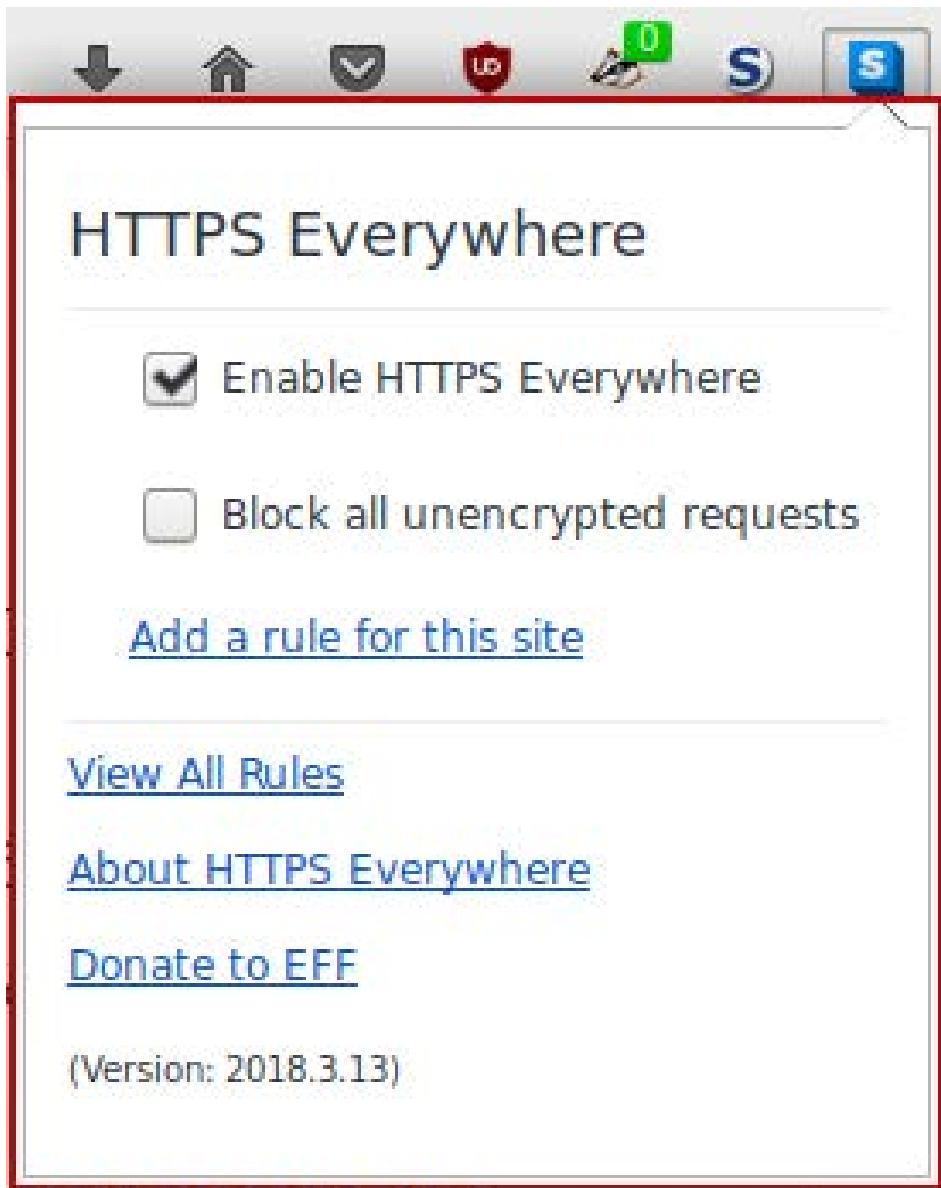


Figure 2. HTTPS Everywhere Settings

uBlock Origin

I used the AdBlock Plus plugin for many years to block ads on sites. I didn't use it only to block ads themselves but also for my personal privacy and security because along with the ads are privacy-defeating tracking and often malicious JavaScript as well. Advertisers were strongly opposed to ad blocking as you might imagine and in particular focused their attention on AdBlock Plus due to its popularity. Eventually AdBlock Plus announced a program where it would default to allowing certain approved ad networks through. For

me, that defeated the whole purpose, so I looked for an alternative and uBlock Origin seemed to fit the bill.

The uBlock Origin plugin works much like AdBlock Plus. Once you install the plugin, it detects and blocks ads from appearing on sites you visit. If you click its icon, it also will show you what it's blocked on the current page and let you allow ads through temporarily just for the current page or for the site overall. It's simple, works well and doesn't require much intervention, so it's become my preferred ad blocker for Firefox.

NoScript

All of the previously mentioned plugins are things I'd recommend just about anyone use to increase their privacy. This last plugin is a bit different though. I'm old enough to remember when JavaScript was considered a bad thing, and many users disabled it entirely on their browsers (and actually still were able to browse the web). These days, you can't get away with disabling JavaScript entirely—almost all of the web will break. That said, one of the biggest threats to both privacy and security is via third-party JavaScript, so if you want your web browsing to be very secure, you have to figure out a way to block all but the most essential JavaScript from loading.

NoScript is a plugin that lets you select, site-by-site, which JavaScript to run when you load a page. By default, NoScript blocks all JavaScript. When you visit a page that provides JavaScript, NoScript's icon changes to a red circle with a slash through it, and when you click the icon, it shows you the URLs for different JavaScript it has blocked. You then can decide on a per-domain basis which JavaScript to load either temporarily or permanently (Figure 3). Once you make your selections, the page will reload, and hopefully, you'll see more of the site.

When you use NoScript, the web becomes very interesting, because at first, a site might have only JavaScript from itself for you to load. Once you allow that and refresh, you might notice a huge number of new sites from all over the web that the first JavaScript now wants to load. It's because of this that using NoScript sometimes can be annoying. Often to get a site to load, you have to become a web detective and track down which of those third-party sites are necessary to load (such as cloudfront.net AWS content

DEEP DIVE

The screenshot shows a web browser window for www.latimes.com. The page displays several articles and a sidebar with a list of blocked scripts. The main content includes an article about the Supreme Court weighing partisan gerrymandering, another about police killings in Sacramento, and a photo of Kim Jong Un and Xi Jinping. A sidebar on the right lists numerous blocked scripts from various domains like amazon-adsystem.com, krxd.net, userzoom.com, indexww.com, newsinc.com, ensighten.com, chartbeat.com, google-analytics.com, googletagservices.com, go-mpulse.net, tribdss.com, and Facebook. At the bottom, a status bar indicates "Scripts Partially Allowed, 1/13 (latimes.com) | <SCRIPT>: 40 | <OBJECT>: 0".

Forbid latimes.com

- Temporarily allow amazon-adsystem.com
- Allow amazon-adsystem.com
- Temporarily allow krxd.net
- Allow krxd.net
- Temporarily allow userzoom.com
- Allow userzoom.com
- Temporarily allow indexww.com
- Allow indexww.com
- Temporarily allow newsinc.com
- Allow newsinc.com
- Temporarily allow ensighten.com
- Allow ensighten.com
- Temporarily allow chartbeat.com
- Allow chartbeat.com
- Temporarily allow google-analytics.com
- Allow google-analytics.com
- Temporarily allow googletagservices.com
- Allow googletagservices.com
- Temporarily allow go-mpulse.net
- Allow go-mpulse.net
- Temporarily allow tribdss.com
- Allow tribdss.com
- Untrusted
- Recently blocked sites
- Make page permissions permanent
- Temporarily allow all this page
- Allow all this page
- Revoke Temporary Permissions
- Allow Scripts Globally (dangerous)

Facebook unveils new ways to find privacy shortcuts

54m

Options... About NoScript 5.1.8.4... <http://www.latimes.com/> [5/5] Top

Figure 3. NoScript Output on the *Los Angeles Times* Website

or cloudflare.com CDN content) and which sites are loading ads (their domains often contain words like “metrics” or “monitor” in them). It’s this detective work that ends up being draining, and you become tempted just to allow all the JavaScript on a site (for

which NoScript provides via a handy button).

Because of all of the extra work NoScript requires to work well, and how frustrating it can be sometimes to load new JavaScript continually, only to have yet more to load before a site will work, I don't recommend every user install and use NoScript. For users like me who are particularly concerned about security and want more control over JavaScript though, it's invaluable.

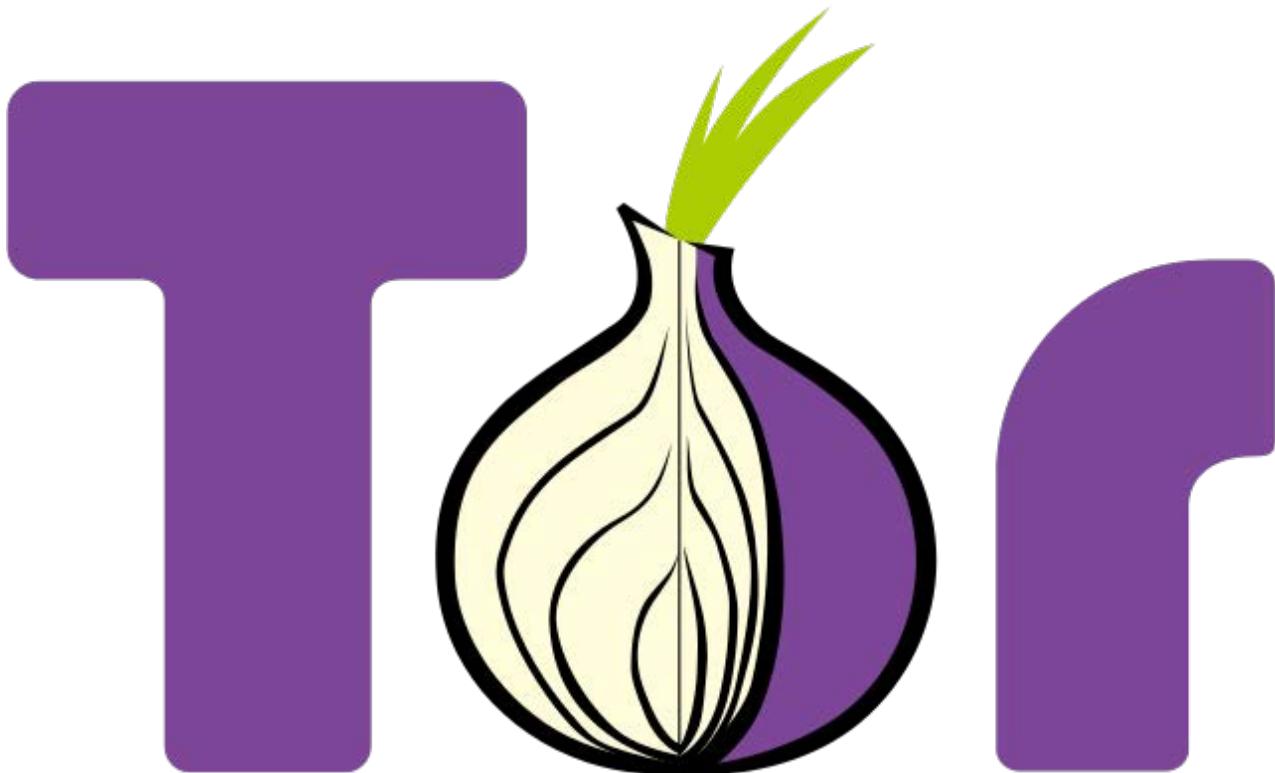
Conclusion

Although many different privacy plugins exist, these four are the ones I use on a daily basis and install across my machines. I hope you find them useful too. These privacy plugins not only protect your privacy, they also can help protect you against malicious sites, and by blocking sluggish third-party sites from loading, they also often can make sites load much faster. ■



Kyle Rankin is a Tech Editor and columnist at *Linux Journal* and the Chief Security Officer at Purism. He is the author of *Linux Hardening in Hostile Networks*, *DevOps Troubleshooting*, *The Official Ubuntu Server Book*, *Knoppix Hacks*, *Knoppix Pocket Reference*, *Linux Multimedia Hacks* and *Ubuntu Hacks*, and also a contributor to a number of other O'Reilly books. Rankin speaks frequently on security and open-source software including at BsidesLV, O'Reilly Security Conference, OSCON, SCALE, CactusCon, Linux World Expo and Penguinon. You can follow him at @kylerankin.

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



Tor Hidden Services

Why should clients get all the privacy? Give your servers some privacy too!

By *Kyle Rankin*

When people write privacy guides, for the most part they are written from the perspective of the client. Whether you are using HTTPS, blocking tracking cookies or going so far as to browse the internet over Tor, those privacy guides focus on helping end users protect themselves from the potentially malicious and spying web. Since many people who read *Linux Journal* sit on the other side of that equation—they run the servers that host those privacy-defeating services—system administrators also should step up and do their part to help user privacy. Although part of that just means making

sure your services support TLS, in this article, I describe how to go one step further and make it possible for your users to use your services completely anonymously via Tor hidden services.

How It Works

I'm not going to dive into the details of how Tor itself works so you can use the web anonymously—for those details, check out <https://tor.eff.org>. Tor hidden services work within the Tor network and allow you to register an internal, Tor-only service that gets its own .onion hostname. When visitors connect to the Tor network, Tor resolves those .onion addresses and directs you to the anonymous service sitting behind that name. Unlike with other services though, hidden services provide two-way anonymity. The server doesn't know the IP of the client, like with any service you access over Tor, but the client also doesn't know the IP of the server. This provides the ultimate in privacy since it's being protected on both sides.

Warnings and Planning

As with setting up a Tor node itself, some planning is involved if you want to set up a Tor hidden service so you don't defeat Tor's anonymity via some operational mistake. There are a lot of rules both from an operational and security standpoint, so I recommend you read this [excellent guide](#) to find the latest best practices all in one place.

Without diving into all of those steps, I do want to list a few general-purpose guidelines here. First, you'll want to make sure that whatever service you are hosting is listening only on localhost (127.0.0.1) and isn't viewable via the regular internet. Otherwise, someone may be able to correlate your hidden service with the public one. Next, go through whatever service you are running and try to scrub specific identifying information from it. That means if you are hosting a web service, modify your web server so it doesn't report its software type or version, and if you are running a dynamic site, make sure whatever web applications you use don't report their versions either.

Some services need to talk to the internet to resolve DNS names or download other resources. It's important that you configure your service so that all of those external requests route over Tor. You can do this with iptables rules that force all of your traffic

through a local Tor proxy like you would for a client, or if your service has SOCKS proxy support, you can configure it to use the built-in Tor SOCKS proxy.

Finally, although you can run a hidden service and a relay node from the same host, it's considered a best practice to keep them separated so you can't correlate a hidden service with a particular relay node. Plus, this means you can just run a hidden service without worrying about any risks involved in running a relay node.

How to Configure a Hidden Service

The first step in configuring a hidden service is to install Tor. Tor should be packaged for most major distributions, so you can just use your package manager to pull down the latest version. If you like to do things the hard way, or want to make absolutely sure to get the latest version, you also could sidestep your package manager and build Tor from sources on <https://torproject.org>.

You will configure your hidden services the same way whether you use a Red Hat or Debian-based distribution via the /etc/tor/torrc configuration file. As you'll see, the configuration is nice and simple.

Hidden HTTP Service

For starters, let's assume you want to host a web service. Make sure that you configure your web server so that it's only listening on localhost (127.0.0.1), so as not to leak data that may make it easier to correlate your hidden service with a public service. Next, add the following two lines to your /etc/tor/torrc:

```
HiddenServiceDir /var/lib/tor/hidden_service/http
HiddenServicePort 80 127.0.0.1:80
```

Now restart the Tor service (`sudo systemctl restart tor`), and your service will be ready. That wasn't so bad, right? The `HiddenServiceDir` option will tell you where Tor should store information about this service (including its .onion address). The `HiddenServicePort` option tells Tor on which port it should listen for the hidden service (80 in this case) and to which address to forward that traffic

(127.0.0.1:80 for this example).

Once the service has started, you will notice two different files under /var/lib/tor/hidden_service/http: a “private_key” and a “hostname” file. The private_key file is used to authenticate this particular hidden service. It’s important that you protect this file, because anyone who has a copy of it can impersonate your service. The hostname file contains the name for your hidden service:

```
$ sudo cat /var/lib/tor/hidden_service/http/hostname  
o9asojd8aymqqtoa.onion
```

This .onion hostname is what visitors to your service would enter into their Tor browser to visit you. Note that you’re hosting HTTP and not HTTPS here. HTTPS requires you to get a valid certificate for f27sojd8aymqqtwa.onion, which wouldn’t be possible. In any case, Tor takes care of authenticating and encrypting your communication to the site, and you can think of that private_key file as acting kind of like a private TLS key.

Adding Additional Services

You can host multiple hidden services from one server, and Tor allows you to set it up in multiple ways. If you want the same hostname to host multiple services, simply add extra **HiddenServicePort** directives under the same **HiddenServiceDir**. So for instance, if I wanted to add SSH to my existing HTTP service, I’d do this:

```
HiddenServiceDir /var/lib/tor/hidden_service/http  
HiddenServicePort 80 127.0.0.1:80  
HiddenServicePort 22 127.0.0.1:22
```

In reality though, you probably will want to segregate different services into their own .onion addresses—that way you have the option of splitting them up in the future by copying their particular directory under /var/lib/tor/hidden_service to a new server along with their configuration settings. To do that, give each service its own **HiddenServiceDir** option:

```
HiddenServiceDir /var/lib/tor/hidden_service/http
HiddenServicePort 80 127.0.0.1:80
HiddenServiceDir /var/lib/tor/hidden_service/ssh
HiddenServicePort 22 127.0.0.1:22
```

Now you can look at /var/lib/tor/hidden_service/ssh/hostname to find the new .onion address for your SSH service.

Conclusion

As you can see, configuring Tor hidden services isn't nearly as difficult as you might have originally assumed. Arguably, it will take longer for you to reconfigure your services to listen on localhost than it will to configure Tor itself! ■



Kyle Rankin is a Tech Editor and columnist at *Linux Journal* and the Chief Security Officer at Purism. He is the author of *Linux Hardening in Hostile Networks*, *DevOps Troubleshooting*, *The Official Ubuntu Server Book*, *Knoppix Hacks*, *Knoppix Pocket Reference*, *Linux Multimedia Hacks* and *Ubuntu Hacks*, and also a contributor to a number of other O'Reilly books. Rankin speaks frequently on security and open-source software including at BsidesLV, O'Reilly Security Conference, OSCON, SCALE, CactusCon, Linux World Expo and PenguinCon. You can follow him at @kylerankin.

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

Facebook Compartmentalization

I don't always use Facebook, but when I do, it's over a compartmentalized browser over Tor.

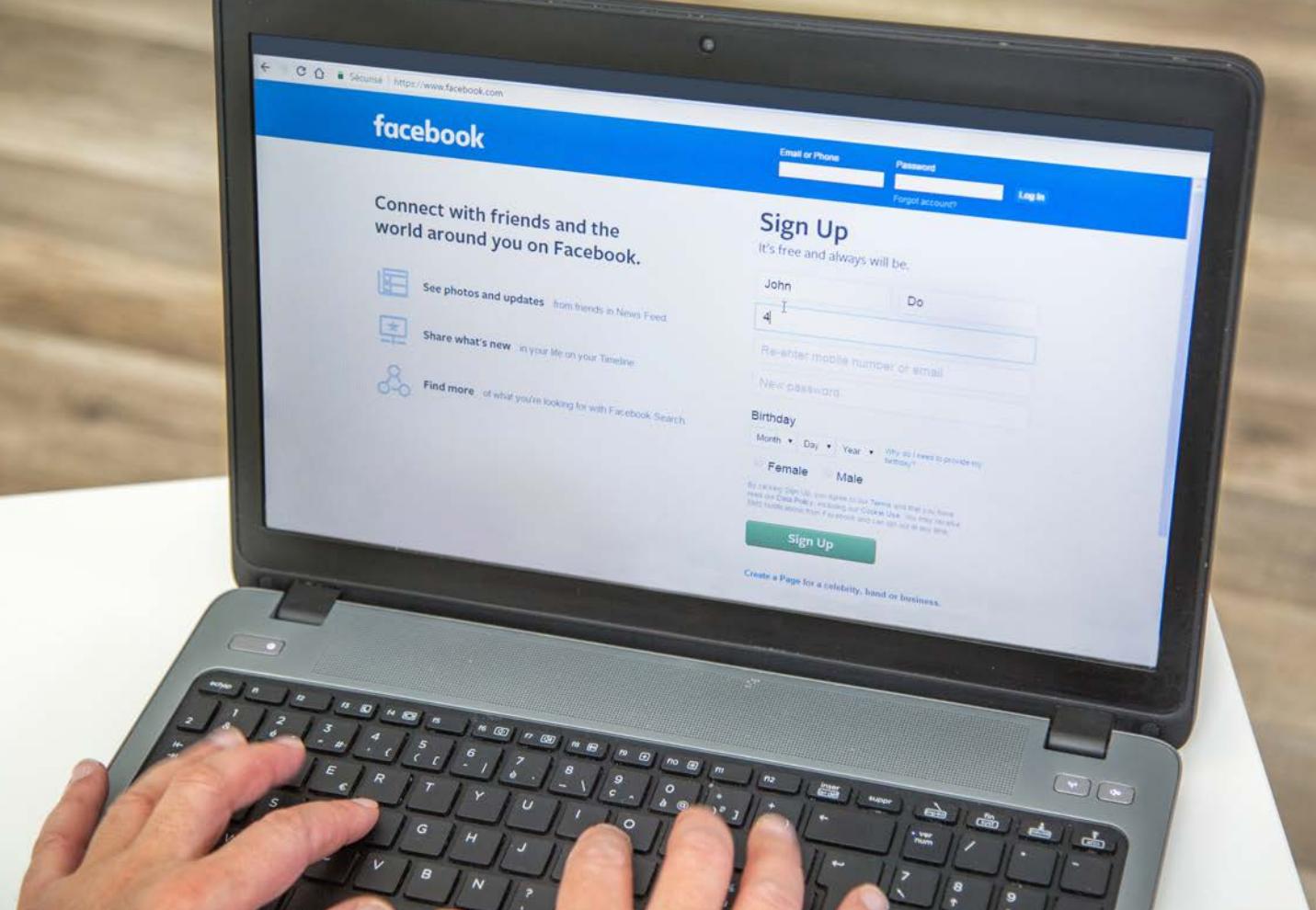
By Kyle Rankin

Whenever people talk about protecting privacy on the internet, social-media sites like Facebook inevitably come up—especially right now. It makes sense—social networks (like Facebook) provide a platform where you can share your personal data with your friends, and it doesn't come as much of a surprise to people to find out they also share that data with advertisers (it's how they pay the bills after all). It makes sense that Facebook uses data you provide when you visit that site. What some people might be surprised to know, however, is just how much. Facebook tracks them when they aren't using Facebook itself but just browsing around the web.

Some readers may solve the problem of Facebook tracking by saying “just don't use Facebook”; however, for many people, that site may be the only way they can keep in touch with some of their friends and family members. Although I don't post on Facebook much myself, I do have an account and use it to keep in touch with certain friends. So in this article, I explain how I employ compartmentalization principles to use Facebook without leaking too much other information about myself.

1. Post Only Public Information

The first rule for Facebook is that, regardless of what you think your privacy settings are, you are much better off if you treat any content you provide there as being fully public. For one, all of those different privacy and permission settings can become complicated, so it's easy to make a mistake that ends up making some of your data



more public than you'd like. Second, even with privacy settings in place, you don't have a strong guarantee that the data won't be shared with people willing to pay for it. If you treat it like a public posting ground and share only data you want the world to know, you won't get any surprises.

2. Give Facebook Its Own Browser

I mentioned before that Facebook also can track what you do when you browse other sites. Have you ever noticed little Facebook "Like" icons on other sites? Often websites will include those icons to help increase engagement on their sites. What it also does, however, is link the fact that you visited that site with your specific Facebook account—even if you didn't click "Like" or otherwise engage with the site. If you want to reduce how much you are tracked, I recommend selecting a separate browser that you use only for Facebook. So if you are a Firefox user, load Facebook in Chrome. If you are a Chrome user, view Facebook in Firefox. If you don't want to go to the trouble of managing two different browsers, at the very least, set up a separate Firefox profile (run `firefox -P` from a terminal) that you use only for Facebook.

3. View Facebook over Tor

Many people don't know that Facebook itself offers a .onion service that allows you to view Facebook over Tor. It may seem counterintuitive that a site that wants so much of your data would also want to use an anonymizing service, but it makes sense if you think it through. Sure, if you access Facebook over Tor, Facebook will know it's you that's accessing it, but it won't know from where. More important, no other sites on the internet will know you are accessing Facebook from that account, even if they try to track via IP.

To use Facebook's private .onion service, install the Tor Browser Bundle, or otherwise install Tor locally, and follow the Tor documentation to route your Facebook-only browser to its SOCKS proxy service. Then visit <https://facebookcorewwwi.onion>, and only you and Facebook will know you are hitting the site. By the way, one advantage to setting up a separate browser that uses a SOCKS proxy instead of the Tor Browser Bundle is that the Tor Browser Bundle attempts to be stateless, so you will have a tougher time making the Facebook .onion address your home page.

Conclusion

So sure, you could decide to opt out of Facebook altogether, but if you don't have that luxury, I hope a few of these compartmentalization steps will help you use Facebook in a way that doesn't completely remove your privacy. ■



Kyle Rankin is a Tech Editor and columnist at *Linux Journal* and the Chief Security Officer at Purism. He is the author of *Linux Hardening in Hostile Networks*, *DevOps Troubleshooting*, *The Official Ubuntu Server Book*, *Knoppix Hacks*, *Knoppix Pocket Reference*, *Linux Multimedia Hacks* and *Ubuntu Hacks*, and also a contributor to a number of other O'Reilly books. Rankin speaks frequently on security and open-source software including at BsidesLV, O'Reilly Security Conference, OSCON, SCALE, CactusCon, Linux World Expo and PenguinCon. You can follow him at @kylerankin.

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



The Fight for Control: Andrew Lee on Open-Sourcing PIA

When I learned that our new sister company, Private Internet Access (PIA), was opening its source code, I immediately wanted to know the backstory, especially since privacy is the theme of this month's *Linux Journal*. So I contacted Andrew Lee, who founded PIA, and an interview ensued. Here it is.

By Doc Searls

DS: What made you start PIA in the first place? Did you have a particular population or use case—or set of use cases—in mind?

AL: Primarily PIA was rooted in my humble beginnings on IRC where it had quickly become important to protect one's IP from exposure using an IRC bouncer. However, due to jumping around in various industries thereafter, I learned a lot and came to an understanding that it was time for privacy to go mainstream, not in the "hide yourself" type of sense, but simply in the "don't watch me" sense.

DS: Had you wanted to open-source the code base all along? If not, why now?

AL: We always wanted to open-source the code base, and we finally got around to it. It's late, but late is better than never. We were incredibly busy, and we didn't prioritize it enough, but by analyzing our philosophies deeply, we've been able to re-prioritize things internally. Along with open-sourcing our software, there are a lot of great things to come.

DS: People always wonder if open-sourcing a code base affects a business model. Our readers have long known that it doesn't, and that open-sourcing in fact opens more possibilities than leaving code closed. But it would be good to hear your position on the topic, since I'm sure you've thought about it.

AL: Since Private Internet Access is a service, having open-source code does not affect the business' ability to generate revenue as a company aiming for sustainable activism. Instead, I do believe we're going to end up with better and stronger software as an outcome.

DS: Speaking of activism, back in March, you made a very strong statement, directly to President Trump and Congress, with a two-page ad in *The New York Times*, urging them to kill off SESTA-FOSTA. I'm curious to know if we'll be seeing more of that and to hear what the response was at the time.

AL: Absolutely! We ran a few newspaper campaigns, including one for the Internet Defense League. It's a very strong place to mobilize people for important issues for

DEEP DIVE

Dear America,

Imagine a world where you can't say what you want on the Internet.

Let's tweet @realDonaldTrump and tell him how you feel while we still can.



#HaveAVoice

(This space intentionally left as blank as most of the internet will be if you sign SESTA-FOSTA.)

Dear President Trump,

If you sign SESTA-FOSTA into law, even Twitter will be affected.

The title may sound good, but it isn't what it seems.

If you sign this bill, every website will be affected.

Free speech dies.

Weren't you Making America Great Again?

Signing SESTA-FOSTA into law is an act of war against the Constitution and the people.

If you don't believe us, listen to the American people.



#HaveAVoice

(This space intentionally left as blank as most of the internet will be if you sign SESTA-FOSTA.)

society. As a result of the campaign, many tweets from concerned Americans were received by President Trump. I would say it was a success, but from here it's up to our President. Let's hope he does the right thing and vetoes it. That said, if the bill is signed in its current form [which it was after this interview was conducted], the internet is routing, and the cypherpunks have the power of the crypto. We will decentralize and route around bad policy.

DS: Our readers have always cared a lot about licenses, so here's a question

for them: why the MIT license?

AL: Our internal open-source task force was given the mission of choosing the least restrictive open-source license possible, and they landed on MIT. I hope that anyone and everyone can benefit from our code however they see fit.

DS: Why release code repositories gradually instead of all at once? What kind of work do you need to do to make the code ready?

AL: In order to release our code properly, we're making sure we're distributing everything properly and with clean, readable code.

DS: Is the code on GitHub?

AL: Yes, at <https://pia-foss.github.io>.

DS: Tell us more about the VPN industry. How has it changed since you started PIA? And do you expect that open-sourcing PIA's code will help the company lead the market in new ways?

AL: I think a lot more companies have entered the VPN industry. For us, open-sourcing our code is part of a multi-part strategy to create what we call the “next VPN”. We’re not intending to lead the market, but instead to create a new market that will essentially put the existing market, in its current form, into extinction immediately. This strategy includes a heap of technology stacks we are building internally as well as simple feature additions. While we’ve definitely earned the reputation as the most-trusted VPN in the space, the primary goal of the our “next VPN” project is to remove trust from the equation. After all, we’re strong believers in the words “in crypto we trust”.

DS: I know PIA always has been adamant about not logging its customers. In 2015, the company had a chance to show why when a court subpoenaed customer usage records—and it was unable to provide any. I’d like to hear more about your philosophy there.

AL: Simply put, everyone has a right to privacy, but there are also choices. That's why I think it is imperative for people in the VPN consumer market to do research beyond simple reviews. Instead, find forums and look for dirt on companies. That's the best way to verify any company—in our space or any others. Do searches that fill in the blanks on who sucks, who monitors their users, who logs their users and so on.

DS: Make the connection, if you don't mind, between open source and privacy.

AL: For us, open-sourcing is vital given that, in order to protect one's privacy, it is important for people to know exactly what it is their software is doing. Having the source code available makes this possible. I also believe that it further enhances security, in addition to our third-party audits that we already performed, since more eyes will be able to review the code.

DS: What's next?

AL: With PIA we're really building the “next VPN”, and it will be more private than the way current providers look, on an order of magnitude. However, I really don't want to talk about it. We prefer to deliver, rather than talking about what we'll deliver.

DS: Today we're seeing the pendulum swinging toward decentralization, and greater individual autonomy and control. I'd love to hear about how you see that playing out, in what sequence and with what likely populations.

AL: Everyone has a different threat model, and everyone needs clear choices about trade-offs. To start, we're providing Tor to provide people with one of the most essential choices. I don't like talking about stuff in the pipeline, but I will say we're launching full Tor support in all of our clients on desktop and mobile. This is going to allow our end users to route through Tor, which effectively allows them to mask their identity further. I believe that this will be used by a smaller set of users than our overall customer base, because the Tor network is still small. However, by educating people about Tor while the network grows, so will Tor's efficiency.

DS: I assume crypto will be involved. Can you say more about how?

AL: Everything we do uses crypto, from the algorithms used to even some of the accepted payment methods, such as cryptocurrencies. In launching our “next VPN” solution, we are relying heavily on cryptography and the unique applications to which it can be applied. It’s pretty crazy that nature, and brilliant people, have given us a gift, weapon and protection in the form of cryptography, and we’re damn sure going to be betting everything we’ve got on it.

DS: How do you see VPN usage, and the whole VPN market, evolving and changing, especially in different settings?

AL: I believe the VPN market and usage will continue to increase, as it already has, given the political and social climate. Many countries and companies are totally abusing their citizens and users, and people are learning that they need to take matters into their own hands to protect themselves.

DS: With all the bad news around Facebook and the approach of the GDPR and other privacy regulations, what changes do you see coming, from your perspective as a provider of privacy tech?

AL: I’m guessing privacy will continue to be an important value to people. That being said, the government is constantly in a never-ending battle to remove our privacy. This is pretty bad, because privacy is more than just our right to disclose information to whomever we want, when we want; it’s actually a tool that allows us to be unique. Without privacy, we will all become conformists and do whatever it is “the man” deems appropriate. Screw that world. Seriously. We have to blend into the crowd and become anonymous, believe it or not, in order to become different and characteristically unique. But this fight won’t be easy. It’s a fight for control.

DS: So you see a conflict coming—or already here and headed into some showdowns?

AL: Yes. My perspective is that the Crypto War is heading to the main event. We’ll all

need to work together to fight for the sake of cryptography and, even more broadly, the internet. ■



Doc Searls is a veteran journalist, author and part-time academic who spent more than two decades elsewhere on the *Linux Journal* masthead before becoming Editor in Chief when the magazine was reborn in January 2018. His two books are *The Cluetrain Manifesto*, which he co-wrote for Basic Books in 2000 and updated in 2010, and *The Intention Economy: When Customers Take Charge*, which he wrote for Harvard Business Review Press in 2012. On the academic front, Doc runs ProjectVRM, hosted at Harvard's Berkman Klein Center for Internet and Society, where he served as a fellow from 2006–2010. He was also a visiting scholar at NYU's graduate school of journalism from 2012–2014, and he has been a fellow at UC Santa Barbara's Center for Information Technology and Society since 2006, studying the internet as a form of infrastructure.

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

Programming in Color with ncurses

Jim demonstrates color manipulation with `curses` by adding colors to his terminal adventure game.

By Jim Hall

In parts [one](#) and [two](#) of my article series about programming with the `ncurses` library, I introduced a few `curses` functions to draw text on the screen, query characters from the screen and read from the keyboard. To demonstrate several of these functions, I created a simple adventure game in `curses` that drew a game map and player character using simple characters. In this follow-up article, I show how to add color to a `curses` program.

Drawing on the screen is all very well and good, but if it's all white-on-black text, your program might seem dull. Colors can help convey more information—for example, if your program needs to indicate success or failure. In such a case, you could display text in green or red to help emphasize the outcome. Or, maybe you simply want to use colors to “snazz” up your program to make it look prettier.

In this article, I use a simple example to demonstrate color manipulation via the `curses` functions. In my previous article, I wrote a basic adventure-style game that lets you move a player character around a crudely drawn map. However, the map was entirely black and white text, relying on shapes to suggest water (~) or mountains (^), so let's update the game to use colors.

Color Essentials

Before you can use colors, your program needs to know if it can rely on the terminal to display the colors correctly. On modern systems, this always should be true. But

in the classic days of computing, some terminals were monochromatic, such as the venerable VT52 and VT100 terminals, usually providing white-on-black or green-on-black text.

To query the terminal capability for colors, use the `has_colors()` function. This will return a true value if the terminal can display color, and a false value if not. It is usually used to start an `if` block, like this:

```
if (has_colors() == FALSE) {  
    endwin();  
    printf("Your terminal does not support color\n");  
    exit(1);  
}
```

Having determined that the terminal can display color, you then can set up `curses` to use colors with the `start_color()` function. Now you're ready to define the colors your program will use.

In `curses`, you define colors in pairs: a foreground color on a background color. This allows `curses` to set both color attributes at once, which often is what you want to do. To establish a color pair, use `init_pair()` to define a foreground and background color, and associate it to an index number. The general syntax is:

```
init_pair(index, foreground, background);
```

Consoles support only eight basic colors: black, red, green, yellow, blue, magenta, cyan and white. These colors are defined for you with the following names:

- `COLOR_BLACK`
- `COLOR_RED`
- `COLOR_GREEN`
- `COLOR_YELLOW`
- `COLOR_BLUE`

- COLOR_MAGENTA
- COLOR_CYAN
- COLOR_WHITE

Applying the Colors

In my adventure game, I'd like the grassy areas to be green and the player's "trail" to be a subtle yellow-on-green dotted path. Water should be blue, with the tildes in the similar cyan color. I'd like mountains to be grey, but black text on a white background should make for a reasonable compromise. To make the player's character more visible, I'd like to use a garish red-on-magenta scheme. I can define these color pairs like so:

```
start_color();
init_pair(1, COLOR_YELLOW, COLOR_GREEN);
init_pair(2, COLOR_CYAN, COLOR_BLUE);
init_pair(3, COLOR_BLACK, COLOR_WHITE);
init_pair(4, COLOR_RED, COLOR_MAGENTA);
```

To make my color pairs easy to remember, my program defines a few symbolic constants:

```
#define GRASS_PAIR      1
#define EMPTY_PAIR       1
#define WATER_PAIR       2
#define MOUNTAIN_PAIR    3
#define PLAYER_PAIR      4
```

With these constants, my color definitions become:

```
start_color();
init_pair(GRASS_PAIR, COLOR_YELLOW, COLOR_GREEN);
init_pair(WATER_PAIR, COLOR_CYAN, COLOR_BLUE);
```

```
init_pair(MOUNTAIN_PAIR, COLOR_BLACK, COLOR_WHITE);
init_pair(PLAYER_PAIR, COLOR_RED, COLOR_MAGENTA);
```

Whenever you want to display text using a color, you just need to tell `curses` to set that color attribute. For good programming practice, you also should tell `curses` to undo the color combination when you’re done using the colors. To set the color, use `attron()` before calling functions like `mvaddch()`, and then turn off the color attributes with `attroff()` afterward. For example, when I draw the player’s character, I might do this:

```
attron(COLOR_PAIR(PLAYER_PAIR));
mvaddch(y, x, PLAYER);
attroff(COLOR_PAIR(PLAYER_PAIR));
```

Note that applying colors to your programs adds a subtle change to how you query the screen. Normally, the value returned by `mvinch()` is of type `chtype`. Without color attributes, this is basically an integer and can be used as such. But, colors add extra attributes to the characters on the screen, so `chtype` carries extra color information in an extended bit pattern. If you use `mvinch()`, the returned value will contain this extra color value. To extract just the “text” value, such as in the `is_move_okay()` function, you need to apply a bitwise & with the `A_CHARTEXT` bit mask:

```
int is_move_okay(int y, int x)
{
    int testch;

    /* return true if the space is okay to move into */

    testch = mvinch(y, x);
    return (((testch & A_CHARTEXT) == GRASS)
            || ((testch & A_CHARTEXT) == EMPTY));
}
```

With these changes, I can update the adventure game to use colors:

```
/* quest.c */

#include <curses.h>
#include <stdlib.h>

#define GRASS      ' '
#define EMPTY      ' . '
#define WATER      ' ~ '
#define MOUNTAIN   ' ^ '
#define PLAYER     ' * '

#define GRASS_PAIR    1
#define EMPTY_PAIR    1
#define WATER_PAIR    2
#define MOUNTAIN_PAIR 3
#define PLAYER_PAIR   4

int is_move_okay(int y, int x);
void draw_map(void);

int main(void)
{
    int y, x;
    int ch;

    /* initialize curses */

    initscr();
    keypad(stdscr, TRUE);
    cbreak();
    noecho();
```

```
/* initialize colors */

if (has_colors() == FALSE) {
    endwin();
    printf("Your terminal does not support color\n");
    exit(1);
}

start_color();
init_pair(GRASS_PAIR, COLOR_YELLOW, COLOR_GREEN);
init_pair(WATER_PAIR, COLOR_CYAN, COLOR_BLUE);
init_pair(MOUNTAIN_PAIR, COLOR_BLACK, COLOR_WHITE);
init_pair(PLAYER_PAIR, COLOR_RED, COLOR_MAGENTA);

clear();

/* initialize the quest map */

draw_map();

/* start player at lower-left */

y = LINES - 1;
x = 0;

do {

    /* by default, you get a blinking cursor - use it to
       indicate player * */

    attron(COLOR_PAIR(PLAYER_PAIR));
    mvaddch(y, x, PLAYER);
```

```
attroff(COLOR_PAIR(PLAYER_PAIR));
move(y, x);
refresh();

ch = getch();

/* test inputted key and determine direction */

switch (ch) {
case KEY_UP:
case 'w':
case 'W':
    if ((y > 0) && is_move_okay(y - 1, x)) {
        attron(COLOR_PAIR(EMPTY_PAIR));
        mvaddch(y, x, EMPTY);
        attroff(COLOR_PAIR(EMPTY_PAIR));
        y = y - 1;
    }
    break;
case KEY_DOWN:
case 's':
case 'S':
    if ((y < LINES - 1) && is_move_okay(y + 1, x)) {
        attron(COLOR_PAIR(EMPTY_PAIR));
        mvaddch(y, x, EMPTY);
        attroff(COLOR_PAIR(EMPTY_PAIR));
        y = y + 1;
    }
    break;
case KEY_LEFT:
case 'a':
case 'A':
    if ((x > 0) && is_move_okay(y, x - 1)) {
```

```
    attron(COLOR_PAIR(EMPTY_PAIR));
    mvaddch(y, x, EMPTY);
   attroff(COLOR_PAIR(EMPTY_PAIR));
    x = x - 1;
}
break;
case KEY_RIGHT:
case 'd':
case 'D':
    if ((x < COLS - 1) && is_move_okay(y, x + 1)) {
        attron(COLOR_PAIR(EMPTY_PAIR));
        mvaddch(y, x, EMPTY);
       attroff(COLOR_PAIR(EMPTY_PAIR));
        x = x + 1;
    }
break;
}
}
while ((ch != 'q') && (ch != 'Q'));

endwin();

exit(0);
}

int is_move_okay(int y, int x)
{
    int testch;

/* return true if the space is okay to move into */

testch = mvinch(y, x);
return (((testch & A_CHARTEXT) == GRASS)
```

```
    || ((testch & A_CHARTEXT) == EMPTY));  
}  
  
void draw_map(void)  
{  
    int y, x;  
  
    /* draw the quest map */  
  
    /* background */  
  
    attron(COLOR_PAIR(GRASS_PAIR));  
    for (y = 0; y < LINES; y++) {  
        mvhline(y, 0, GRASS, COLS);  
    }  
   attroff(COLOR_PAIR(GRASS_PAIR));  
  
    /* mountains, and mountain path */  
  
    attron(COLOR_PAIR(MOUNTAIN_PAIR));  
    for (x = COLS / 2; x < COLS * 3 / 4; x++) {  
        mvvline(0, x, MOUNTAIN, LINES);  
    }  
   attroff(COLOR_PAIR(MOUNTAIN_PAIR));  
  
    attron(COLOR_PAIR(GRASS_PAIR));  
    mvhline(LINES / 4, 0, GRASS, COLS);  
   attroff(COLOR_PAIR(GRASS_PAIR));  
  
    /* lake */  
  
    attron(COLOR_PAIR(WATER_PAIR));  
    for (y = 1; y < LINES / 2; y++) {
```

```
    mvhline(y, 1, WATER, COLS / 3);
}
attroff(COLOR_PAIR(WATER_PAIR));
}
```

Unless you have a keen eye, you may not be able to spot all of the changes necessary to support color in the adventure game. The `diff` tool shows all the instances where functions were added or code was changed to support colors:

```
$ diff quest-color/quest.c quest/quest.c
12,17d11
< #define GRASS_PAIR      1
< #define EMPTY_PAIR      1
< #define WATER_PAIR      2
< #define MOUNTAIN_PAIR   3
< #define PLAYER_PAIR     4
<
33,46d26
<     /* initialize colors */
<
<     if (has_colors() == FALSE) {
<         endwin();
<         printf("Your terminal does not support color\n");
<         exit(1);
<     }
<
<     start_color();
<     init_pair(GRASS_PAIR, COLOR_YELLOW, COLOR_GREEN);
<     init_pair(WATER_PAIR, COLOR_CYAN, COLOR_BLUE);
<     init_pair(MOUNTAIN_PAIR, COLOR_BLACK, COLOR_WHITE);
<     init_pair(PLAYER_PAIR, COLOR_RED, COLOR_MAGENTA);
<
61d40
```

```
<      attron(COLOR_PAIR(PLAYER_PAIR));
63d41
<      attroff(COLOR_PAIR(PLAYER_PAIR));
76d53
<              attron(COLOR_PAIR(EMPTY_PAIR));
78d54
<              attroff(COLOR_PAIR(EMPTY_PAIR));
86d61
<              attron(COLOR_PAIR(EMPTY_PAIR));
88d62
<              attroff(COLOR_PAIR(EMPTY_PAIR));
96d69
<              attron(COLOR_PAIR(EMPTY_PAIR));
98d70
<              attroff(COLOR_PAIR(EMPTY_PAIR));
106d77
<              attron(COLOR_PAIR(EMPTY_PAIR));
108d78
<              attroff(COLOR_PAIR(EMPTY_PAIR));
128,129c98
<      return (((testch & A_CHARTEXT) == GRASS)
<             || ((testch & A_CHARTEXT) == EMPTY));
---
>      return ((testch == GRASS) || (testch == EMPTY));
140d108
<      attron(COLOR_PAIR(GRASS_PAIR));
144d111
<      attroff(COLOR_PAIR(GRASS_PAIR));
148d114
<      attron(COLOR_PAIR(MOUNTAIN_PAIR));
152d117
<      attroff(COLOR_PAIR(MOUNTAIN_PAIR));
154d118
```

```
<      attron(COLOR_PAIR(GRASS_PAIR));  
156d119  
<     attroff(COLOR_PAIR(GRASS_PAIR));  
160d122  
<      attron(COLOR_PAIR(WATER_PAIR));  
164d125  
<     attroff(COLOR_PAIR(WATER_PAIR));
```

Let's Play—Now in Color

The program now has a more pleasant color scheme, more closely matching the original tabletop gaming map, with green fields, blue lake and imposing gray mountains. The hero clearly stands out in red and magenta livery.



Figure 1. A Simple Tabletop Game Map, with a Lake and Mountains



Figure 2. The player starts the game in the lower-left corner.



Figure 3. The player can move around the play area, such as around the lake, through the mountain pass and into unknown regions.

With colors, you can represent information more clearly. This simple example uses colors to indicate playable areas (green) versus impassable regions (blue or gray). I hope you

will use this example game as a starting point or reference for your own programs. You can do so much more with **curses**, depending on what you need your program to do.

In a follow-up article, I plan to demonstrate other features of the **ncurses** library, such as how to create windows and frames. In the meantime, if you are interested in learning more about **curses**, I encourage you to read Pradeep Padala's **NCURSES Programming HOWTO**, at the Linux Documentation Project. ■

Jim Hall is an advocate for free and open-source software, best known for his work on the FreeDOS Project, and he also focuses on the usability of open-source software. Jim is the Chief Information Officer at Ramsey County, Minnesota.

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

FOSS as a Part of a Corporate Sustainability Plan

Free and open-source software is a critical part of your company's supply chain. Here's why and how you can include it in your corporate sustainability plan.

By VM (aka Vicky) Brasseur

In 1983 the United Nations convened a commission of 22 people to investigate the question of the worldwide environmental and social impact of human development. Four years later, in 1987, the commission released **Our Common Future**, more commonly known as the *Brundtland Report* in honour of Gro Harlem Brundtland, chairperson of the commission. This report detailed the very real socio-environmental issues facing humanity. One of its recommendations was for governments, organizations and companies to start engaging in what it called *sustainable development*. That is, "...development that meets the needs of the present without compromising the ability of future generations to meet their own needs".

Since then there's been **steep growth** in the number of corporations that maintain and operate according to a corporate sustainability plan. These plans encompass environmental as well as social aspects of doing business. They encompass actions within an organization—such as natural resource usage, diversity and inclusion, and fair treatment of employees—as well as those external to the organization—such as the sustainability operations of their entire supply chain as well as the overall impact the corporation has on the Earth and its inhabitants.

The Benefits of Sustainability

A sustainability plan impacts every facet of an organization's operations and can take a fair bit of effort to implement and maintain. If that's the case, why are more corporations putting these plans into action every year? While it would be nice to think that this occurs for entirely altruistic reasons—taking care of the Earth and its inhabitants is simply the right thing to do, after all—the fact of the matter is that **studies repeatedly show** that properly implemented corporate sustainability plans are very good for the bottom line.

Innovations and Profitability

Sustainability requires partnering and collaborating not only across groups within the organization, but also with many external to it. These collaborations expose a corporation to new ideas and innovations in market, process, policy and product, all of which can lead to improved profitability while moving the company toward its sustainability goals. Without the support and engagement of all employees, a sustainability plan will fail. Therefore, the execution of a sustainability strategy requires a shift in management from the old style of command and control toward a modern style of trust, communication and employee development. By empowering and training employees, a sustainable corporation improves its ability to execute plans while also increasing employee satisfaction and retention. Beyond just innovations, profitability is increased through reduced expenditures, entry into new markets and a lower cost of capital.

Investments

Investments are also impacted, as more investors are now looking at a corporation's sustainability plan as a deciding factor on where to place their money. When you think about it, it's surprising that sustainability hasn't been a primary investor consideration before now. The strategies that support a successful corporate sustainability plan, focusing on what's good for society and the environment, are also those that do good for the company. Protecting the organization's value and supply chains enables it to ensure it will be around for a good, long while. That longevity—and plan to maintain it—is very appealing to investors in search of both stability and a positive return on their investments. Despite this, **studies show** that most high-level corporate managers

still don't recognize the importance sustainability holds for investors. This provides a great opportunity for those managers who can launch and make a corporate sustainability plan successful before their competitors.

So, yes, corporate sustainability planning (when done properly) is that proverbial rising tide that floats all boats. However, for companies that rely upon technology to keep their business operating (read: all of them), there's a major component that's usually left out of their sustainability plan: the software.

Open-Source Software Is a Part of Your Supply Chain

Operating systems, databases, libraries, infrastructure management and orchestration—no matter what component, all software is a part of your company's supply chain. How much do you know about it? Do you know what your company's software supply chain looks like? How much of that chain is or depends upon free and open-source software? If you haven't investigated this before, you'll likely be surprised by the answer. If you look at little deeper, you'll be even more surprised to discover just how few people maintain and support the software on which your company relies to operate and do business.

However you look at it, this arrangement is fraught with sustainability concerns. Socially, allowing so few people to perform so much work for so little compensation is inappropriate and does not scale. From a business perspective, it's very poor practice to rely upon suppliers who may burn out and disappear at any moment. This does nothing good for the longevity prospects for your organization and dramatically increases its risk profile.

Open-Source Software Is Good for Sustainability

Don't get the wrong idea: I'm not here to spread Fear, Uncertainty, and Doubt (aka *FUD*) about having free and open-source software projects as a part of your supply chain. Even a quick glance will show that you're unlikely to receive as much value and flexibility from proprietary solutions (if they even exist). Proprietary software is incredibly expensive, less flexible, leads to vendor lock-in and is at least as likely to disappear without notice. **90% of software startups fail**, taking their products and

their code with them. The high cost, low innovation and equal risk of mortality for proprietary software makes it a less appealing solution when considering the longevity and sustainability of your own company. Free and open-source solutions are simply the better option.

What Is YOUR Software Supply Chain?

The answer, then, is not to cut out the free and open-source links in your supply chain. No, the actual answer is to include free and open-source software in your corporate sustainability plan. Unlike proprietary software, free and open-source (FOSS) solutions enable your company to take action to ensure that the FOSS projects themselves are sustainable. So, how can your company do that?

For starters, if you haven't done so yet, take this opportunity to become very familiar with your company's software supply chain. Certain links in that chain will be more important and more strategic than others. While your software sustainability efforts should include all of the links in the chain—FOSS or proprietary—you may find that it makes sense to focus more of your investment in these strategic links.

Build a Culture of Contribution

As you're evaluating and integrating your software supply chain into your corporate sustainability plan, start communicating internally about the importance of this supply chain. This communication should be across the entire organization, not simply constrained to the technical departments. By communicating and including the entire company, you open up more avenues for ideas and innovations in sustainability as well as start to build a culture of sharing and contribution that will aid in all your sustainability efforts, FOSS or otherwise.

That culture of contribution will be critical for the next step: empowering staff members to contribute back to the FOSS projects on which your company relies. Reporting bugs and requesting features is one way to contribute, but fixing bugs, adding features and contributing those changes upstream to the main project is what will lead to improved project longevity. It often can make sense for a company to pay team members to work full time maintaining and enhancing strategic free

and open-source software projects, but even small contributions submitted by individuals scattered across the organization can aggregate into a large amount of work toward sustaining the projects on which your company relies. Making it easy for team members to contribute will make it much more likely that they will support the projects through their contributions. Remember also that FOSS projects need more than just programmers in order to thrive. Enabling contributions from your QA, design, marketing and other departments will enhance the entire ecosystem of the free and open-source software that enables your company to operate, adding to its longevity and reducing risk both for it and for your own company.

Share the Load

As you review your software supply chain, consider the problem that each one of those links in the chain is there to solve. Your company is not the only one to have those problems. These are issues faced by many organizations, and no one company will be able to solve these shared problems.

Therefore, the next step in integrating free and open-source software into your corporate sustainability planning is to release the software your company has developed to solve problems shared by others. Of course, this doesn't mean you should necessarily **release the software that provides legitimate market differentiation and value for your company**, but there are undoubtedly several tools developed internally that are not of strategic market value but that solve specific problems. By releasing those tools, you enable collaboration with organizations that share similar problems.

By working together to solve these problems through free and open-source software, all collaborators share the load of development. This allows them to benefit from the tool while also freeing up internal resources to focus on strategic value creation for their markets. In a case study created by **Open Tech Strategies** and **released by the World Bank**, collaboration on free and open-source software tools led to a 200% return on investment for the collaborators. Sharing the burden of maintaining these tools ensures their longevity and stability while reducing institutional risk. The end result is a more sustainable business and a healthier free

and open-source software ecosystem.

Join a Foundation

Depending upon the free and open-source software in your supply chain, one approach toward sustainability may be participating in the organizations and foundations formed to support projects that are strategic to your company. By joining organizations like [Open Source Initiative](#), [Eclipse Foundation](#), [Apache Software Foundation](#), and the [Linux Foundation](#), your organization not only has a seat at the table when discussing the future of strategic FOSS projects, it also gets the opportunity to meet, discuss and collaborate with companies it may not have had occasion to interact with otherwise. By bringing together disparate organizations and providing not only a shared purpose but also a safe environment for collaboration, these foundations enable member companies to learn from each other and to open the doors for new partnerships and innovations.

Think Strategically

I've said it already in this article, but it's worth repeating that, just as with the rest of your corporate sustainability plan, it's crucial that your company's approach to supporting free and open-source software projects be *strategic*. Although it's tempting to contribute to all FOSS projects equally—and certainly the projects themselves would not complain—from a sustainability point of view, your organization should try to focus its resources on those projects that make a material difference to your company. *Material difference* means that a project, were it to stop being developed or maintained, would [severely impact the operations of your company](#). These are the FOSS projects most directly linked to your organization's longevity, and it also makes them risk factors for your corporate sustainability plan.

In order to determine which projects are material to your corporation, you must be fully aware of your software supply chain. This involves not only looking at the free and open-source software that allows your company to operate, but also digging deeper to learn on what projects *those* projects rely. It's all well and good to acknowledge that the authentication library used for your product is material to company longevity, but if that library itself relies upon cryptographic functionality

in another—woefully under-funded—project, your company may be exposed to **unforeseen but preventable risks**. Ignorance of your software supply chain is no defense against vulnerabilities. Cultivating an awareness of the entire ecosystem within which strategic and material FOSS projects operate will help secure corporate longevity and sustainability.

Make a Business Case

Focusing software sustainability efforts on material and strategic concerns makes it considerably easier to incorporate free and open-source software into the business case for your company's corporate sustainability plan. It may be tempting to skip this step, but if you do so **studies show that it's likely your sustainability plan will fail**. When you pause to consider this, it makes perfect sense. Why should a corporation invest so many resources in shifting the way it does business, up to and including its very business model, if there is nothing in it for them? Again, setting aside altruistic appeals and the fact that taking care of the Earth and its inhabitants is just the right thing to do, being able to make a well-researched and detailed case that doing so is good for the bottom line makes it more likely that even the less environmentally and socially minded leaders in your company will get on board with the plan. Don't simply list what changes will be necessary to enact a corporate sustainability plan, but also reveal exactly what is in it for the company for making this effort.

You're All in This Together

As mentioned earlier, it's important for any corporate sustainability plan to create a culture of contribution across the entire organization. If upper management does not communicate what the plan is meant to accomplish and why, it will fail to get the support of the employees who actually have to do the heavy lifting of implementing that plan. Without the support of the entire organization, **your corporate sustainability plan will fail**. Sustainability must be a 360-degree effort, not simply something dictated from on high. All members and stakeholders of the company must understand what the sustainability plan is, what impact it will have, what part they specifically will play, and how they will make a difference to the overall outcome.

This communication is hard enough when the plan is simply about environmental and

social concerns, issues with which most people are familiar, but it becomes much more difficult when unfamiliar concepts like “free software” and “open source” enter the mix. This is why it’s so important to begin the discussion as early as possible. It can take some time for people to understand that “contribute back to free and open-source software projects” can be as meaningful and impactful to the corporate sustainability plan as “reduce water usage”, “use energy from renewable sources” or “eliminate the gender/race pay gap”.

To help with this communication, rather than dictating implementation details to middle management, company leaders should instead set software sustainability goals for each team or department and then allow each group to define how it can best contribute toward those goals. For technical departments, their contributions may be as straightforward as providing patches or releasing internal tools, while less technical departments may come up with less obvious approaches to helping the effort. By empowering each team to approach the problem in its own way, the company is not only encouraging innovation, but it’s also building trust within its ranks. This trust, as mentioned before, increases the effectiveness not only of the sustainability plan but also of other operations undertaken by these teams. This trust also leads to higher employee satisfaction and retention. The overall culture of the company evolves into one that is more collaborative, innovative and therefore sustainable.

Keep Yourself Accountable

A vital part of any corporate sustainability plan is the company holding itself accountable for delivering on its promises to itself, to its stakeholders and to the communities on which it relies. Fostering a culture of collaboration includes supporting open communication. That openness must permeate all levels of the company. Embracing and supporting free and open-source software and their underlying values can help cultivate the communication required for accountability. Instituting **inner sourcing** is a good way to encourage collaboration, communication and cross-functional accountability on your corporate sustainability plan.

Accountability also can be included at an organizational level by way of performance reviews. For this to work, all employees must understand how they contribute to

the overall corporate sustainability plan so that they can work toward meeting their particular goals. This is especially important at the executive level. When leaders are held accountable for their sustainability strategies and plans, it reinforces the importance of the effort to the organization and encourages participation by all members. Reporting on the company's progress on its sustainability plan also provides accountability to external stakeholders like investors, partners and customers. For many companies, it can be uncomfortable at first to embrace this form of open communication both internally and externally, but if they are able to make this shift, they'll **reap the benefits of sustainability**. It can be a difficult journey, but it's one entirely worth taking. ■

VM (aka Vicky) spent most of her 20 years in the tech industry leading software development departments and teams, and providing technical management and leadership consulting for small and medium businesses. Now she leverages nearly 30 years of free and open-source software experience and a strong business background to advise companies about free/open source, technology, community, business and the intersections between them. She is the author of *Forge Your Future with Open Source*, the first book to detail how to contribute to free and open-source software projects. Think of it as the missing manual of open-source contributions and community participation. The book is published by The Pragmatic Programmers and is now available in an early release beta version at <https://fossforge.com>. Vicky is the proud winner of the Perl White Camel Award (2014) and the O'Reilly Open Source Award (2016). She's a moderator and author for opensource.com, a Director for the Open Source Initiative, and a frequent and popular speaker at free/open-source conferences and events. She blogs about free/open source, business and technical management at <http://anonymousshash.vmbrasseur.com>.

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

Nextcloud 13: How to Get Started and Why You Should

Nextcloud could be the first step toward replacing proprietary services like Dropbox and Skype.

By Marco Fioretti

In its simplest form, the [Nextcloud](#) server is “just” a personal, free software alternative to services like Dropbox or iCloud. You can set it up so your files are always accessible via the internet, from wherever you are, and share them with your friends. However, Nextcloud can do so much more.

In this article, I first describe what the Nextcloud server is and how to install and set it up on GNU/Linux systems. Then I explain how to configure the optional Nextcloud features, which may be the first steps toward making Nextcloud the shell of a complete replacement for many proprietary platforms existing today, such as Dropbox, Facebook and Skype.

Why Nextcloud and Not ownCloud?

Nextcloud, whose version 13 was released in February 2018, was spun off the popular [ownCloud](#) project in 2016, out of licensing and other disagreements. See the Resources section for some of the most complete feature-by-feature comparisons between Nextcloud and ownCloud. The most basic capabilities are still almost identical, two years after the fork. Some of the functions described here, however, are easier to integrate in Nextcloud than in its ancestor. In addition, my personal reasons for recommending Nextcloud over ownCloud are the following:

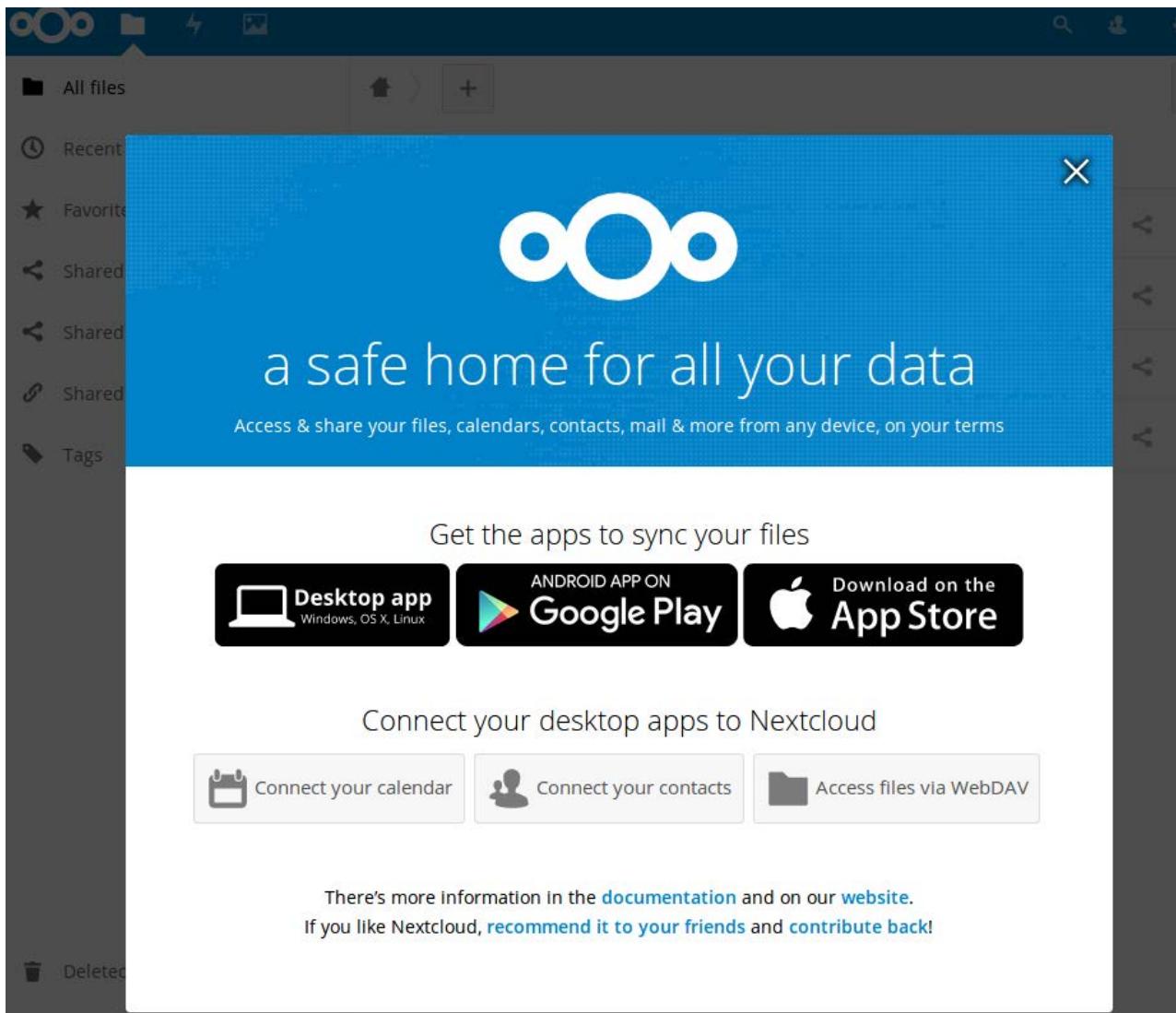


Figure 1. A safe home for all your data that all your devices can reach—that’s what Nextcloud wants to be.

- Licensing and pricing policies: all the official components of Nextcloud are both free as in freedom and as in free beer. You pay only for support and update services. That's not the case with ownCloud.
- Long-term roadmap: at the moment, ownCloud seems to be more focused on corporate customers and more relevant for investors, while Nextcloud seems to be more focused on extending “direct” user-to-user communication and cooperation features.

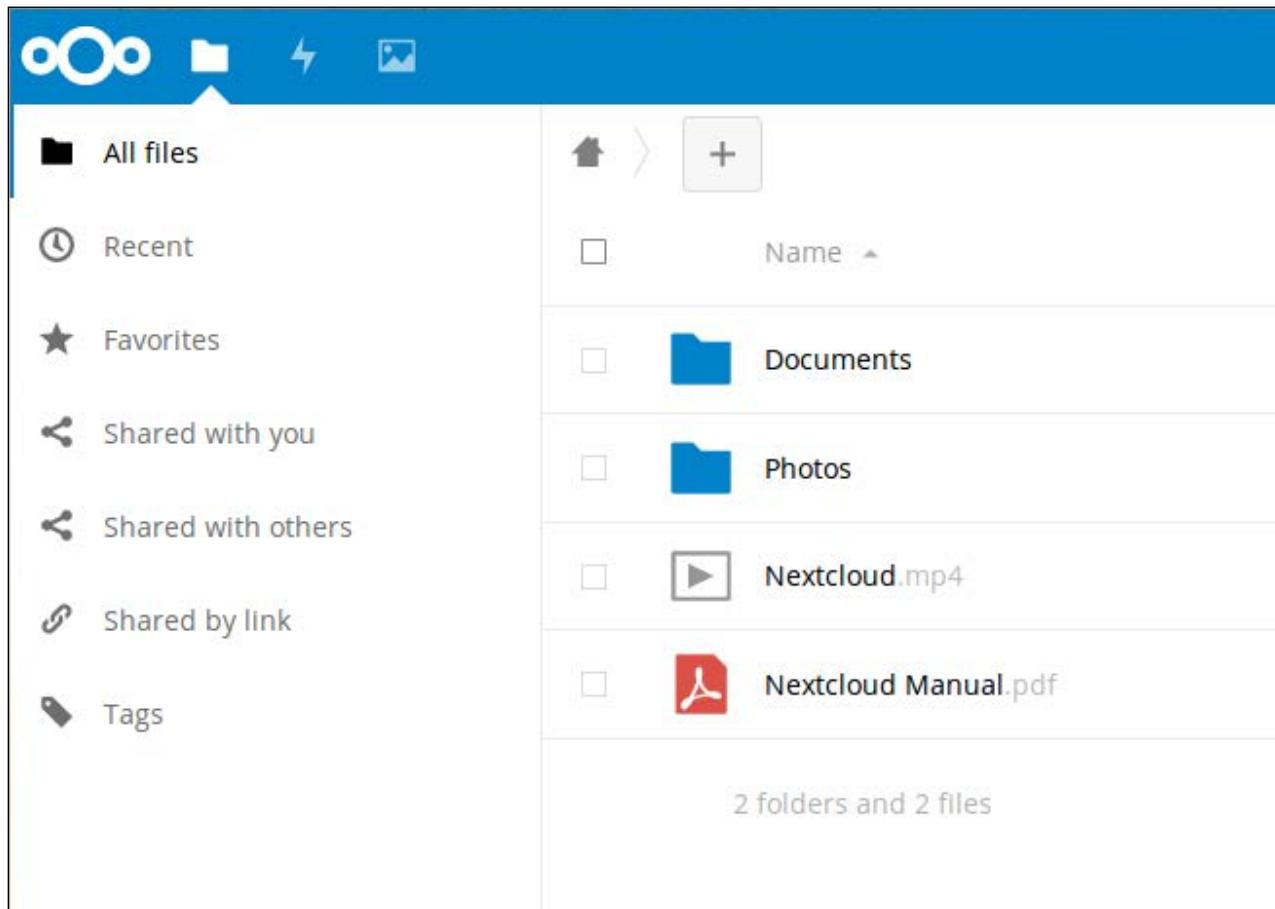


Figure 2. The Original Nextcloud/ownCloud Functions: File and Picture Storage, Dropbox-Style

A Word on Security

Several good reasons to choose Nextcloud as the online home for your own files and data are related to security. I don't cover them in detail in this introductory article, but I want to mention at least some of them.

Nextcloud refuses continuous (that is, malicious) attempts to authenticate from any computer, except those whose IP addresses are included in "brute-force IP whitelists". (Of course, the best possible whitelist you can configure is an empty one.)

Content Security Policy (CSP), instead, is the way a Nextcloud server may, for

General

Attribute to map the UID to.

Only allow authentication if an account exists on some other backend. (e.g. LDAP)

Use SAML auth for the Nextcloud desktop clients (requires user re-authentication)

Service Provider Data

If your Service Provider should use certificates you can optionally specify them here. Hide Service Provider settings ...

X.509 certificate of the Service Provider

Private key of the Service Provider

Identity Provider Data

Configure your IdP settings here.

Identifier of the IdP entity (must be a URI)

URL Target of the IdP where the SP will send the Authentication Request Message

Show optional Identity Provider settings...

Attribute mapping

If you want to optionally map attributes to the user you can configure these here. Hide attribute mapping settings ...

Attribute to map the displayname to.

Attribute to map the email address to.

Security settings

Figure 3. Configuring SAML for secure single-sign-on is a delicate process, but the Nextcloud interface makes it simple with plenty of instructions.

example, tell a browser “if you found this script in, or linked from, a page from me, do *not* trust it. It must have been injected there by some attacker!”

SAML (Security Assertion Markup Language) is an XML-based open standard for secure, single sign-on (SSO) to web-based applications across different, independent servers. Nextcloud 13 supports SSO with SAML natively through a dedicated app. If you log in to your own Nextcloud, you then can use any service, on any other SAML-enabled website for which you have access rights, without entering any more credentials.

Prerequisites

In order to install Nextcloud, you need basic Linux administration skills, familiarity with the command line and some patience. Software-wise, the Nextcloud server is a PHP application that needs a LAMP (Linux, Apache, MySQL, PHP) or similar software stack to work. You can install it on almost any box permanently connected to the internet, from bare metal in a server farm to ordinary web-hosting accounts, or even home-based minicomputers like the Raspberry Pi.

Nextcloud 13 can run in different environments, from shared hosting accounts to servers using nginx instead of Apache or as an Ubuntu snap package. The configuration officially recommended (quoting the website) “for the best compatibility, especially if you plan to use lots of plugins”, is Apache 2.4 or later, and a MySQL or MariaDB database. This is why I’m describing command-line installation of Nextcloud 13 server on a computer running Ubuntu 16.04 LTS, PHP 7, Apache2 and a MariaDB 10.0 database.

The procedure is relatively lengthy to explain, but it’s worth it. Nextcloud has many more features and options than what I describe here, and you can use it to store some of your most sensitive documents and data. Therefore, I strongly suggest that before actually exposing it on the internet, be sure to play with it locally on your home Linux box as much as you can, even if it means re-installing it from scratch several times.

And, there’s only one way to do all that testing efficiently: an installation method that can be entirely automated with a shell script.

Installation and Initial Configuration First, get *all* the necessary software, because Nextcloud 13 depends on several packages. In the case of Ubuntu 16.04, the ones you must install with **apt-get** are these:

```
sudo apt-get install apache2 mariadb-server
↳libapache2-mod-php7.0
sudo apt-get install php7.0-gd php7.0-json php7.0-mysql
↳php7.0-curl php7.0-mbstring
sudo apt-get install php7.0-intl php7.0-mcrypt php-imagick
↳php7.0-xml php7.0-zip
```

(Don't worry if some of those packages are already installed on your system, `apt-get` will just skip to the next one.)

After that, download the Nextcloud tarball from the website, unpack it, and copy it into its own folder under the Web server document root, which, in this example, is `/var/www/html/`:

```
tar -xjf nextcloud-13.0.0.tar.bz2
sudo cp -r nextcloud /var/www/html/
```

Preparing the Database and Web Servers

On Ubuntu 16.04 (and, likely, on most Ubuntu derivatives), the command-line installation of Nextcloud won't work unless there already is a MariaDB account that is not root, but does have all the privileges needed to create new users and databases. Here's how to create such an account, if needed, with name dbadmin and password dbadminpw (note that mdb is my own MariaDB prompt, not the default one):

```
sudo mysql -u root
mdb>use mysql;
mdb>CREATE USER 'dbadmin'@'localhost' IDENTIFIED BY 'dbadminpw';
mdb>GRANT ALL PRIVILEGES ON *.* TO 'dbadmin'@'localhost'
↳WITH GRANT OPTION;
mdb>FLUSH PRIVILEGES;
mdb>exit;
```

Apache, on the other hand, needs a dedicated configuration file, which on Ubuntu 16.04

is /etc/apache2/sites-available/nextcloud.conf, to handle Nextcloud properly. If your server is example.com, and you want your Nextcloud available at example.com/nextcloud, that file should look like this:

```
#####
Alias /nextcloud "/var/www/html/nextcloud/"

# the following two directives are needed for picoCMS

ProxyPass /nextcloud/sites/ http://localhost/nextcloud/
    ~index.php/apps/cms_pico/pico/
ProxyPassReverse /nextcloud/sites/ http://localhost/nextcloud/
    ~index.php/apps/cms_pico/pico/

<Directory /var/www/html/nextcloud/>
    Options +FollowSymlinks
    AllowOverride All

    <IfModule mod_dav.c>
        Dav off
    </IfModule>

    SetEnv HOME /var/www/html/nextcloud
    SetEnv HTTP_HOME /var/www/html/nextcloud

</Directory>
#####
```

Once that file is ready, type the following commands at the prompt to enable the modules that Apache also needs to handle Nextcloud:

```
sudo a2enmod rewrite
sudo a2enmod headers
```

```
sudo a2enmod env
sudo a2enmod dir
sudo a2enmod mime
sudo a2enmod proxy_http
```

Finally, here are the commands to type to make the Apache user own the Nextcloud files, enable the configuration files shown above and, finally, restart Apache:

```
sudo chown -R www-data:www-data /var/www/html/nextcloud/
sudo ln -s /etc/apache2/sites-available/nextcloud.conf
  ↳/etc/apache2/sites-enabled/nextcloud.conf
sudo service apache2 restart
```

Actually Installing Nextcloud

Once the Web and database servers are ready and the Nextcloud files are in place, the actual Nextcloud installation may happen entirely by pointing your browser (in the “local testing” phase I already recommended, at least) at <http://localhost/nextcloud>. As promised, however, I’m going to show you how to continue on the command line.

This is possible thanks to a PHP tool called **occ** (from “ownCloud console”) distributed with Nextcloud. To use **occ**, move to the nextcloud base directory, and then, using the Apache server account (www-data, in this example) to preserve the right permissions on files and folders, run it as follows:

```
cd /var/www/html/nextcloud/
sudo -u www-data php occ maintenance:install --database "mysql"
  ↳--database-name "mynextcloud" --database-user "dbadmin"
  ↳--database-pass "dbadminpw" --admin-user "nextcloudadmin"
  ↳--admin-pass "nextcloudadminpw"
```

If everything goes well, **occ** will exit with a “Nextcloud was successfully installed” message. At that point, you’ll finally be able to log in to Nextcloud at <http://localhost/nextcloud> with

the admin account (“nextcloudadmin”) and password “nextcloudadminpw”.

Using **occ**, you also can create users or enable previously downloaded Nextcloud apps, among other things. The **occ** equivalent of the GUI procedure for creating a user named marco in the mycloudusers group, with display name “Marco F”, is:

```
sudo -u www-data php occ user:add --display-name="Marco F"  
--group="mycloudusers" marco
```

Measuring and Optimizing Performances

Nextcloud 13 has a tab, shown in Figure 4, that gives the administrator a first, quick idea of how loaded it is. In order to avoid performance bottlenecks, the easiest solution seems to be the memory cache called OPcache. To enable it, follow the instructions in the Nextcloud

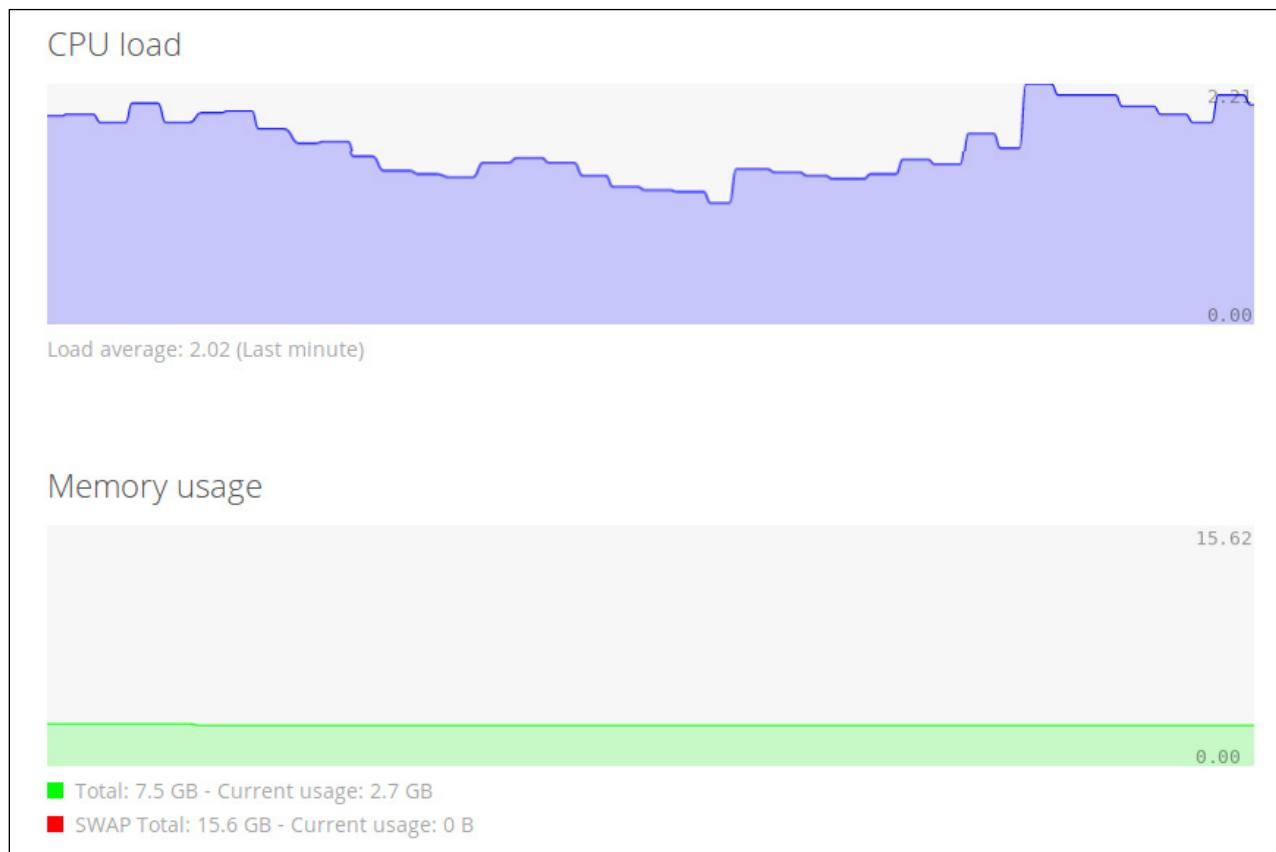
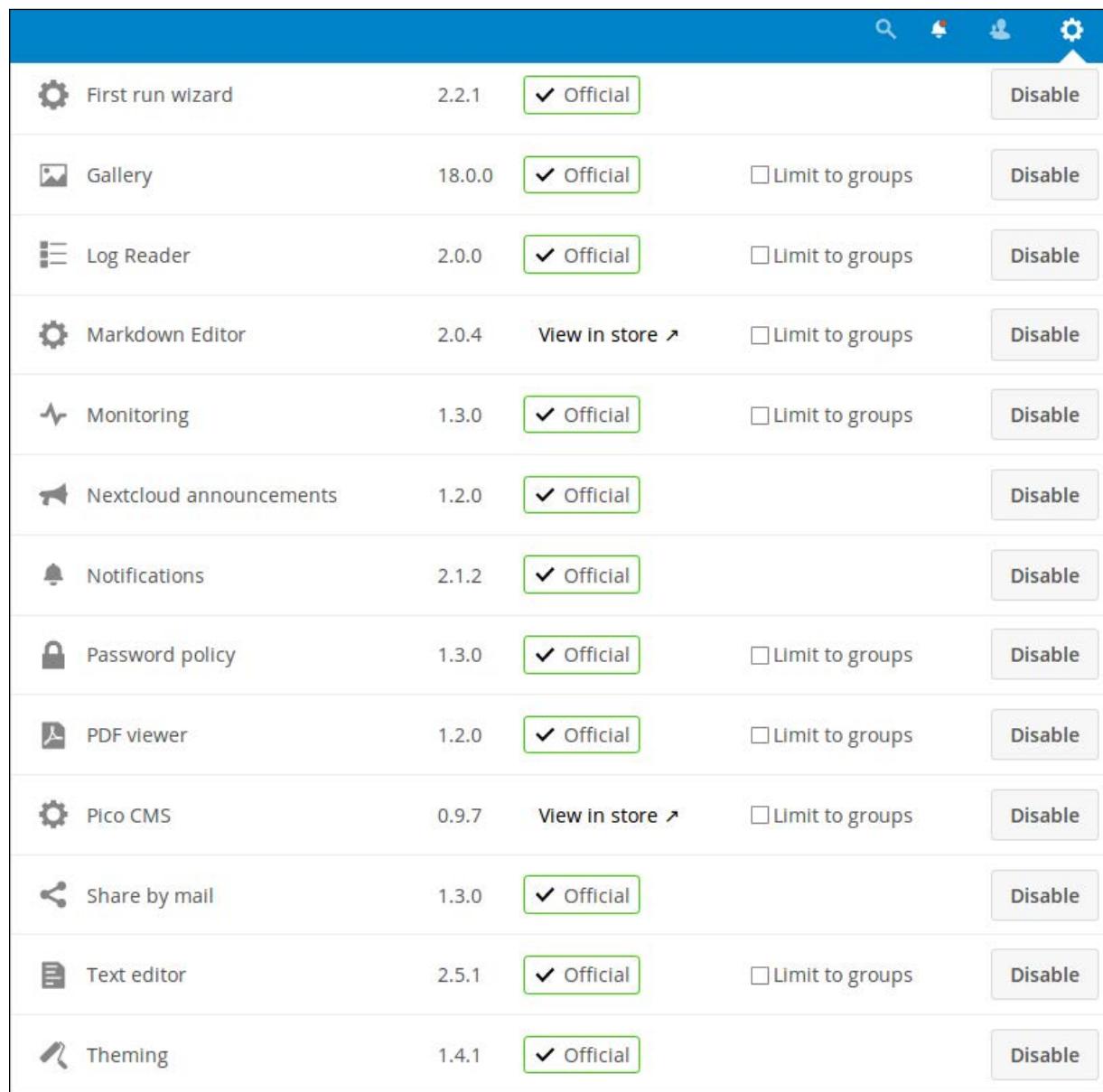


Figure 4. The Nextcloud 13 Real-Time CPU and Memory Load Monitors

Administration/Basic Settings tab. You also can install the Redis database for local caching and file locking. (For details, see “[Tuning Nextcloud for Optimal Performance](#)”.)

The Real Power of Nextcloud Is Its Apps

If Nextcloud were only a personal alternative to file-hosting services like Dropbox, it wouldn’t be such a big deal. Its real power, however, is in the many extensions, or “apps”,



The screenshot shows the Nextcloud Apps page with a list of installed applications. Each app entry includes its name, version, official status, and a 'Disable' button.

App	Version	Official	Limit to groups	Action
First run wizard	2.2.1	<input checked="" type="checkbox"/> Official		<button>Disable</button>
Gallery	18.0.0	<input checked="" type="checkbox"/> Official	<input type="checkbox"/> Limit to groups	<button>Disable</button>
Log Reader	2.0.0	<input checked="" type="checkbox"/> Official	<input type="checkbox"/> Limit to groups	<button>Disable</button>
Markdown Editor	2.0.4	View in store ↗	<input type="checkbox"/> Limit to groups	<button>Disable</button>
Monitoring	1.3.0	<input checked="" type="checkbox"/> Official	<input type="checkbox"/> Limit to groups	<button>Disable</button>
Nextcloud announcements	1.2.0	<input checked="" type="checkbox"/> Official		<button>Disable</button>
Notifications	2.1.2	<input checked="" type="checkbox"/> Official		<button>Disable</button>
Password policy	1.3.0	<input checked="" type="checkbox"/> Official	<input type="checkbox"/> Limit to groups	<button>Disable</button>
PDF viewer	1.2.0	<input checked="" type="checkbox"/> Official	<input type="checkbox"/> Limit to groups	<button>Disable</button>
Pico CMS	0.9.7	View in store ↗	<input type="checkbox"/> Limit to groups	<button>Disable</button>
Share by mail	1.3.0	<input checked="" type="checkbox"/> Official		<button>Disable</button>
Text editor	2.5.1	<input checked="" type="checkbox"/> Official	<input type="checkbox"/> Limit to groups	<button>Disable</button>
Theming	1.4.1	<input checked="" type="checkbox"/> Official		<button>Disable</button>

Figure 5. Work, entertainment, administration, sharing—Nextcloud apps can do a lot.

that provide many additional functions, often through extra buttons in Nextcloud's top bar. Figure 5 shows only a partial idea of how diverse the apps can be.

To use an app not shown in the administration interface, download and unpack it in the apps subfolder of your Nextcloud installation, then make the Apache user owner of its files. After that, you just need to enable the app, with **occ** or in the Nextcloud interface.

In the Nextcloud interface, you also can enable bundles of apps with one click or limit access to most apps to selected groups of users. The app bundles in Nextcloud 13 are Enterprise, Groupware, Social sharing and an “Education Edition”.

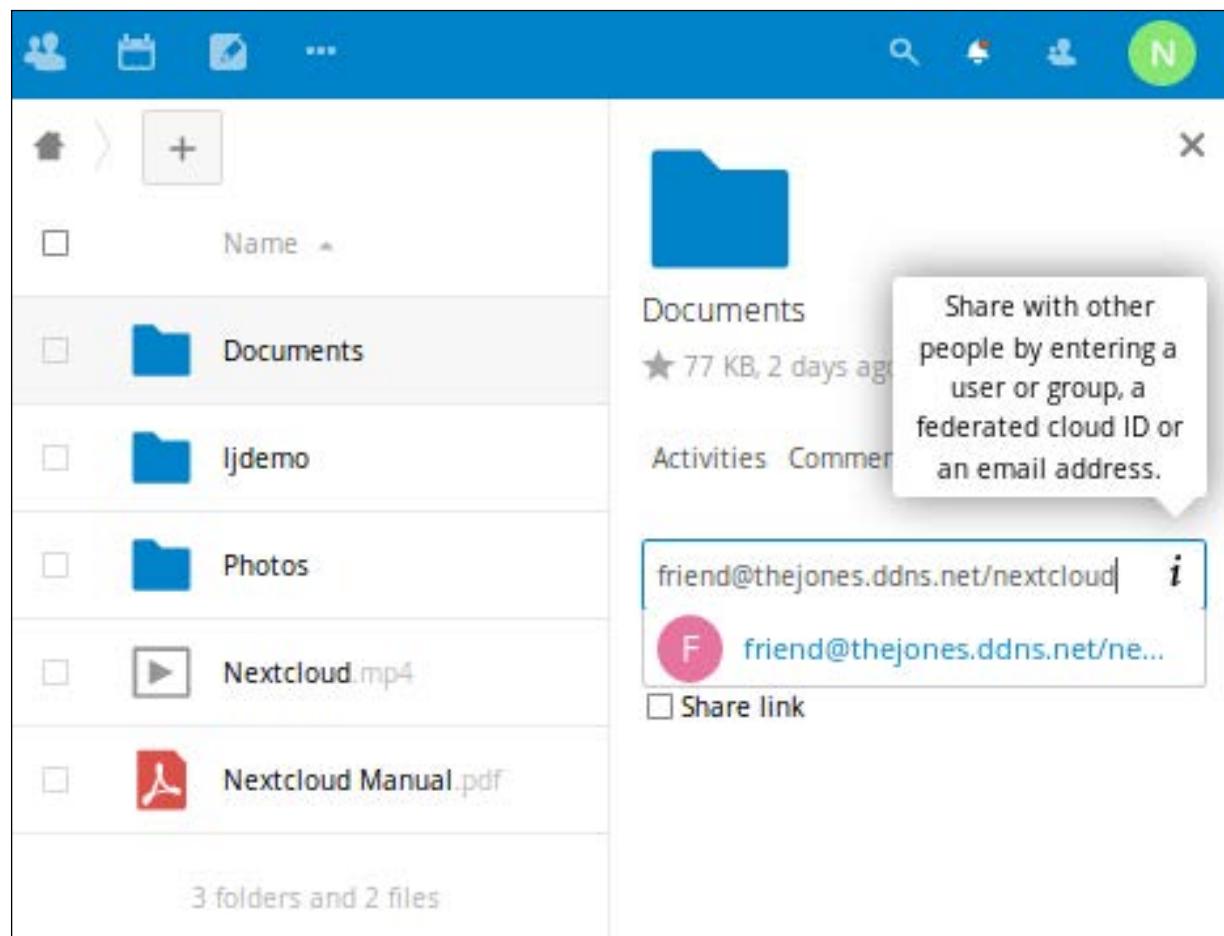


Figure 6. A detail of how you can share files and folders from your Nextcloud with any other user of other Nextcloud instances.

Beyond Files: Federation, Video Calls and Web Publishing Even if you need it only to host your files online, Nextcloud can do much more than provide a container for keeping those files. To begin with, all users of a Nextcloud server can share single files, or whole folders, with whomever they want by giving them a link, with or without an associated password. At the same time, a Nextcloud administrator easily can prevent single apps from sharing files and data, or it can allow file sharing only inside a group of users.

The really interesting thing, however, is “federation”. This name indicates the capability

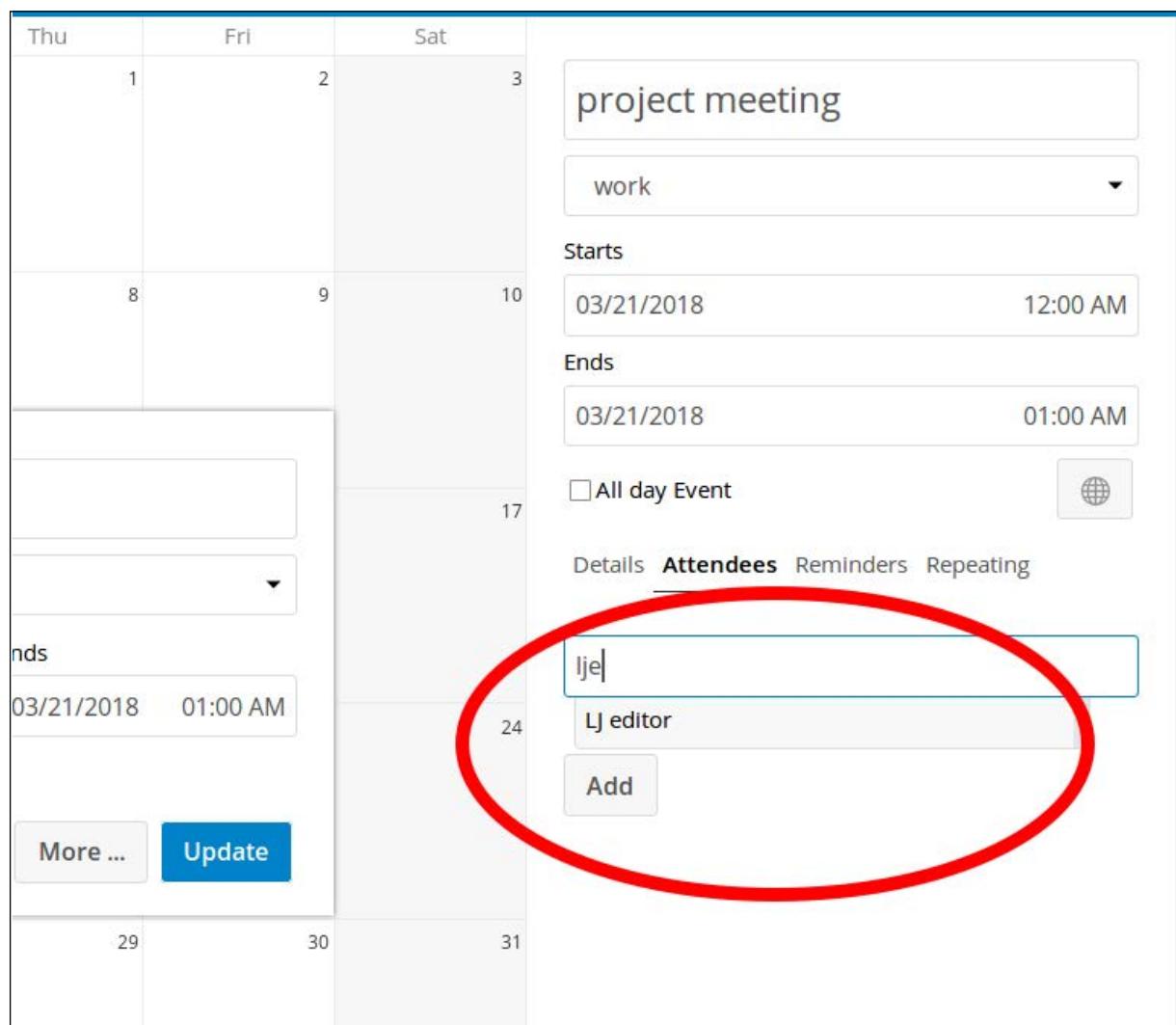


Figure 7. Nextcloud recognizes and auto-completes the addresses of all its users and those of any other federated Nextclouds.

to connect totally independent installations of this server in one, seamless “cloud of Nextclouds”. It is thanks to federation that, for example, all your relatives living in different states can see, each as a local folder of their own Nextcloud server, the same gallery of photographs that you host inside yours—even if that folder is not public and none of them has a user account on your server. Another common usage of federation is merging the user profiles of several servers in one common address book. This lets all those users find each other more easily, with their Nextcloud interface auto-completing the names of the other users when they start typing them.

Nextcloud’s federation-related features are accessible from the “Sharing” tab of the administration panel. From there, with a few clicks, you can define if and how users can share their own content with other Nextcloud servers, see the folders in those same servers or access a “global address book”.

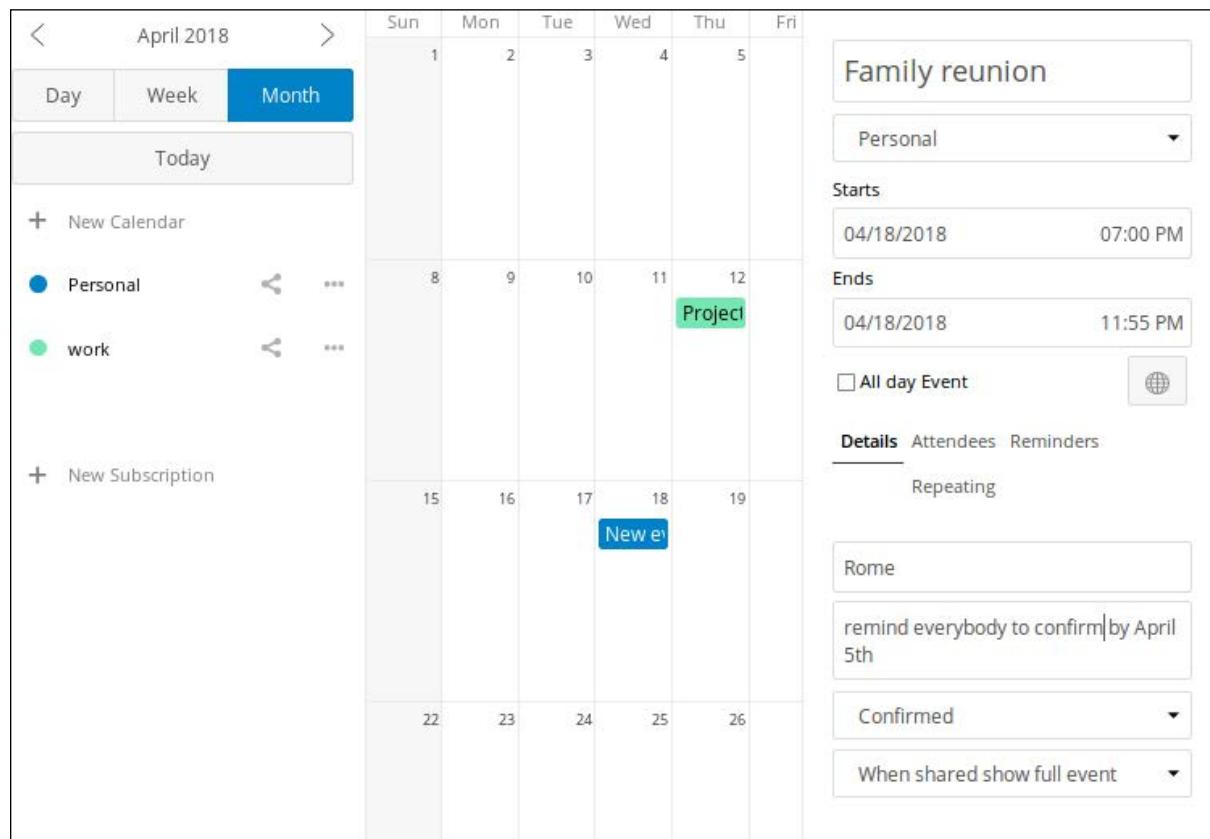


Figure 8. Scheduling appointments and inviting your fellow Nextcloud users? No problem.

That sharing of user directories can happen only with the servers that you declare “trusted” in the same tab. Synchronization of the local address book with those of the trusted servers happens with this `occ` command that you can put inside a cron job:

```
sudo -u www-data php occ federation:sync-addressbooks
```

Hey Nextcloud, Call My Mother

What’s the next step after easily sharing pictures with distant family members or documents with colleagues? Discussing them in an easy-to-use, privacy-friendly environment, of course.

Integration of the Calendar and users profiles of Nextcloud makes scheduling online meetings with them a snap. When the time comes, the Nextcloud Talk app lets you chat, make audio or video calls and share your screen, without installing any software (except, of course, a modern browser, or the Nextcloud Android or iOS apps, on one’s desktop or smartphone).

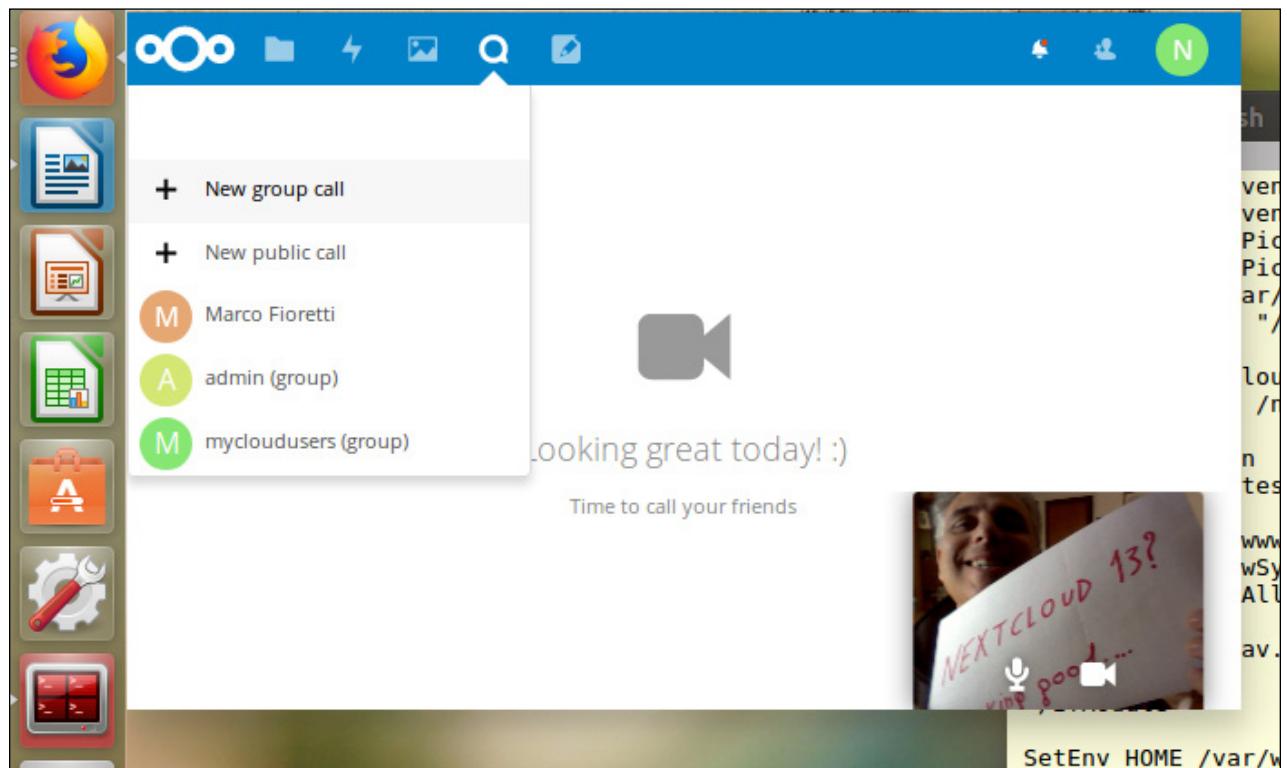


Figure 9. Video calls with integrated chats look really promising in Nextcloud 13.

Both chats and calls are peer-to-peer and end-to-end encrypted, without embedded advertising, or any central organization logging everything. Oh, and users get instant notifications, in their browsers or in the mobile apps, whenever other users want to talk with them, or have commented on some file they shared.

Now do you see why I say that Nextcloud, and its federation, may be the first step toward replacing proprietary platforms, from Dropbox to Skype?

Blogging with Nextcloud

Online self-publishing for the masses, via blogs or social networks, is one of the greatest features (and sometimes problems, of course) of the current, still open web. The Nextcloud 13 server provides an easy, if basic way to do this by integrating [picoCMS](#), the pico Content Management System.

picoCMS creates websites by rendering as HTML, with menus and all, all the Markdown plain-text files (with .md extension) that it finds inside some predefined folder. In Nextcloud, the best tool to edit .md files is the Markdown Editor app, so enable it if you decide to use picoCMS.

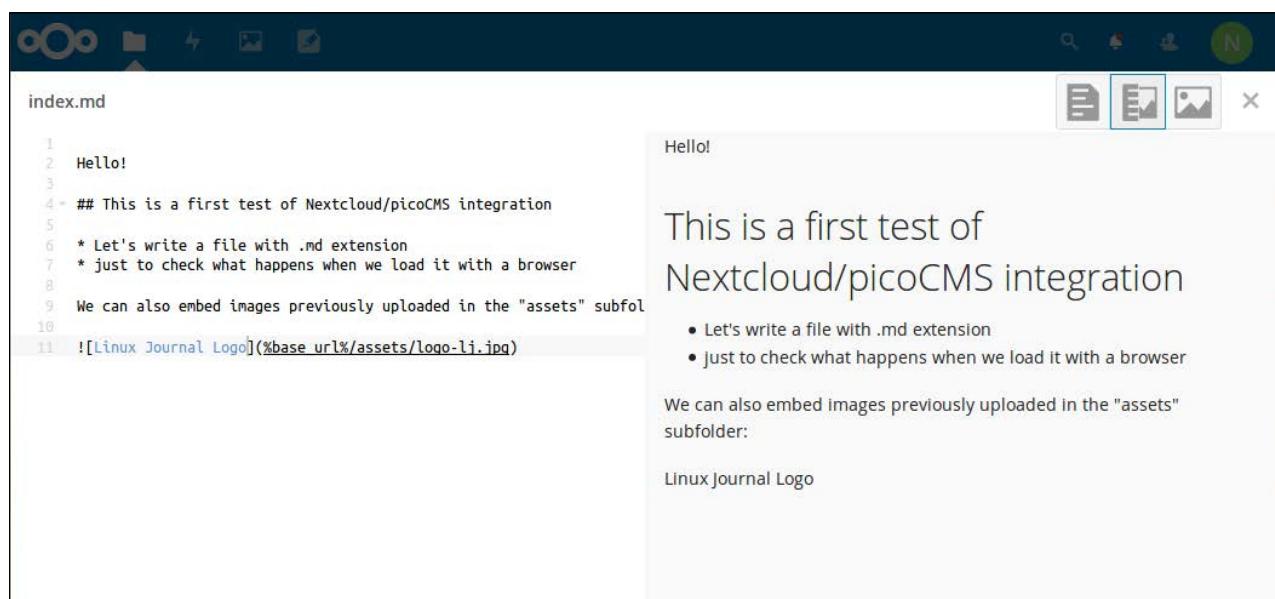


Figure 10. The Nextcloud Markdown editor, with its optional live preview of what you type.

Nextcloud users can independently define, in the picoCMS tab of the “Settings” interface, both the folder that contains the source files and the name of the website. Running on your own computer, the Apache configuration shown here would make Nextcloud serve the home page of a picoCMS website called “ljdemo” at the URL <http://localhost/nextcloud/sites/ljdemo/>.

To let all the users of your Nextcloud create inside it all the picoCMS websites they want, **download** the compressed archive of the app, and unpack it on the computer running Nextcloud. Then move the resulting folder (cms_pico) inside the apps subfolder of Nextcloud, change its permission, and enable it with these three commands:

```
sudo mv -i cms_pico          /var/www/html/nextcloud/apps/
sudo chown -R www-data:www-data /var/www/html/nextcloud/apps/cms_pico
sudo -u www-data php occ app:enable cms_pico
cms_pico enabled
```

(Of course, you even can put these commands into a script to make re-installations quicker!)

The next step is to tell the Apache Web server how to cooperate with picoCMS. The meaning of the two “ProxyPass” directives in the nextcloud.conf file already shown is this: “whenever a browser asks for an URL in the /nextcloud/sites/ subfolder, pass that URL to picoCMS, and then pass to the browser whatever you get in return”.

Note that those ProxyPass settings make picoCMS publish as websites only what it finds in certain folders of Nextcloud. They do not generate clean, short URLs for all the pages of those websites. To get that, you must adapt the **MOD_REWRITE** suggestions contained in the Administration picoCMS tab of the Nextcloud panel to your specific Apache configuration.

How to Write and Publish a Web Page with Nextcloud and picoCMS

Once it's up and running, publishing a web page in a Nextcloud/picoCMS environment is surely not as simple as it would be with systems like WordPress.

For example, the only way to add new Markdown files in any Nextcloud folder, except uploading them from your desktop, seems to be to copy and rename an already existing one. To insert a figure in a post, instead, you must separately upload it in the "asset" subfolder, and then point to it in the Markdown source, as shown below.

If these annoyances are not an issue for you, you may really like the Nextcloud/picoCMS flow. The Markdown editor and its integrated preview work great, and whatever you write instantly goes online. As a practical example, here's the source code, preview and rendering, at the local address <http://localhost/nextcloud/sites/ljdemo/testing/> of this index.md file placed in the Nextcloud folder ljdemo/content/testing/:

```
#####
```

Hello!

```
## This is a first test of Nextcloud/picoCMS integration
```

```
* Let's write a file with .md extension  
* just to check what happens when we load it with a browser
```

We can also embed images previously uploaded in the "assets" subfolder:

```
![Image Title](%base_url%/assets/logo-lj.jpg)
```

```
#####
```

Demo for Linux Journal

Welcome Sub Page Index

Sub Page

Hello!

This is a first test of Nextcloud/picoCMS integration

- Let's write a file with .md extension
- just to check what happens when we load it with a browser

We can also embed images previously uploaded in the "assets" subfolder:



The image is the official logo for Linux Journal, featuring the word "LINUX" in large, bold, white letters on a red background, with "JOURNAL" in smaller white letters below it. A small "TM" symbol is located at the top right of the "X".

Figure 11. This is what the first page of your Nextcloud/picoCMS website may look like.

What Next? A Lot!

Nextcloud seems to be a great platform for integrating online services of all kinds. In this article, I explained how to set it up and tried to provide an idea of its flexibility, but there is much more you could do with it. In future articles, I plan to cover how to integrate

with Nextcloud email, secure browsing with Let's Encrypt and collaborative editing with Etherpad. Stay tuned! ■

Resources

- [Nextcloud Source Installation Manual](#)
- [Official Nextcloud Apps Directory](#)
- [Using Nextcloud's Command Line](#)
- [PicoCMS Installation and Configuration Instructions](#)
- [Tuning Nextcloud for Optimal Performances](#)
- [The Nextcloud "Compare Cloud Technologies" Page](#)
- [ownCloud vs. Nextcloud: comparing cloud storage services \(February 2018\)](#)
- [Nextcloud vs ownCloud—the Whole Story \(February 2018\)](#)

Marco Fioretti is a free software user and author since 1995, board member of the Free Knowledge Institute and author of the Percloud proposal for a truly usable alternative to Facebook, Gmail and similar services.

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

The GDPR Takes Open Source to the Next Level

Richard Stallman will love the new GDPR.

By **Glyn Moody**

It's not every day that a new law comes into force that will have major implications for digital industries around the globe. It's even rarer when a such law will also bolster free software's underlying philosophy. But the European Union's **General Data Protection Regulation** (GDPR), which will be enforced from May 25, 2018, does both of those things, making its appearance one of the most important events in the history of open source.

Free software is famously **about freedom, not free beverages**:

"Free software" means software that respects users' freedom and community. Roughly, it means that the users have the freedom to run, copy, distribute, study, change and improve the software. Thus, "free software" is a matter of liberty, not price. To understand the concept, you should think of "free" as in "free speech," not as in "free beer".



Glyn Moody has been writing about the internet since 1994, and about free software since 1995. In 1997, he wrote the first mainstream feature about GNU/Linux and free software, which appeared in *Wired*. In 2001, his book *Rebel Code: Linux And The Open Source Revolution* was published. Since then, he has written widely about free software and digital rights. He has a [blog](#), and he is active on social media: @glynmoody on [Twitter](#) or [identi.ca](#), and +glynmoody on [Google+](#).



Richard Stallman's great campaign to empower individuals by enabling them to choose software that is under their control has succeeded to the extent that anyone now can choose from among a wide range of free software programs and avoid proprietary lock-in. But a few years back, Stallman realized there was **a new threat to freedom**: cloud computing. As he told The Guardian in 2008:

One reason you should not use web applications to do your computing is that you lose control. It's just as bad as using a proprietary program. Do your own computing on your own computer with your copy of a freedom-respecting program. If you use a proprietary program or somebody else's web server, you're defenseless. You're putty in the hands of whoever developed that software.

Stallman pointed out that running a free software operating system—for example Google's ChromeOS—offered **no protection against this loss of control**. Nor does requiring the cloud computing service to use the **GNU Affero GPL** license solve the problem: just because users have access to the underlying code that is running on the servers does not mean they are in the driver's seat. The real problem lies not with the code, but elsewhere—with the data.

OPEN SAUCE

Running free software on your own computer, you obviously retain control of your own data. But that's not the case with cloud computing services—or, indeed, most online services, such as e-commerce sites or social networks. There, highly personal data about you is routinely held by the companies in question. Whether or not they run their servers on open-source code—as most now do—is irrelevant; what matters is that they control your data—and you don't.

The new GDPR changes all that. Just as free software seeks to empower individuals by giving them control over the code they run, so the GDPR empowers people by giving them the ability to control their personal data, wherever it is stored, and whichever company is processing it. The GDPR will have a massive impact on the entire online world because **its reach is global**, as this EU website on the subject explains:

The GDPR not only applies to organisations located within the EU but it will also apply to organisations located outside of the EU if they offer goods or services to, or monitor the behaviour of, EU data subjects. It applies to all companies processing and holding the personal data of data subjects residing in the European Union, regardless of the company's location.

And if you think that the internet giants based outside the EU will simply ignore the GDPR, think again: under the legislation, companies that fail to comply with the new regulation can be fined up to 4% of their global turnover, wherever they are based. Google's total turnover last year was **\$110 billion**, which means that non-compliance could cost it \$4.4 billion. Those kinds of figures guarantee that every business in the world that has dealings with EU citizens anywhere, in any way, will be fully implementing the GDPR. In effect, the GDPR will be a privacy law for the whole world, and the whole world will benefit. According to a report in the Financial Times last year, the top 500 companies in the US alone will spend **\$7.8 billion in order to meet the new rules** (paywall). The recent scandal over **Cambridge Analytica's massive collection of personal data** using a Facebook app is likely to increase pressure globally on businesses to strengthen their protections for personal data for everyone, not just for EU citizens.

The **GDPR's main features** are as follows. Consent to data processing “must be

clear and distinguishable from other matters and provided in an intelligible and easily accessible form, using clear and plain language. It must be as easy to withdraw consent as it is to give it.” Companies will no longer be able to hide bad privacy policies in long and incomprehensible terms and conditions. The purpose of the data processing must be clearly attached to the request for consent, and withdrawing consent must be as easy to do as giving it.

There are two important rights in the GDPR. The “right to access” means people are able to find out from an organization whether or not personal data concerning them is being processed, where and for what purpose. They must be given a copy of the personal data, free of charge, on request. That data must be in a “commonly used” and machine-readable format so that it can be easily transferred to another service. The other right is to data erasure, also known as the “right to be forgotten”. This applies when data is no longer relevant to the original purposes for processing, or people have withdrawn their consent. However, that right is not absolute: the public interest in the availability of the data may mean that it is not deleted.

One of the innovations of the GDPR is that it embraces “[privacy by design and default](#)”. That is, privacy must be built in to technology from the start and not added as an afterthought. In many ways, this mirrors free software’s insistence that freedom must suffuse computer code, not be regarded as something that can be bolted on afterward. The original [Privacy by Design framework](#) explains what this will mean in practice:

Privacy must become integral to organizational priorities, project objectives, design processes, and planning operations. Privacy must be embedded into every standard, protocol and process that touches our lives.

Open-source projects are probably in a good position to make that happen, thanks to their transparent, flexible processes and feedback mechanisms. In addition, under the GDPR, [computer security](#) and encryption gain a heightened importance, not least because there are new requirements for “breach notifications”. Both [the relevant authorities](#) and [those affected](#) must be informed rapidly of any breach. Again, open-

OPEN SAUCE

source applications may have an advantage here thanks to the ready availability of the source code that can be examined for possible vulnerabilities. The new fines for those who fail to comply with the breach notifications—up to 2% of global turnover—could offer an additional incentive for companies to require open-source solutions so that they have the option to look for problems before they turn into expensive infractions of the GDPR.

It would be hard to overstate the importance of the GDPR, which will have global ramifications for both the privacy sector in particular and the digital world in general. Its impact on open source is more subtle, but no less profound. Although it was never intended as such, it will effectively address the key problem left unresolved by free software: how to endow users with the same kind of control that they enjoy over their own computers, when they use online services. As a result, May 25, 2018 should go down as the day when the freedom bestowed by open source went up a notch. ■

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