FREE DVD

PRIVACY
Stay free with these anonymity solutions

NOW FEATURING
LINUXVOICE

PRO LINUX MAGAZINE

LINUX PRO MAGAZINE

MARCH 2017

PRIVACY

Safe surfing with Tails

Signal
Snowden's choice for a messenger app

Set Up Amazon Web Services

Bitmessage
Private communication without an email server

Linux Foundation's Jim Zemlin:
"Great developers are like poets."

KDE Connect
Linking Android with the KDE desktop

Expose
Easy and powerful photo gallery

LINUXVOICE

- What will rock in 2017
- IPv6 on Linux
- Lineage OS: a freer Android

WELCOME LINUXVOICE READERS! p. 63

FOSSPicks
- Make music with Radium
- PowerShell on Linux

Tutorials
- Build the Linux kernel
- Inav: Analyze logfiles

Issue 196
Mar 2017
US$ 15.99
CAN$ 17.99

HETZNER ONLINE

Network security Firewall.

$0 Firewall free of charge

intel XEON inside

e.g. Dedicated Root Server PX61-NVMe

Intel® Xeon® E3-1275 v5
Quad-Core Skylake Processor
64 GB DDR4 ECC RAM
2 x 512 GB NVMe Gen3 x4 SSD
Guaranteed 1 Gbit/s bandwidth
100 GB Backup Space
30 TB traffic inclusive*
No minimum contract
Setup Fee $128.00

monthly $ 64

* There are no charges for overage. We will permanently restrict the connection speed if more than 30 TB/month are used. Optionally, the limit can be permanently cancelled by committing to pay $1.30 per additional TB used.

Free Firewall for Your Dedicated Root Servers!

Hetzner Online's stateless firewall is a free security solution for your dedicated root server. Starting now on the customer interface Robot, you can use the firewall feature to define your own filtering settings for traffic, such as the originating IPv4 address or TCP/UDP sender port. With this feature, Hetzner Online helps you protect your dedicated root server from Internet dangers. **And it is naturally free of cost.**

www.hetzner.de/us

WILL SOMEONE?

Dear Reader,

One of the more interesting news items that crossed my desk – or my desktop, as in Mate, since loose papers don't really fly through my office like they used to, was a quote from Hewlett-Packard Enterprise's CEO Meg Whitman, "AI and robots? Will someone think of the jobs?"

The quote came from a speech Whitman gave at the World Economic Forum in Davos Klosters, Switzerland. I wish I could find a full transcript of the speech, but so far it doesn't seem to be out there. Lots of videos are available at the World Economic Forum website [2] – and probably more will be there by the time you read this – so if you look around enough, you might run across it.

The Register and other commentators reporting on the speech have noted the irony of Whitman making these remarks, since Hewlett-Packard Enterprise has laid off 90,000 people since Whitman took over. I don't really have a solution to the problem Whitman raises, which is the displacement of workers due to automation and artificial intelligence, and if I did, I'm pretty sure it wouldn't fit on this page. All I can really do in this column is point things out, and a couple things about this story strike me as interesting.

First, it has always seemed strange that all these predictions of the techno apocalypse seem so passive – no one really has an answer for what to do about them. "*Someone* had better do something about this," is a common theme, but no one says "*I'm* going to do something about this." I would, of course, have to include myself in this category of people who talk but don't really have a roadmap, but at least I am willing to register some alarm and don't attempt to soft-pedal this potential world fiasco as yet another emerging business trend.

Second, it seems a testament to our politically polarized world that only our allies get to state the obvious. Yes, Meg Whitman has laid off lots of people, but that does not seem particularly relevant to the substance of her warnings. The head of an oil company might not be the ideal person to raise awareness about climate change, but if an oil executive said we'd better do something about global warming, that would actually be a quite a welcome development.

I don't have a solution for what to do about the displacement of jobs due to automation, but I have a sense that Free Software is probably helping somehow because the problem seems related to the concentration of wealth, and Free Software is a counter-balancing influence to the concentration of wealth.

I agree with Ms. Whitman's observation that we'd better pay attention to the effect automation is having on our economy, and I have a feeling the solution to this problem is going to be much more radical and disruptive than anyone is talking about right now. So wake up everybody.

Joe Casad,
Editor in Chief

INFO

[1] "AI and Robots? Will Someone Think of the Jobs?" by Paul Kunert: *http://www.theregister.co.uk/2017/01/17/ai_hpe_ceo_whitman_job_irony/*

[2] World Economic Forum: *https://www.weforum.org/events/world-economic-forum-annual-meeting-2017*

LINUX MAGAZINE

WHAT'S INSIDE

This month you'll learn about Tails – the easiest path to the Tor anonymity network – and you'll discover some tools for private communication, including Signal and Bitmessage.

Other Highlights:

- **KDE Connect** – a cool tool for syncing your phone with the KDE desktop (page 48).
- **Amazon Web Services** run your homegrown programs in Amazon's popular cloud (page 52).

Plus plenty more in LinuxVoice, including a look at what's ahead for the Linux faithful in 2017 (page 63).

SERVICE

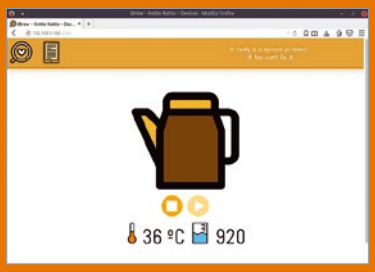
NEWS

COVER STORIES

SEE PAGE 6 FOR DETAILS

IN-DEPTH

LINUXVOICE



On the DVD

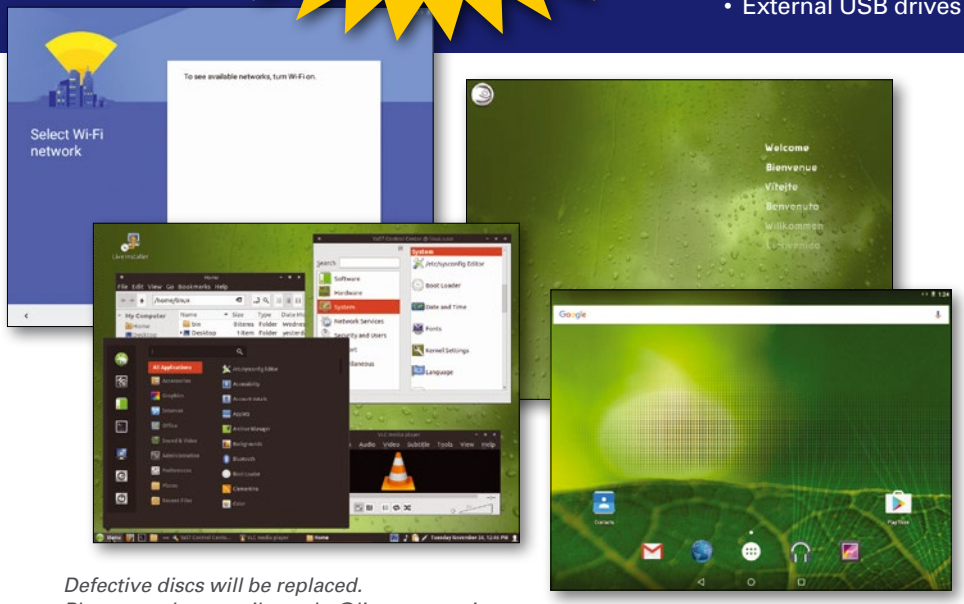
GeckoLinux (64-bit Live): Linux for Detail-Oriented Geckos

GeckoLinux is an openSUSE spin with a focus on out-of-the-box usability. On this DVD, you'll find the Static edition (based on openSUSE Leap 42.2) with an optimized Cinnamon desktop environment (the password is *linux*). Unlike openSUSE, which includes a non-Live installer that requires you know how to install patterns and packages for different desktop environments, GeckoLinux comes as an offline installable Live DVD/USB image, and whereas openSUSE omits patent-restricted codecs and drivers, GeckoLinux includes proprietary media codecs. Other refinements include improved font rendering and the ability to uninstall desktop programs with all their dependencies without the fear of the packages being re-installed automatically.

Android-x86 (32-bit Live): Run Android on Your PC

The Android-x86 project has ported the Android open source software stack for mobile devices to the x86 platform. The version on your DVD is the first stable release of the Android Marshmallow-MR2 release with the latest security updates. Other highlights include:

- Kernel 4.4.20.
- 3D graphics library update to Mesa 12.0.2.
- HDMI audio support.
- Flash-friendly filesystem (F2FS) support
- Multitouch, audio, WiFi, Bluetooth, sensor, camera, and Ethernet support.
- External USB drives and SD cards automounted.



Defective discs will be replaced.
Please send an email to subs@linux-magazine.com.

ADDITIONAL RESOURCES

[1] GeckoLinux: *https://geckolinux.github.io*

[2] GeckoLinux release notes (Static): *https://groups.google.com/forum/#!topic/geckolinux-updates/q3k0Dli10T8*

[3] GeckoLinux project wiki: *https://github.com/geckolinux/geckolinux-project/wiki*

[4] Android-x86 project: *http://www.android-x86.org*

[5] Android-x86 6.0-r1 release notes: *http://www.android-x86.org/releases/releasenote-6-0-r1*

DrupalCon Baltimore

The Drupal community is one of the largest open source communities in the world. We're developers, designers, strategists, coordinators, editors, translators, and more. Each year, we meet at DrupalCamps, meetups, and other events in more than 200 countries. But once a year, our community comes together in a U.S. city for our largest gathering, **DrupalCon**. This year we're going to Baltimore!

Tickets on sale now

https://events.drupal.org/baltimore2017

Important Dates

Call for Papers closes: February 1

Early Bird ticket pricing ends: March 3

Schedule published: March 15

NEWS

Updates on technologies, trends, and tools

THIS MONTH'S NEWS

Red Hat Releases RHEL 6.9 Beta

As the world moves from server-client to cloud-mobile, Red Hat is bringing cloud capabilities to one of the older releases of Red Hat Enterprise Linux (RHEL). Red Hat has released RHEL 6.9 Beta, which supports the next generation of cloud-native applications through an updated RHEL 6 base image.

Red Hat said in a press release, "The Red Hat Enterprise Linux 6.9 Beta base image enables customers to migrate their existing Red Hat Enterprise Linux 6 workloads into container-based applications – suitable for deployment on Red Hat Enterprise Linux 7, Red Hat Enterprise Linux Atomic Host, and Red Hat OpenShift Container Platform."

Since RHEL subscriptions are not locked into any particular release, customers can easily upgrade their infrastructure from RHEL 6 to RHEL 6.9 at any given time.

From the security point of view, RHEL 6 adds TLS 1.2 support to the GnuTLS component, which allows customers to use RHEL 6 with future revisions of security standards that may require TLS 1.2 support.

RHEL 6.9 beta is available immediately for testing.

SUSE Working on a New Operating System Called MicroOS

Cloud and containers are the next frontier for Linux companies. Responding to Container OS, Project Atomic, and Snappy Core, SUSE is working on MicroOS. The new operating system by SUSE is based on SUSE Enterprise Linux and focuses on delivering microservices.

In an exclusive interview with The New Stack, SUSE's newly appointed CTO, Dr. Thomas Di Giacomo, said that it will help those customers who are running legacy systems but want to migrate to modern technologies over time. "We want to make sure that companies that have legacy infrastructure and legacy applications that can move to modern technologies, where container as a service is offered through that OS itself."

One of the core components of MicroOS is transactional updates, which use the snapshot capabilities of Btrfs. All updates will be installed automatically, and a reboot will switch the system to latest packages. If anything fails, it will roll back to the older working version. The beta version of the project is expected to be released in March; the final release is expected in June.

Critical Security Holes Found in PHP 7

IT security firm, Check Point, has found serious vulnerabilities in PHP 7. Check Point has analyzed the code of PHP 7 to look into any vulnerabilities, especially "the unserialize mechanism" that was heavily exploited in PHP 5 that compromised platforms like Magento, vBulletin, Drupal, Joomla, etc.

What they found was not encouraging, Check Point wrote in a blog post: "Throughout our investigation we discovered 3 fresh and previously unknown vulnerabilities (CVE-2016-7479, CVE-2016-7480, CVE-2016-7478) in the PHP 7 unserialize mechanism. These vulnerabilities can be exploited using a technique we've discussed back in August."

The first two vulnerabilities, according to Check Point, give attackers complete control over servers. The third can create a DoS (Denial of Service) attack, which exhausts the memory consumption of the target site and shuts it down.

The PHP team was informed of the vulnerabilities in August and September. The fix for two vulnerabilities was released on October 13 and December 1. Users are advised to ensure they are running the latest version of PHP.

Check Point has issued IPS signatures for these vulnerabilities to protect users from possible attacks.

New Android Malware Found

Security researchers at Kaspersky Labs have discovered a new malware that affects Android devices. Nikita Buchka wrote on a blog post: "Instead of attacking a user, it attacks the WiFi network the user is connected to, or, to be precise, the wireless router that serves the network."

The trojan deploys the brute-force attack to guess the password and access the device. Once the password is cracked, it modifies the DNS server in the router, redirecting all traffic through their own servers and malicious websites.

What makes things really bad is, as Buchka explained, that instead of affecting users, the malware affects the entire network, which means every user on that network is exposed. Kaspersky recommends checking the DNS settings of your router.

There are currently two versions of the app: one is a fake mobile client for Chinese search engine Baidu and the second one is about WiFi network. It's the same old story where cybercriminals are offering malicious fake apps outside of official app stores. Always use the official app stores. Anyone using official Google Play Store for app installation is safe.

The lesson here is: don't install random apps from random websites.

Serious Bug Found in Ubuntu

An Irish security researcher Donncha O'Cearbhaill found a remote execution bug in Ubuntu's Apport crash reporter that can infect a system with malicious code.

O'Cearbhaill wrote on his blog, "The bug allows for reliable code injection when a user simply opens a malicious file. The following video demonstrates the exploit opening the Gnome calculator. The executed payload also replaces the exploit file with a decoy zip file to cover its tracks."

O'Cearbhaill reports that Ubuntu will open any unknown file with apport-gtk if it begins with ProblemType . What makes things worse is that Apport is installed by default on all Ubuntu systems after 12.10, which also includes forks like Linux Mint.

If you are using any Ubuntu-based distribution, you are vulnerable. The hole has been patched, but it does expose one major problem with Linux: Often such bugs hide for years and even decades, and security experts often lack incentives for finding them. Unlike Google, which rewards such discoveries, Linux vendors often depend on the community.

Commercial Linux distributions like Ubuntu should start a reward program to encourage security researchers to find such bugs. Without enough eyes, all bugs are deep.

If you are using any Ubuntu-based distribution, please update your system immediately.

MORE ONLINE

Linux Magazine

www.linux-magazine.com

Off the Beat • Bruce Byfield

LibreOffice MUFFIN risks being half-baked
On December 21, The Document Foundation announced that LibreOffice 5.3 would include MUFFIN (My User Friendly & Flexible INterface).

How Signal does security right
A couple of weeks ago, I was writing about Echo Whisper Systems' Signal, which encrypts voice and text messages for Android and iOS phones.

Taking a stand for ethical tech
Several weeks ago, I discussed taking a stand against unethical parts of your work.

Paw Prints • Jon "maddog" Hall

LPIC OT DevOPs Engineer – Request for help in the Job Task Analysis
Some of my readers may know that I am the Chair for the Board of Directors of the Linux Professional Institute (LPI).

ADMIN HPC

http://hpc.admin-magazine.com/

Modern Fortran – Part 2 • Jeff Layton
Fortran 90 catapulted Fortran from a perceived "old" language to a modern language on equal footing with any other.

ADMIN Online

http://www.admin-magazine.com/

Hyper-V containers with Windows Server 2016 Nils Kaczenski
The release of Windows Server 2016 also heralds a new version of Hyper-V, with improved cloud security, flexible virtual hardware, rolling upgrades of Hyper-V clusters, and production checkpoints.

A script for strict packet filter updates Matthias Wübbeling
Automatically create restrictive rules in Linux iptables packet filters.

Writing SELinux modules • Thorsten Scherf
Much has happened in the field of SELinux in the last few years, including the development of new usability features.

Setting up Windows clients with Chef Tam Hanna
Chef administrators unafraid of a learning curve can employ a powerful tool for Windows client management.

Zack's Kernel News

Chronicler Zack Brown reports on the latest news, views, dilemmas, and developments within the Linux kernel community.

By Zack Brown

ZACK BROWN

The Linux kernel mailing list comprises the core of Linux development activities. Traffic volumes are immense, often reaching 10,000 messages in a week, and keeping up to date with the entire scope of development is a virtually impossible task for one person. One of the few brave souls to take on this task is **Zack Brown**.

When to Use a Filesystem Capability

Michael Kerrisk wanted to address the need for developers to know which filesystem capability to associate with new features that they want to add to the kernel. This topic has traditionally been a subject of much confusion. There hasn't been enough documentation, and the POSIX standards bodies never really nailed things down sufficiently, so it's a bit of a mess. According to the Linux man page on capabilities, "traditional UNIX implementations distinguish two categories of processes: privileged processes (whose effective user ID is 0, referred to as superuser or root), and unprivileged processes (whose effective UID is nonzero). Privileged processes bypass all kernel permission checks, while unprivileged processes are subject to full permission checking based on the process's credentials (usually: effective UID, effective GID, and supplementary group list).

"Starting with kernel 2.2, Linux divides the privileges traditionally associated with superuser into distinct units, known as capabilities, which can be independently enabled and disabled."

Michael posted some documentation that, after some back-and-forth with Casey Schaufler, read:

When adding a new kernel feature that should be governed by a capability, consider the following points.

- *The goal of capabilities is to divide the power of superuser into pieces, such that if a program that has one or more capabilities is compromised, its power to do damage to the system is less than that of the same program running with root privilege.*
- *You have the choice of either creating a new capability for your new feature, or associating the feature with one of the existing capabilities. In order to keep the set of capabilities to a manageable size, associating a feature with an existing capability is preferable, unless there are compelling reasons to create a new one. (You also face a technical limit: the size a capability sets is currently limited to 64 bits.)*
- *To determine which existing capability might best be associated with your new feature, review the list of capabilities above in order to find a "silo" into which the new feature best fits. One approach is to determine if there are other features requiring capabilities that will always be used along with the new feature. If the new feature is useless without these other features, you should use the same capability as the other features.*
- *Don't choose* `CAP_SYS_ADMIN` *if you can possibly avoid it! A vast proportion of existing capability checks are associated with this capability, to the point where it can plausibly be called "the new root." Don't make the problem worse. The only new features that should be associated with* `CAP_SYS_ADMIN` *are ones that closely match existing uses in that silo.*
- *If you have determined that it really is necessary to create a new capability for your feature, don't make or name it as a "single-use" capability. Thus, for example, the addition of the highly specific* `CAP_PACCT` *was probably a mistake. Instead, try to identify and name your new capability as a broader silo into which other related future use cases might fit.*

Casey disagreed with Michael's admonition not to use `CAP_SYS_ADMIN` unless absolutely necessary. Casey felt that anything to do with system administration belonged with that capability. But Michael replied, "To me, the `CAP_SYS_ADMIN` situation is a terrible mess. Around a third of all of the capability checks in the kernel are for that capability. Or, to put it another way, it is so broad, that if a process has to have that capability, it may as well be root. And because it is so broad, the number of binaries that might need that file capability is large."

Michael also offered an incomplete list of all the abilities currently associated

with CAP_SYS_ADMIN and said that Casey would need a very broad definition of "system administration" in order to truly include all those abilities in that bailiwick.

Casey replied:

Back in the days of the POSIX P1003.1e/2c working group, we struggled with what to do about the things that required privilege but that were not related to the enforcement of security policy. Everyone involved was looking to use capabilities to meet B2 least-privilege requirements in NSA security evaluations. Because those evaluations were of security policy, by far the easiest thing to do was to create a single capability for all the things that didn't show up in the security policy and declare that the people doing the evaluation didn't have to look over there. Since then, people have taken a more practical view that includes security relevance in addition to security policy.

In retrospect, we should have grouped all of the attribute changes (chmod, chown, …) into one capability and broken the non-policy actions into a set of 2 or three.

The way that we think of privilege has evolved. We're not focused on policy the way we used to be. We'll never get everyone to agree on what the right granularity and grouping is, either.

Michael found that bit of history fascinating. But at this point, the discussion veered off into the question of finding the most intuitive names for each capability, and the conversation petered out.

Sometimes it feels as though the history of operating system design does as much to hold back its proper implementation as it does to advance it. There's so much room for discussion at all levels of an issue, and at the same time, there's a world of hardware that's constantly changing to suit a human market that is truly bizarre. It's amazing that something like filesystem capabilities has any rhyme or reason at all.

Cleaning Out FBDev Drivers

The once cutting-edge fbdev drivers have been sinking further and further into the backwaters of the kernel. Recently Tomi Valkeinen posted a patch to remove them from the staging area of Linux entirely. His reasoning was that all new display drivers should be using the DRM framework, and the FBDev drivers have been in maintenance mode, with no new drivers or major features coming down the pike. It was time to get rid of them! Specifically the xgifb, sm750fb, and fbtft drivers.

Daniel Vetter agreed wholeheartedly, remarking that, "we have the simple pipe helpers in drm-kms, and a few drivers starting to use them; there's really no reasons left anymore to have fbdev drivers." And Tomi agreed.

Geert Uytterhoeven wanted to see an example of a DRM driver that used the simple pipe helper, and Greg Kroah-Hartman also said that it only made sense to remove those remaining FBDev drivers if there were DRM replacements that worked on the same hardware.

But Tomi asked what it meant for code to be in the staging directory of the kernel. If it was the same as being out-of-tree, but just with greater accessibility via the Git repository, then it made sense to remove any FBDev driver for the same reasons no new ones would be added.

But Greg reiterated the need for Linux to continue to support existing hardware. He had no objection to keeping the FBDev drivers in the staging directory until it was safe to remove them, but he neither wanted to migrate them into the kernel proper, nor remove them from the tree entirely. Instead, Greg said, they should remain in staging until suitable replacements could be written.

Meanwhile Daniel replied to Geert, saying that the simple DRM drivers still hadn't appeared, although there were some projects "floating around in various places".

At the same time, Benjamin Herrenschmidt had his own objections to ditching the FBDev drivers. He said, "DRM drivers don't strike me as suitable for small/slow cores with dumb framebuffers or simple 2D only accel, such as the one found in the ASpeed BMCs. With drmfb, you basically have to shadow everything into memory & copy over everything, which locks you out of simple 2D accel. For a simple text console, the result is orders of magnitude slower and more memory hungry than a simple fbdev." And he added, "Not everything has a powerful 3D GPU."

But Tomi replied that if DRM was too heavy-weight, it should be fixed to be better. That wasn't a justification for leaving old, crufty FBDev drivers in the kernel forever.

Daniel pointed out, "we have full fbdev emulation, and drivers can implement the 2d accel in there. And a bunch of them do. It's just that most teams decided that this is a pointless waste of their time." He added that "compared to fbdev, there's a very active community who improves and refactors it every kernel release to make it even better. Since about 2 years (when atomic landed) we merge new drivers at a rate of 2-3 per kernel release, and those new drivers get ever simpler and smaller thanks to all this work."

But Geert simply replied: "This has been going on for years: 1. fbdev is obsolete, everybody should use DRM instead! 2. Can you please point me to a small sample driver for a dumb frame buffer? 3. Several are being written, but none of them is upstream yet. 4. Go to 1."

To which Daniel said that there were more than 20 small sample drivers using DRM already. And Geert pointed to Daniel's earlier quote where he said that some were floating around, but none had landed. Geert said he wanted "simple dumb memory-mapped frame buffers, which is what fbdev was initially developed for." And Daniel said, "small drivers like these we have piles now; things exploded a lot after atomic landed two years ago. And they seem to shrink with every release a bit more."

At this point Benjamin realized there was a lot of DRM documentation that had recently gone into the kernel. He ran off to read it, and returned, saying that his objections may be out of date.

Daniel also pointed out the MXSFB DRM driver, which he said was a good example of a simple driver using the display pipe helpers.

At this point, even Geert started to feel like maybe DRM was ready – or at least nearly ready – to replace fbdev fully.

The discussion continued for a bit, with more people joining in to discuss specific abilities of various drivers and specific needs of various sectors of users. But it does seem as though, finally, the main kernel folks who objected to DRM as too heavy weight have been mollified to some extent. The DRM folks have extended their code to begin to handle the most simple cases, which is what the FBDev folks want, and we can probably look forward to the final FBDev drivers being rooted out

of the staging directory in the relatively near future.

Plugging Security Holes at The Hardware Level

Paolo Bonzini wanted to lock down KVM security a bit further by preventing users from invoking certain assembly instructions – specifically, `SLDT`, `SGDT`, `STR`, `SIDT`, and `SMSW`. Each of these instructions have certain security holes. For example, as Paolo explained, `SGDT` and `SIDT` "can leak kernel-mode addresses to userspace, and can be used to defeat kernel ASLR [address space layout randomization]." He went on: "`SLDT`, `STR`, and `SMSW` aren't as bad because `SLDT` and `STR` only leak selectors, while `SMSW` only leaks `CR0.TS` in practice."

There wasn't much discussion, although Liang Z. Li offered to help with any further assembly lockdown efforts.

Struggling To Support USB Type-C

Heikki Krogerus posted a patch to add support for USB's Type-C connector class. As he put it, "The purpose of the USB Type-C connector class is to provide a unified interface for the user space to get the status and basic information about USB Type-C connectors on a system, take control over data role swapping, and, when the port supports USB Power Delivery, also control power role swapping and Alternate Modes."

Heikki's interface went into the `sysfs` directory and exported a lot of data about current operational role, current power role, the port VCONN Source, and many other data items. Ports would be named `usbc0`, `usbc1`, and so on.

Greg Kroah-Hartman offered some comments. After a cursory glance, he liked the patch, but there were a few things that needed to change. They went over some of those technical details together. Other people joined in, and Heikki spun out a few more versions of the patch. Among the issues under discussion were temporary problems like memory leaks. In terms of the overall feature, everyone seemed to be basically in support of it.

The thing about USB Type-C is that it's the new hotness in terms of data exchange and power delivery. Support for it is not optional if Linux wants to retain world domination status. For example, the MacBook Pro relies on USB-C. Among everything else, you can plug the connector in either right side up or upside down, a feature other USB connectors don't have.

At the same time, even though it's indispensable, the developers still have to figure out whether various behaviors should be set as module parameters, boot options, `sysfs` files, IOCTLs, system calls, or any number of other possibilities.

It's also necessary, as was pointed out in the discussion, to compensate for deficiencies in manufacturer's firmware. As Heikki put it at one point, "Unfortunately, we cannot assume the firmware to be always correct. Companies love to recycle the firmware. We are going to see products from a company X that should prefer source role, a desktop for example, but still give the OS a device property that says otherwise. The reason for that is most likely because the previous product from that company was some kind of mobile device."

While Greg at one point said, "firmware is 'hard', but it gets really really tiring constantly having to paper over firmware and hardware bugs in the kernel just because we seem to be the ones that are willing to actually fix problems that others cause."

Ultimately, Greg took a closer look at Heikki's patch and discovered some eyebrow-raising constructions, and the deeper he looked, the worse the code appeared to him. Eventually, he told Heikki that the patch needed a lot more work, and that "I'm now going to require that you get other internal Intel developers to sign off on this code before I review it again. You have resources at your disposal that others do not with your internal mailing lists containing senior kernel developers. Use it and don't waste the community's time to do basic code review that they should be doing instead."

So Greg chucked the code back to the mother ship, Intel, for a thorough going-over before it could even enter the sphere of kernel developers again.

Guenter Roeck, who had previously given his "tested-by" thumbs up for this patch, now said to Heikki, "This hurts both your and my reputation, and obviously will make me quite hesitant to add a 'Reviewed-by:' to the next version of the series." But Greg tried to tone things down, saying, "it doesn't bother me at all. I want and need your reviews of those portions that I don't know as well (i.e., the userspace api and functionality.) So don't take it personally; the driver model isn't that easy of a topic to mess with in places. Loads of people get it wrong." And he said he just really wanted those internal Intel reviews. But Heikki still apologized for the quality of the code.

Overall, a harsh debate. But not very far out of step with the sort of criticism that might normally be leveled against a patch submission. However, it's fairly rare for someone like Greg to decide to halt all reviews of a given patch until the corporation of origin does more work on it.

Blocking Hardware Input Events

Pali Rohár posted a patch to support disabling events from any input device connected to the system. Once disabled, a piece of hardware would no longer deliver any events to userspace. Some common hardware that might use this feature would be keyboards, touchpads, and touchscreens.

Nikita Yushchenko liked the patch, but Bastien Nocera thought it might need more thought. He had implemented something similar in userspace to disable touch devices when the screen has been suspended. Among other questions, he asked if Pali's patch might just be better in userspace and not be a kernel feature at all.

David Herrmann also pointed out that if you didn't want events from a device, you should simply not open that device.

Pali replied that doing this in userspace might require making changes to every piece of user software that took input, but Bastien replied that, "There's usually a display manager in between the application and the input device. Whether it's X.org, or a Wayland compositor." And so the input could be trapped at that level, instead of in the kernel.

There was a bit more discussion before the conversation petered out. In general, though, to paraphrase Murphy's Law: Anything that can be done outside the kernel, will be done outside the kernel. There are very few exceptions, especially when there are already existing userspace implementations that are very similar to the proposed kernel feature. ■■■

An interview with Jim Zemlin

Swimming with the Poets

By Swapnil Bhartiya

Jim Zemlin has directed the Linux Foundation since 2007, when the foundation began with the merger of the Open Source Development Labs (OSDL) and the Free Standards Group (FSG). Today the Linux Foundation has gone beyond Linux and become a huge umbrella that houses many open source projects that are critical to our economy and society.

To celebrate the 25th anniversary of Linux, I sat down with Zemlin for a 40 minute interview. He is very humble and doesn't like to talk much about himself. However, since we have known each other for such a long time, he opened up a bit. Here is the edited version of that exclusive interview with Jim Zemlin.

Linux Magazine: Can you tell us a bit about your childhood and your exposure to computers? What kind of technical background do you have?

Jim Zemlin: I have a very weak technical background.

When we were really young kids, personal computers were just starting to emerge. The computing industry had not yet consolidated. There was Commodore 64, the Ataris, Apple… and there were the Tandy kits. So I grew up with computers.

My grandfather was one of the founders of Cray Research. He had been in the computing industry since World War II. Then my father was a computer programmer at Control Data Corporation, which was headquartered in Minnesota. At that time, Minnesota was the hub of companies like Control Data Corporation, Honeywell, and 3M.

My dad was actually really smart about how to teach us about computers. He said that we could play games for 10 minutes, or we could program for an hour. We just wanted to be with the computer (and those were boring games), so even as a kid I started doing programming work. Just simple things like Basic. My brother and I ran a bulletin board system back in the mid 80s. Later on, my brother also worked as a developer on a game titled Oregon Trail (the game we used to play).

That's about where my technical acumen started and stopped. Just a lifelong fascination with computing systems.

LM: When did open source enter your life?

JZ: After the collapse of the dot-com bubble, I got involved with Covalent Technologies, which was started by early Apache Software Foundation developers. That was my first introduction to open source. I think it was through Apache that I originally got involved in open source.

LM: What led to the creation of the Linux Foundation?

JZ: When Covalent Technologies ended up being acquired by VMware, I kind of started having a midlife crisis. At that time I thought of becoming a chef, a professional rock climber… But I really got intrigued by the open source community. I got involved with the idea of creating a standardized way to run applications across the variety of different Linux distributions. That was sort of a response to fragmentation in Unix. That led me to working on the Linux Foundation by essentially consolidating a lot of different open source organizations into what the modern Linux Foundation is.

LM: The Linux Foundation has come a long way from those early days. It has become a huge umbrella that goes beyond Linux. What are your reflections on those early years?

JZ: You've known me for a very long time. You have known me before I was married, and I have been married for 11 years now. I remember my first date with my wife, and she asked me what did I do for a living, and I said, "I work at this non-profit organization; it's open source; the software is all freely available, and we give everything away."

She went to Harvard Business School. She is a very Type A senior executive at a technology firm in the Valley. When I told her about what I was doing, there was this disappointment on her face; it was just palpable. She started glancing at her watch to get out of there.

Fortunately (chuckles), I was charismatic enough to get her to marry me. And the Linux Foundation turned out to do something that I had hoped and thought it could do. It continues to be our North Star. At that time, we were very focused on Linux; we were creating the greatest shared technology in the history of computing.

I think everyone stumbles across this idea of what is the purpose of a particular organization. Once you understand that, once you grasp it, you can move ahead. In our case, it became clear that the DNA of Linux – the development model, the economy around it – could become a template for other large-scale open source endeavors. We realized that it would lead to the creation of a great collective shared technology investment, beyond Linux.

That continues to be our goal. Everyone at the Linux Foundation knows that we're here to support the creation of these great shared technology resources that are shared in the sense of the code, that are consumed by society – resources that governments and industries depend on.

Linux is a great responsibility today because the world is truly dependent on the software to remain secure and stable. It has become an effective platform for a lot of important computing systems that run our daily lives.

We are building a foundation for every level of the software stack, whether it's web frameworks, network function virtualization, software-defined networking controllers, or container management systems like Kubernetes. These are the projects that define the most important aspects of the technology economy.

LM: I think the Linux Foundation's biggest contribution is the fact that you have made it easier for corporations to work together on these technologies that are building our future.

JZ: As I said in my keynote speech, Linux has proved that you can better yourself while bettering others at the same time. The ability to make your company (and yourself) better and share that with everyone else is the real value of what we are creating.

Commercial adoption and sharing are not mutually exclusive. It is better to have a situation where everyone wins as opposed to someone wins and someone loses.

If we look at the code part of it, at the highest level, it's like writing poetry. People like Linus Torvalds and many other great developers are like poets. They're writing this incredibly important software and enjoying it, because it's an active artistic creation for them. That is then being used in some of the most interesting things in the world. We see people using Linux in unmanned aerial vehicles; they are using it in some interesting aspects of search and rescue; they are using a Raspberry Pi running Linux to teach kids about computers… it's amazing.

It's also important that companies have committed to running Linux on all of their modern systems. Amazon is a good example. Their shopping system, their entire e-commerce platform, devices like Kindle… all of that runs on Linux.

That dependency then begets Amazon improving that code, sharing those changes back with the open source community. Those changes then improve that open source code base, and then that cycle starts over again, but maybe this time with Facebook and many other players.

You get this virtuous cycle, which is complementary in nature and in fact is the whole point.

LM: Linux has played a huge role in making companies comfortable with the open source development model.

JZ: Yes. It's the proof that we can all work together on things that are non-differentiated plumbing, for example, as competitors and still compete effectively. In fact, we can compete more effectively, which is the key insight that Linux has taught. It has essentially redefined how software gets created.

The thing I like to think about is every time I talk to a company, whether it's GoPro making a camera or Toyota making a future in-vehicle navigation system, they are not going to go write their own kernel. Look at Linux. There are tens of thousands of packages, millions of lines of code, changes seven times an hour. Why would someone want to do all of that alone?

LM: What do you think Linux has achieved in these 25 years, and where do you think it is going?

JZ: I think it's going to be around for a long time, mainly because, if you look at history, operating systems tend to be much longer technology waves than applications. There are so many interdependencies upon it, both from a hardware perspective below the OS, and from the application economies built on top of that. But at the end of the day, you need software to interact with all the hardware underneath, and that's essentially what the operating system in Linux does.

What we've seen over the years is you had the mainframe operating systems, then the Unixes came into being,

then personal computing operating systems came into existence. But in each of those instances, they kind of peaked, and then they sort of slowly eroded away.

What has always limited many of these platforms is their inability to jump from one form of computing to the next. Unix, at the end of the day, is still pretty much a server operating system. I guess you could include Mac OS as, maybe, a counterexample to that. Personal computing operating systems have had a harder time jumping to mobile. The modern smartphone systems that we know today were new operating systems. There has been one exception: Linux.

Linux has jumped from server to high-performance computing to mainframes to mobile devices to embedded systems to tiny little real-time applications. That has not been achieved before, so I think that it's the malleability of Linux, it's the organic nature of the roadmap (which I think Linus, to his credit, has effectively managed by not having a real permanent roadmap) that will make Linux endure for decades to come.

LM: In previous LinuxCons, Torvalds said that Linux can't be shrunk anymore. If there are people looking for really small devices, they should look elsewhere.

JZ: These have been some interesting technology experiments. What becomes too tiny for Linux, or in other words, how small can Linux be before it just doesn't work and you have to move to a more traditional RTOS [real-time operating system]? Developers have got it really small. You can get it small, but there are real limitations.

Real-time operating systems are an example of an industry where there has been a lot of fragmentation. And the reason for that is when you are building some very small application using a tiny, real-time operating system, you generally buy a chip that comes with an RTOS. You then build your app on top of that (it's all custom development) and ship your product.

Today, the reverse is happening. There has to be an ecosystem, an application and developer ecosystem, around an operating system, even for a tiny RTOS, to enable more rich functionality that's required for today's modern computing environment. That's totally opposite of custom development done on one chip, which leads to fragmentation. There is an opportunity to consolidate a lot of that fragmentation in the real-time operating system market. We have a project called Zephyr, which is a good example of that. There are a lot of other real-time operating systems out there, and I expect that there will be consolidation in that space.

I don't think it comes at the detriment of Linux. I think that people trust Linux as a neutrally owned territory, and they can invest in it. People trust that no single entity can control or monetize Linux to the exclusivity of anybody else. We will see how it pans out; there is a lot of exciting development happening in that space.

LM: Recently SiFive announced open source chips. After conquering software, are you interested in doing hardware?

JZ: I haven't looked into those kind of initiatives, lately. We've looked into some things around quantum computing and other areas where the hardware dependencies are unique. But I tell you, our cup runneth over right now, just on the software side.

LM: In that case, what's your big goal now after conquering the world with software?

JZ: If our goal is to create the greatest shared technology resource in history, there are a few things we want to accomplish to help make that happen. One thing we've already done is that now open source essentially powers every modern computing system. If you want to build anything today, you use open source to create that technology.

The problem we have is that the industry and enterprises outside of the very narrow tech sector aren't capable of managing that external research and development as effectively as they should. They don't have a good procurement process. They don't have a good method for letting their developers participate in open source projects, to pull code into the company, to modify it, and to then release it back to the open source community. They don't have good methods for picking open source projects strategically. And as every company becomes a technology company, they need to be good at managing open source.

To address these challenges, we are developing a set of resources, whether it's attending our events, whether it's training, or whether it's participating in our projects, that will help bring in that next set of participants in open source to teach them essentially how to manage external R&D and how to leverage open source effectively.

The biggest bottleneck that we find is when companies come to us and say that they know they are using open source, they know that they need to understand how to maintain it over time, but their lawyers don't understand how to work within the licensing regimes. They don't understand the procurement process. They need help in all those areas, and we are building resources to help enable that to happen.

We also want to make sure that open source software is written securely. Secure coding workshops, threat modeling efforts, and better testing for all open source projects are things that we really want to do.

Those are big things. They will keep us busy for many years to come. So hopefully in 25 years, I'll still be around.

LM: That's too specific, what's the big picture?

JZ: If we can create this great shared software resource that's secure, that's economically sustainable by organizations taking a real commercial dependency upon it, and then in turn underwriting joint development of it, if we can foster a community that embraces diversity and brings in a new generation of the sort of code poets who write great software, I think we will be on the right track.

LM: After doing all of this amazing work, what satisfaction do you get when you go to bed every night?

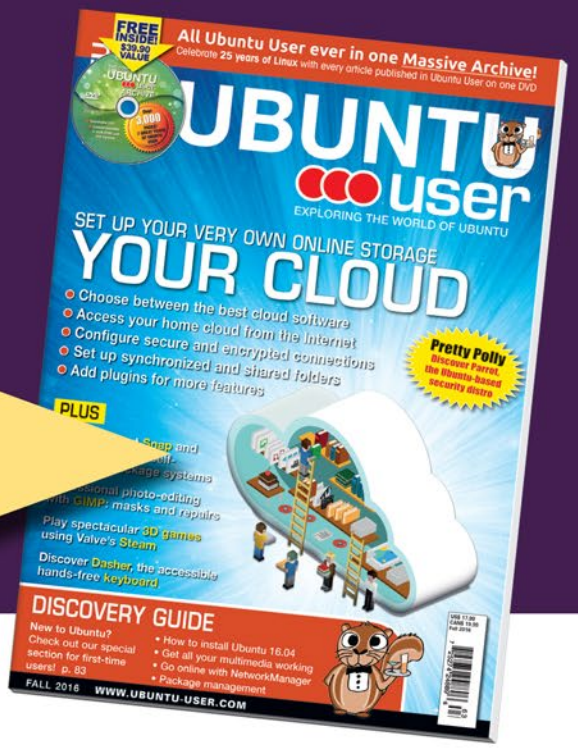
JZ: I read my daughter a book, every night before I go to bed. I'm satisfied that she's going to live in a society (for as many perceived problems as the world has today) that hopefully is a little better because people share, and that's certainly satisfying. ■■■

What?!

Archives

come with my

digital subscription?



Archives + Current!

Sign up for a digital subscription to get the latest issues of Ubuntu User, **PLUS access to archive articles.**

shop.linuxnewmedia.com

That's a lot of articles!

Better privacy with Tails

Invisibility Cloak

The Tails Live Linux distribution provides privacy-conscious users with easy access to the Tor network for anonymous surfing. *By Ferdinand Thommes*

The Internet today makes you transparent and vulnerable. Even popular solutions such as mail encryption and VPNs leave clues for someone who is motivated enough to track your activities. If you are serious about keeping your Internet affairs private, one remedy is an anonymizing distribution such as Tails. Tails automatically routes all connections to the Internet via the anonymizing Tor network.

The Tor network is a system of anonymous relay servers that conceal the location and identity of the computer sending the message or request. The basic techniques that spies and Internet advertisers use to uncover the source of an Internet packet will not work if the traffic is routed through the Tor network. You can download and install a Tor-ready browser directly from the Tor project website, but anonymity depends on more than just the browser. Other configuration settings on your system must reflect the same attention to security and anonymity if you wish to truly go unnoticed.

The Tails Linux distribution is designed to let users boot directly into a preconfigured anonymous environment based on Tor. Tails, a Live system that runs from a

VERSION 2.9.1

Tails 2.9.1, which follows hot on the heels of its predecessor 2.7.1, is more of a bug fix and maintenance release than a major update. The next major release is Tails 3.0, which is scheduled for June 2017 and is already available as an alpha version.

In addition to bug fixes, Tails 2.9.1 mainly focuses on updating the packages included in the bundle. The Debian kernel 4.7.8-1~bpo8+1 provides the basis; system management is handled to a great extent by systemd 215-17. The linchpin in the distribution is version 6.0.8 of the Tor Browser, which is built on Firefox ESR 45.6.0 (Figure 1). Tor itself is included as version 0.2.8.10. The Thunderbird email client, which is currently dubbed Icedove at Debian, is version number 45.5.1. Another change is the default search engine: DuckDuckGo (Figure 2).

Because of a security issue, the Debian developers upgraded the Apt package management front end to version 1.0.9.8.4; other security issues in Firefox ESR and Icedove were remedied at the last minute. The update of the Guest Additions to version 5.1.8 fixed a bug that prevented Tails 2.7.x from launching in VirtualBox.

The preinstalled applications now include the KeePassX password manager, the Dasher accessible text input tool, a Bitcoin wallet, and Gobby as a collaborative text editor.

DVD or USB stick, is not suitable for continuous operation due to the limits imposed by the speed constraints of the Tor network. Most users, instead, deploy Tails on an as-needed basis. Still, if you're looking for a fast and easy way to integrate the safe surfing capabilities of the TOR network, Tails is an easy and convenient alternative.

Also on the Go

The abbreviation Tails [1] stands for The Amnesic Incognito Live System. The motto of the Debian-based distribution is "privacy for everyone, everywhere." You can boot Tails as a DVD, USB memory stick, or SD card, so it is easy to carry around with you.

On Flash devices, you can set up a Persistent mode in a separate partition that allows you to store password-protected data from the Live session in a private, encrypted directory [2]. On the other hand, Tails reliably forgets all data if you do not enable persistence, and the system is immutable – that is, you can't make changes to it. You can thus use Tails without an Internet connection as a completely anonymous typewriter for confidential text.

The developers have already configured the Tails distribution for its intended purpose, which saves the user significant time and helps avoid security-related configuration errors. The project publishes a new version every two months. In mid-December 2016 the developers released Tails 2.9.1 (see the box entitled "Version 2.9.1").

Two-in-One

When looking for a Tails image to download, do not be confused by the fact that the only ISO you find at the Tails website has an identifier of `i368` for 32-bit mode.

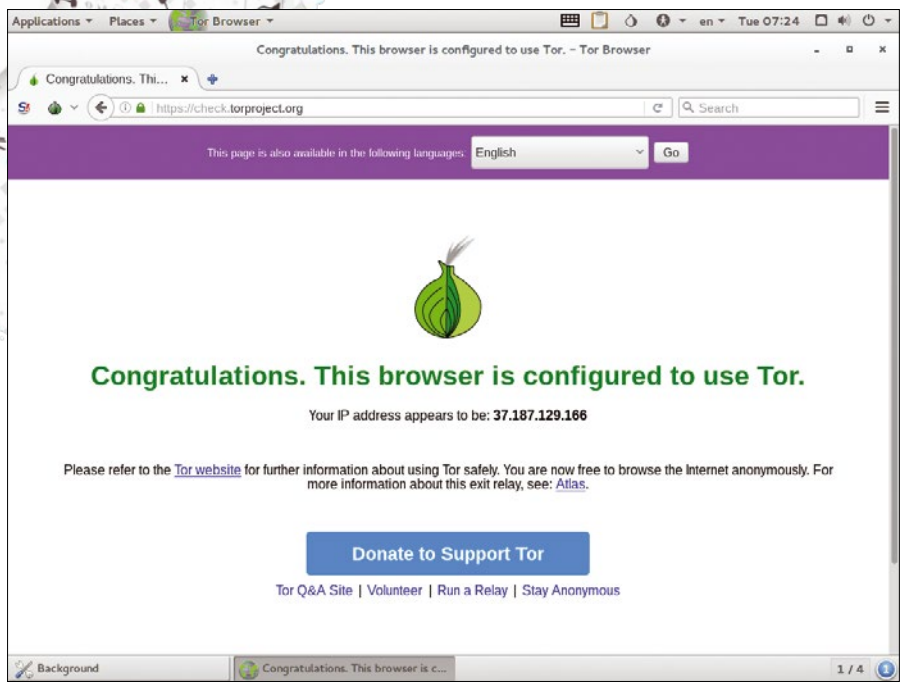


Figure 1: The Tor Browser guarantees the anonymity of the user.

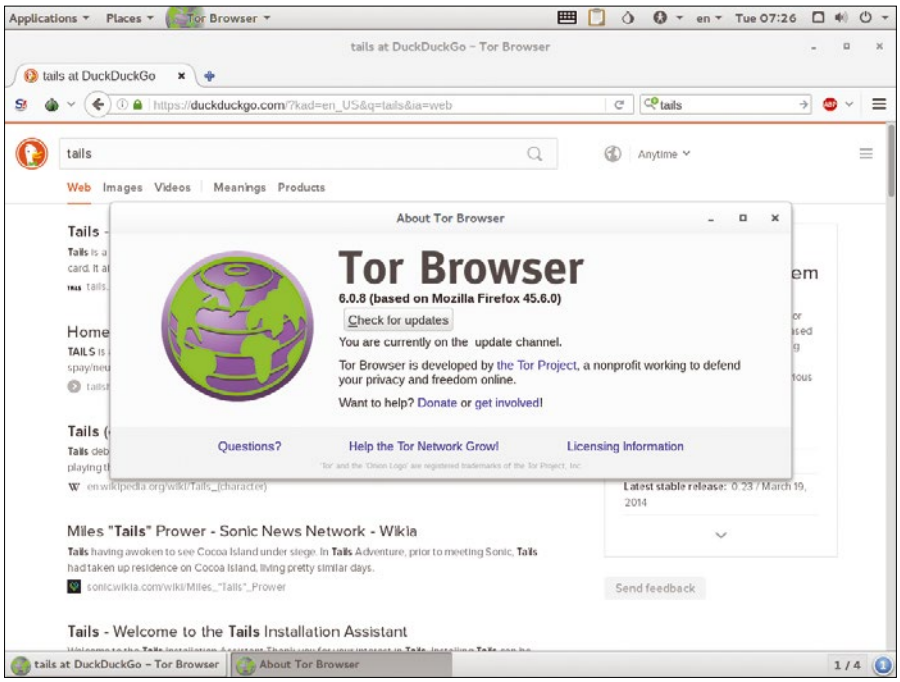


Figure 2: DuckDuckGo supports anonymous searching in the browser.

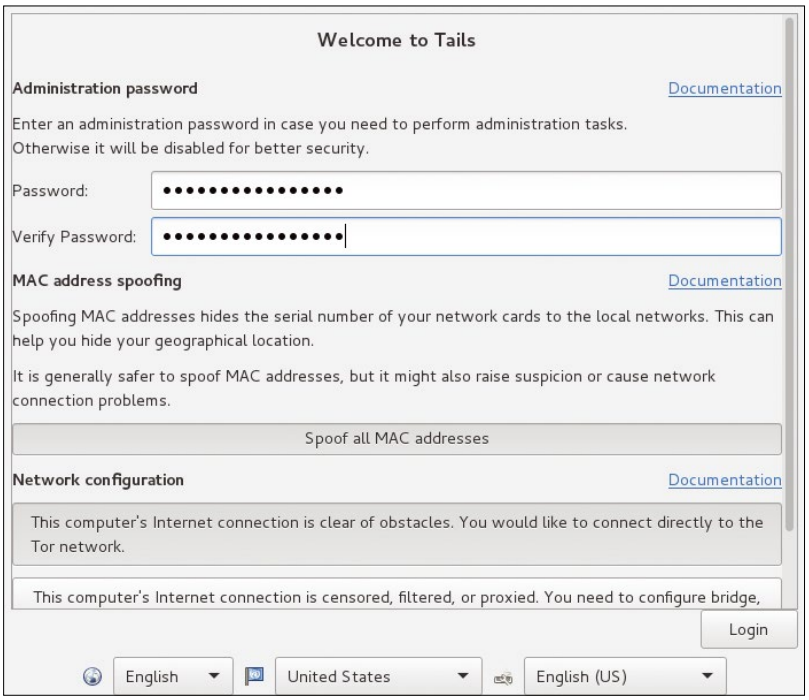


Figure 3: A number of important settings are available in the Tails welcome screen.

It is a hybrid image that boots either a 32- or 64-bit kernel depending on the architecture.

After you start Tails as a Live system, the first screen to appear is *Welcome to Tails* (Figure 3). When prompted about additional options, you will want to say *Yes* to enter a root password, which is disabled by default. You can also manipulate the MAC address to make your system activities more difficult to trace. In addition, you can disable all network functions.

After clicking *Apply*, you are taken to the Gnome 3.14 desktop. The developers use Gnome Classic mode, which more closely matches the design of Gnome 2. In the background, the system sets up access to the Tor network and, after about one minute, prints an announcement at the bottom of the screen saying that Tor is now ready.

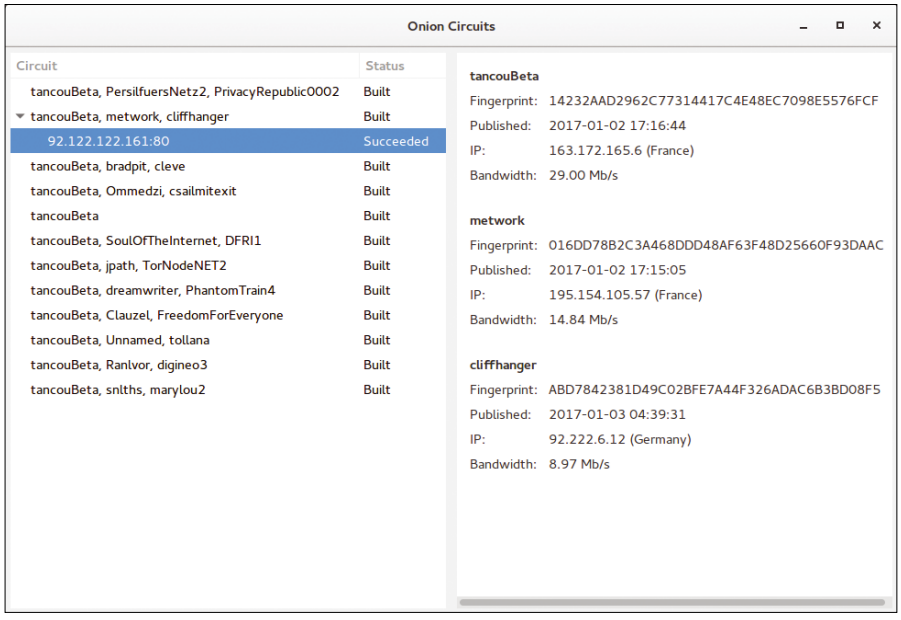


Figure 4: Tails routes traffic across multiple target nodes.

You can then start the Tor Browser; you will notice that the launch is somewhat slower than usual. Tunneling the connection through the Tor network definitely has an effect on performance. To discover whether or not you are actually surfing with Tor, you can check the small onion icon in the top-right notification area. An X in the onion means that Tor is disabled – in which case Tails then automatically blocks all connections to the Internet.

Secure Communication

Pressing the onion icon displays *Open Onion Circuits* with a list of nodes currently used on the Tor network (Figure 4). In each line, you will see three computer names for the input, middle, and output nodes of the Tor network. Clicking on an entry shows the related properties, such as the fingerprint, the IP address, the location, and the node's bandwidth. The *Internet* option in the application menu also offers you the option of choosing *Insecure Browser* to use Firefox without detouring via the Tor network.

The developers have also modified the Icedove email client for Tails, resulting in TorBirdy [3]; view the TorBirdy configuration by clicking the bottom right border of the Icedove window. You can make the profile stricter by forcibly encrypting all outgoing emails with the *Enigma* extension.

Messengers offer another approach to communicating over the Internet. Tails uses the Pidgin instant messenger, which uses the Off-the-Record (OTR) messaging protocol for encryption and secure authentication of the opposite end (see the box entitled "OTR"). However OTR is disabled in Tails by default, because you have to generate a private key before you can use it [4]. To access the configuration in Pidgin, go to *Tools* | *Plugins* | *Off-the-Record Messaging*.

If you use Pidgin for IRC via Tor, keep in mind that some channels (such as Debian) block visitors over Tor because spammers often use Tor to distribute spam. The Tor website has a list of IRC networks blocked for and open to Tor [5]. For more information on secure communication with Pidgin, check out the Tails documentation [6].

Installation

The Tails application menu offers a *Tails* item with documentation and access to the installation options. One

OTR

The OTR messaging protocol regulates the continuous updating and management of short-term session keys. As a special feature compared with classical encryption, OTR ensures that it is no longer possible to determine at a later stage whether a particular key was used by a certain person (plausible deniability).

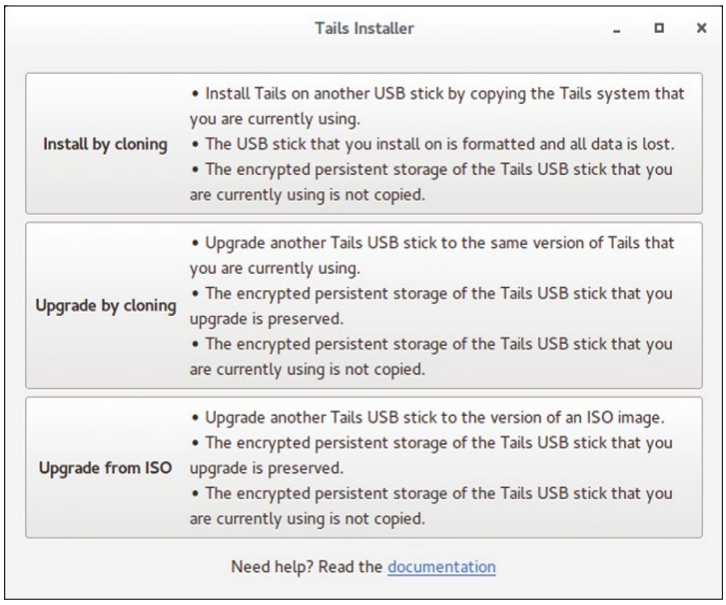


Figure 5: The Tails Installer offers several options.

option is the Tails Installer (Figure 5), which you can use to store the image on a DVD, a USB stick, or an SD card with at least 3.5GB of free space. You also have the option of creating an area for persistent data storage on the installation media (you will need additional storage capacity) or safely overwriting the data storage. You need to enable this area at the start of a session and release it by entering a password. To access this persistent space, select *Places* | *Persistent*.

In addition to accessing the Tails installer through the disc image, you can also download it via the package manager on Debian and its derivatives. Instructions for accessing the Tails installer through Debian [7], other Linux distributions [8], and Windows [9] are described in minute detail in the documentation. Another option is to copy the Tails installer from another Tails installation that you trust [10].

Installing from Debian or one of its derivatives is simple. You need a recent Tails ISO image and a recordable DVD, USB stick, or SD card with at least 3.5GB capacity, and finally the Tails Installer, which you can install via Apt or a graphical package manager. On Debian 8 "jessie," you need to enable backports [11].

The Tails Installer automatically detects a plugged-in USB stick. If you are using multiple external media on the computer, make sure that you choose the correct device when confirming the target, because the installer will erase all data on the device. Finally, select the Tails image and start the installation (Figure 6).

The process takes about ten minutes and ends with a reboot. Subsequently, Tails boots from the USB stick or other boot medium.

Conclusions

The Live distribution Tails uses the Tor network to ensure relatively good anonymity. Relative because the Tor network can also be deanonymized given sufficient time and effort, although only intelligence services or governments have the means to do so.

Tails is not intended for installation on hard drives in computers. The system is based on Debian stable and is updated frequently to keep Firefox and the Tor Browser up-to-date and thus stable. Tor not only protects users when surfing the Internet; it also hardens the email client and the messenger.

Figure 6: In about 10 minutes, the image is stored on the medium.

If you want to delve deeper into Tails, see the Tails documentation and the work of the Capulcu activist group [12]. In the scope of its "Magazine for the promotion of resistance against the digital attack," Capulcu offers a very detailed manual for the operation and use of Tails. ■■■

INFO

[1] Tails: *https://tails.boum.org/*

[2] Persistence: *https://tails.boum.org/contribute/design/persistence/*

[3] TorBirdy: *https://addons.mozilla.org/en-us/thunderbird/addon/torbirdy/*

[4] OTR: *https://securityinabox.org/en/guide/pidgin/windows*

[5] IRC blockade: *https://trac.torproject.org/projects/tor/wiki/doc/BlockingIrc*

[6] Pidgin: *https://tails.boum.org/doc/anonymous_internet/pidgin/index.en.html*

[7] Installation on Debian: *https://tails.boum.org/install/debian/index.en.html*

[8] Installation on Linux: *https://tails.boum.org/install/linux/usb/index.en.html*

[9] Installation on Windows: *https://tails.boum.org/install/win/usb/overview/index.en.html*

[10] Copying an installation: *https://tails.boum.org/install/linux/clone/overview/index.en.html*

[11] Backports: *https://backports.debian.org/Instructions/*

[12] Capulcu: *https://capulcu.blackblogs.org/wp-content/uploads/sites/54/2015/12/BandIlen-2016-02-20_orig.pdf*

SUBSCRIBE NOW

6 issues per year!

The New IT

New tools, new threats, new technologies...Looking for a guide to the changing world of system administration?

shop.linuxnewmedia.com

An open source router built for security

Secure Networker

Home routers are known for weak security. Turris Omnia is an attempt to build a better router through the power of open source. *By Jan Rähm*

Hundreds of Internet routers inhabit the IT consumer marketplace. However, the little boxes that connect our home or work offices to the Internet are continually causing a stir. At the end of November 2016, 900,000 customers of a German telecom company were cut off from the Internet for hours – and even for days – because the Speedport router supplied by the company fell victim to a denial of service attack.

Strangely enough, the attack was not even intended for the routers. Instead, the attackers wanted to penetrate the vulnerable remote maintenance interface of a completely different device type. In order to exploit an existing vulnerability of the targeted routers, and thus integrate them into a botnet, the attackers indiscriminately flooded the Internet with port-knocking packets to open a communication channel to the affected systems.

An investigation revealed that the company had left port 7547/TCP wide open on the devices; customers had warned the company as early as 2014 of this potential and completely unnecessary vulnerability, but for whatever reason, many devices were still vulnerable.

The reward for this incredible sloppiness was the attack: The many incoming request packets to the remote interface maintenance that had needlessly been left open were unable to open the affected port and certainly unable to infect the basically immune systems. But the sheer mass of the packets caused the routers so much confusion that they entirely quit working.

This kind of negligence is, unfortunately, all too common with network device providers: Unneeded ports are left open, default passwords are active for far too long, and vulnerable system software remains unpatched. Users often don't even receive a warning, or if they do – as they did with the German telecom – they often ignore potential security problems until the disaster hits home.

As several studies have revealed, many home routers are incredibly insecure, with out-of-date firmware, open ports, and unpatched operating systems. Most business networks have an IT professional – or sometimes a whole team of pro-

fessionals – overseeing the security system and interpreting event logs to look for signs of intrusion. On home networks, the only "admin" is the router owner, who is typically an amateur with neither the time nor the inclination to become an expert in the home router appliance.

The need for better security, and for ongoing monitoring of home router traffic to look for possible attacks, has led to an innovative program sponsored by CZ.NIC, the non-profit company that manages the Czech Republic's top-level domain. CZ.NIC designed its own router for users on its network. The Turris Omnia router [1], which was financed through a crowd-funding campaign, incorporates some best practices missing from many router designs and is intended to be very secure.

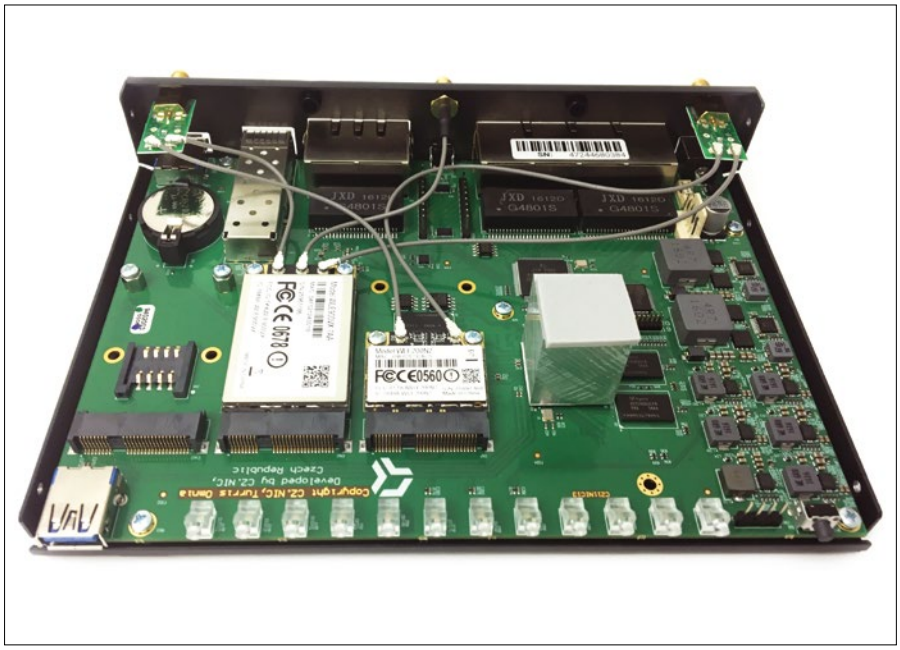


Figure 1: High-performance hardware with a simple look: Turris Omnia is equipped with three mini-PCI Express interfaces, plenty of RAM, a fast ARM processor, and 8GB of non-volatile storage. (Source: CZ.NIC)

One of the more interesting features of the Turris Omnia is that it can collect data on incoming traffic to look for signs of an attack in progress. The data is forwarded back to CZ.NIC and can be used to warn other users and, ideally, develop a remedy for the attack.

The Turris Omnia, which uses open source software and open hardware wherever possible, is a good example of a real-world device built to harness the power of the OpenWrt Linux project [2] for residential gateways and other embedded devices. The Omnia is available now in Europe. FCC approval for use in the US is still pending (see the interview with head developer Bedrich Kosata at the end of this article). We decided to investigate the Turris Omnia because it seemed like an interesting approach for how to address the persistent problem of home router security.

High Tech in Plain Wrappings

The Turris Omnia router comes in two variants that only differ in terms of memory. Just EUR289 (a little over $300) will buy you the router with 1GB of RAM, and EUR329 (around $350) gets you twice the capacity. The Turris Omnia impresses with powerful hardware under the hood. A two-core 1.6GHz ARM CPU is at the heart of the machine, with 8GB of permanent SSD storage.

Thanks to the three mini-PCI Express slots, of which one supports the SATA protocol, hard disk storage can be extended with SSD blades. On delivery, two of the three slots are occupied with WiFi 802.11b/g/n and ac modules (Figure 1). You can also use the LTE modem or other WiFi modules in addition to the storage media. The router is prepared for use with cellular modules and features a SIM card slot on the motherboard.

The entire hardware of the Turris Omnia is fully supported by the current Linux kernel. With one exception, free drivers and documentation are available for all components; only the WiFi interface requires a binary from the manufacturer.

Visually the Turris Omnia is quite unobtrusive. The technology is encased in the kind of unadorned metal housing typical of (semi-)professional network technology. The 12 LEDs on the front deliver information on the current status; you can control the brightness and even color by using a switch on the right side. In addition to the LEDs, the unit has two ready-for-use USB 3.0 ports on the left front.

The back of the router is dominated by three WiFi antennas, which can easily be replaced depending on how you plan to use the device. When you look at the connections, the SFP port is quite a surprise: It can house various networking modules and is rarely found in devices in this price range (see the box titled "SFP").

You can use the SFP interface (Figure 2) to integrate the router with the LAN or on the Internet with a high-end network medium such as glass fiber. Alternatively, the port will accommodate a modem for VDSL or other connection technologies. If you are already using a broadband modem, you can connect it with the Turris Omnia via the GbE-capable WAN interface and a network cable. Five Gigabit network sockets and a second USB 3.0 port complete the range of interfaces.

Open in the Positive Sense

The software of the system is based on the OpenWrt router's open source operating system; the hardware is almost completely documented and described. This open policy regarding information on the device means a user with sufficient skill could check the source code or even build a completely new router.

SFP

The acronym SFP stands for Small Form-factor Pluggable. The SFP specification defines modular, hot-swappable optical or electrical transceivers for Gigabit Ethernet, Fibre Channel, and SONET. The original specification envisages a 5Gbps data rate, although SFP+ interfaces now exist with up to 10Gbps.

Figure 2: Exotic specimen on the router market: The Turris Omnia offers five Ethernet interfaces, as well as one Gigabit Ethernet interface, WiFi, and an SFP port. (Source: CZ.NIC)

Even if you do not want to go that far, you could still expand and rebuild the installed system or replace the software with a different distribution.

The built-in distributed firewall is one of the outstanding and unique features of the router. The firewall is designed to protect the user against zero-day exploits. CZ.NIC wants to detect and analyze these attacks and evolve counter-strategies from the findings as soon as possible, transmitting the necessary fix directly to the consumer equipment.

To monitor the network for a possible attack, the manufacturer collects the data traffic between the client and the Internet. Turris Development Director Bedrich Kosata explains: "The fact that we are familiar with the traffic of many customers, who are anonymous for us, means that we have a large database; this large collection of data will ultimately benefit customers." CZ.NIC analyzes the data for unusual patterns. In case of conspicuous or even suspicious behavior in the traffic, it warns customers. In its delivered state, the firewall is disabled; the customer first needs to enable it. That is, CZ.NIC lets the users decide, said Kosata, whether they want to have their data traffic analyzed. CZ.NIC only uses the acquired data for traffic analysis, and this information is subject to very strict rules on data protection.

The auto-update feature is one of the additional security measures. During setup, the router asks whether you want to use the function. If you prefer mature software, you can do without the auto-updater. In view of recent IT security incidents, however, you are better off installing updates as soon as they are available.

Another indicator of the Turris Omnia's sophisticated security is that no Internet-facing ports are open in the factory defaults. Functions such as UPnP are disabled; you need to enable WiFi only during the initial setup, and you need to set all the access credentials yourself during guided commissioning.

Simple or Complex?

Installing the Turris Omnia is quick and easy. First, connect the device to the power supply and then connect the WAN port to a modem using a cable. Use a network cable to connect one of your own computers to one of the five Ethernet ports on the router for the initial setup. Then, call the router's web interface on the computer to access the homepage of a simple, but quite effectively designed, installation dialog. The installer guides you through the start-up in 10 steps.

You'll create the necessary access credentials and enable the well-secured WiFi network, among other things. The whole device is up and running within a few minutes. If the software needs to download many updates, this can take a little longer, depending on the bandwidth of your Internet connection. You can then access the simple web interface of Turris Omnia, where you can perform rudimentary settings. Although these are sufficient for general operation, they omit important functions, such as network-attached storage (NAS). All the options are described well.

Things look quite different in the advanced section: Regardless of the visually appealing design and good structure, the options offered here require in-depth expertise. The interface offers you the possibility to influence virtually every aspect of the router and the software – far beyond what even a demanding home user or small office will need (Figure 3).

Light and Shadow

During a test in an inner-city office in Berlin, we regularly reached wireless data rates with the Turris Omnia of over 800Mbps. This rate corresponds almost exactly to the maximum possible gross data rate of 1.3Gbps, which theoretically allows 802.11ac. No other 802.11ac router we put into operation for comparison achieved higher data throughput. The router is therefore in the absolute top group in terms of wireless performance.

The Turris Omnia also has its downsides, including its price/performance ratio. At a purchase price of EUR289/329, the router is in the upper mid-price range, although it lacks inter-

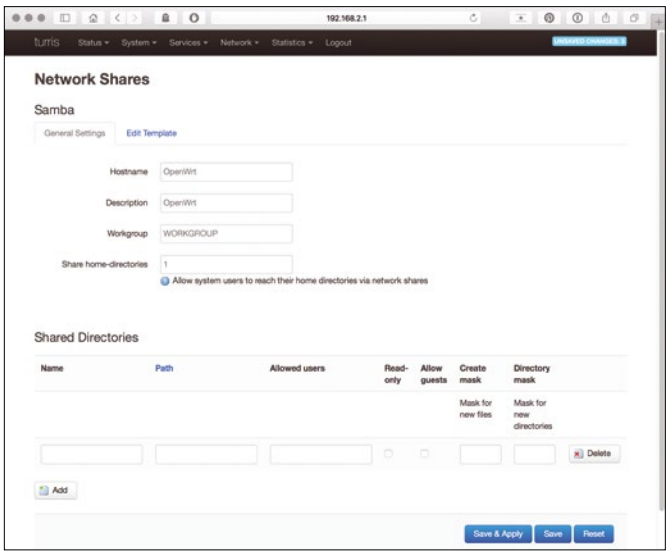


Figure 3: Knowledge needed: The advanced Turris Omnia web interface requires in-depth Linux expertise on the part of the user, as shown here for sharing network drives.

faces for analog or ISDN phones. You can, however, retrofit telephony features if you have some background knowledge.

Conclusions

The Turris Omnia is an excellent piece of hardware. Both the workmanship and the underlying, open concept offer no leeway for criticism. However, this does make the missing telephony features all the more apparent. The update function makes a good impression, and given the organization behind the Turris Omnia project, users can expect long-term support. The distributed firewall will need to demonstrate its capabilities in a long-term test.

If you do not feel comfortable with the thought of a third party analyzing your data, you do not need to activate this option and can simply rely on the built-in firewall. The firewall supports highly granular configuration, given appropriate knowledge. However, the Turris Omnia would benefit from a clearer and more user-friendly configuration interface for non-Linux experts. ■■■

INFO

[1] Turris Omnia: *https://omnia.turris.cz/en/*

[2] OpenWrt: *https://openwrt.org*

INTERVIEW: Turris Omnia Development Head Bedrich Kosata

Bedrich Kosata, Head of Development with CZ.NIC.

The Turris Omnia is not the first hardware project by the CZ.NIC. We caught up with Bedrich Kosata, Head of Development for the Turris Omnia, at the OpenWrt summit in Berlin, Germany, and asked about the objectives for development of this ultra-secure router.

Linux Magazine: *The domain registrar, CZ.NIC manages the top-level domain in the Czech Republic. What prompted the company to also develop network equipment for end users?*

Bedrich Kosata: We are a non-profit company and seek to use profits from the CZ domain for the good of the public. This is why we focus on open source and IT security. So we figured that it would be instructive to see what kind of traffic flows between the Internet and home networks – who attempts access to home networks and in what way. The idea evolved into the Turris project: We gave the people special routers, just to monitor this traffic and to see whether we could identify anomalies, malicious software, or the like.

LM: *When was the Turris project founded?*

BK: We had the idea of the end of 2012, and we started the project in 2013. Initially, we did not want to make our own hardware, but we failed to find any products that met our standards. We thus had to develop the hardware itself from scratch willing or not. In 2014, we delivered the first two router models free of charge, in exchange for data from users. Anyone who wanted to take part just had to sign a contract for three years. In return, we maintained the boxes and provided updates but also collected data for analysis.

LM: *Now the Turris Omnia is ready – the third router by CZ.NIC and the first financed by crowd funding. How did you manage to make the device completely open source?*

BK: We open-sourced all the chips so that the mainline kernel would support them; all the drivers were required to be open source. The only exception is the WiFi driver: You will not find a completely free driver; there is always binary firmware that is not disclosed.

LM: *What makes this router secure?*

BK: It all starts with the basic setup. It is well known that default passwords are some the biggest security problems on the Internet. That's why we force the user to define their own, sufficiently strong passwords during the setup. This makes our router secure from the outset, in addition to regular updates and advanced features such as the distributed firewall.

LM: *The distributed firewall – what is that exactly?*

BK: The firewall collects data from various sources – the routers themselves, but also from our company or externally from the Internet. From this we create an IP graylist and watch the conspicuous addresses in particular. If a router connects to one these addresses and we discover suspicious or malicious activity, we warn our users.

LM: *This is not something that everyone will want – isn't this an invasion of privacy?*

BK: By default, the distributed firewall is not active; the user has to enable it explicitly. We are not interested in the private data but only in the local firewall logs: This information lets us see who is attempting to log onto the router from the outside, and which services are especially subject to attack. To discover what is happening on their own routers, users can use a special portal that also shows the volumes of data exchanged between the router and the Internet.

We collect only the information that we really need, in particular metadata – who is talking to whom. We are not interested in the content at all. Our analysts see only anonymized data sets; also, we destroy all the individual data after ten days and then only keep the aggregated traffic data. This is also part of our privacy policy, which the user has to agree with.

LM: *Are there more security measures in addition to the distributed firewall and local hardening of the router?*

BK: We have also set up honeypots in the form of virtual routers and servers to determine how attackers attempt to intrude. In the case of Telnet access, we only present a login where the attacker can continually enter their username and password, until it gets on their nerves and they give up. But this provides us with interesting data about botnets in particular. The SSH honeypot shows the attacker a system that they can supposedly infiltrate. We thus learn what kind of malware the attackers are trying to install can analyze the results. The honeypot is isolated from the user routers so that real routers will not be compromised.

LM: *For some time, open source routers have had a problem with official approval: The US FCC, in particular, but also the EU, require some kind of lockdown of the wireless interface. How do you handle this?*

BK: That is a real problem. We are in the process of pursuing FCC approval, which is seriously slowing us down. We want to make the router as open as possible, and now we need to lock down part of the hardware. Currently, we are collaborating with the manufacturer of the WiFi cards to find a good solution for all parties. Ultimately, we will probably need to offer a separate version with a lockdown for the US market in order to achieve FCC certification.

The Signal messenger app encrypts voice and text messages

Private Messenger

Signal is an efficient private messenger app that encrypts voice and text messages, integrates easily into existing interfaces, and places all communications in a single display. *By Bruce Byfield*

Dozens of private messenger apps are available today; however, only one has the endorsement of both Edward Snowden and Bruce Schneier and is recommended by both the Electronic Frontier Foundation (EFF) and the American Civil Liberties Union. That app is Signal Private Messenger, developed by the non-profit Open Whisper Systems [1] for Android, iOS, and desktop environments. These endorsements are the result of not just Signal's ability to encrypt voice and text messages, but also its ability to integrate into existing interfaces for ease of installation and use.

Signal originated in RedPhone and TextSecure, two proprietary encryption tools for Android developed by Whisper Systems, founded by Moxie Marlinspike and Stuart Anderson. Whisper Systems was bought by Twitter in November 2011, and within half a year, both RedPhone and TextSecure, were released under the third version of the GNU General Public License. A year later, Marlinspike left Twitter to found Open Whisper Systems, which is funded by donations and grants, a neutrality that partially explains the high regard for its products.

Since 2013, Open Whisper Systems has merged RedPhone and TextSecure into a single application, adding encrypted group chat and gradually developing Android and iOS versions with comparable feature sets. Recently, it released a beta version of Signal Desktop [2] in the form of a Chrome app. So far, the desktop version, compared with the other versions, has a simplified feature set lacking password protection, for example. However, when linked to a mobile device, Signal Desktop provides centralized storage, as well as the increased usability of a mouse and a full-size keyboard.

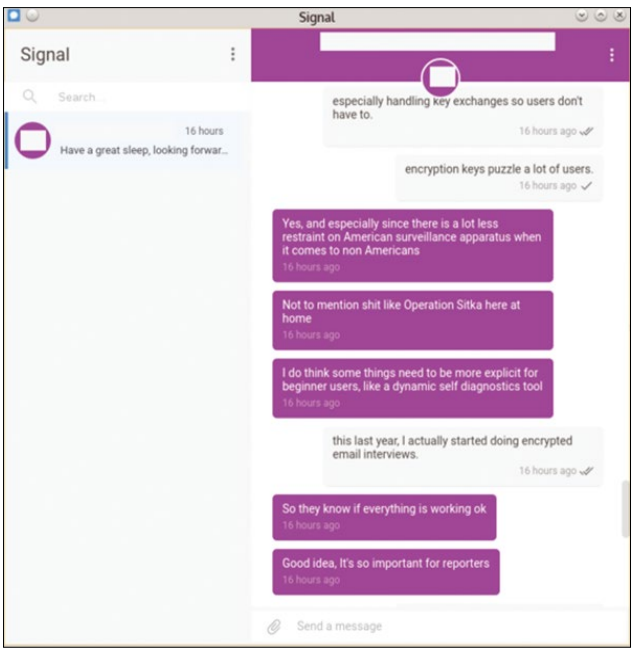


Figure 1: Signal displays both phone and text messages in a single display. Here, the log of a text conversation displays.

Signal is designed as a drop-in replacement for both for voice and text messaging apps (Figure 1). Although voice and text messages use separate protocols, from the perspective of users, the two are treated almost identically, and both are free of cost. Contacts are added from a device's Contact app into Signal, with encryption keys stored locally.

Calls in Signal are routed through Open Whisper Systems' servers, which handles the exchange of public keys without the need of input from users. Unlike the popular Pretty Good Privacy (PGP) [3], Signal's protocols switch encryption keys regularly, making conversations harder to crack. Although such encryption keys are ordinarily called fingerprints, Signal refers to them as safety numbers [4] – presumably to replace the often obscure jargon with a more user-friendly term. Users can manually approve and verify safety numbers, either visually or through a QR code, but Signal can still function without these steps.

Additionally, users can manually delete messages or set times when they will be deleted automatically. Signal and its database can also be protected with a passphrase.

What is noticeable about all of Signal's operations is how much they are hidden by default. In most encryption implementations, encrypting and decrypting are additional steps, and these complications probably deter many from using them regularly. By contrast, encryption in Signal is invisible to users unless they specifically change the settings. From the interface, using Signal appears no more complicated than unencrypted messaging – a claim that few other messaging systems can make, although Signal protocols have been widely borrowed, including in CyanogenMod and Facebook Messenger.

Installing Signal

Signal requires installation on an Android or iOS phone. Tablets are not currently supported. For convenience, you can also install Signal Desktop, although it is not necessary for using Signal and cannot operate on its own.

Installing on an Android phone (Figure 2) is only slightly more complicated than installing any app in the Google Play Store [5]. However, if necessary, you can follow the instructions at the EFF website [6]. Similar instructions are available for installing to iOS devices from the Apple App store. Unlike most Android apps, it requires access to almost all aspects of your phone, which for any other app might be a security risk.

Once Signal installs, enter your country and phone number and click the *Register* button. After you re-enter this information to ensure accuracy, Signal verifies your number and sends you a confirmation text.

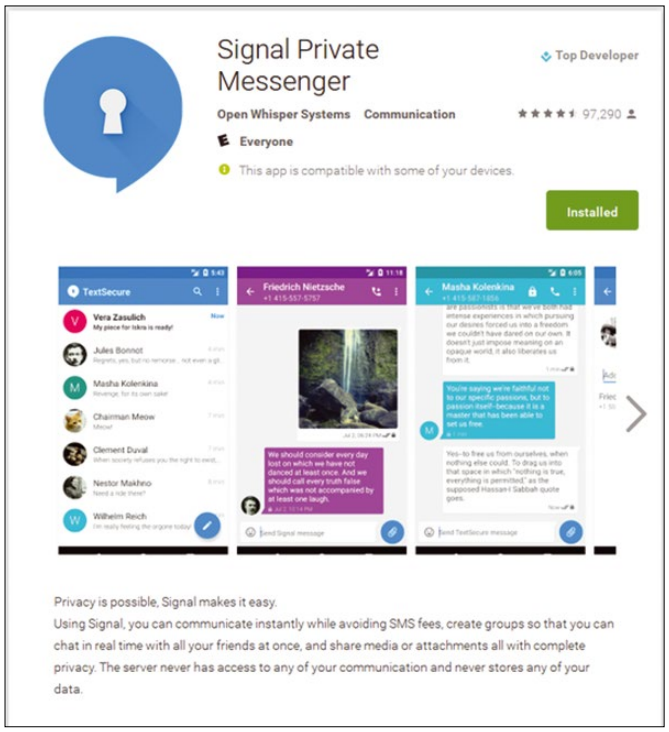


Figure 2: Signal installs as an Android app (shown here) or as an iOS app (not shown).

The installer then asks if you want to make Signal your default messaging app and imports your existing contacts if you accept. Your phone's default app will probably warn of dire consequences if you do so, but you can still use the original app if necessary, so this warning can be safely ignored. In fact, since Signal displays both voice and text messages for which you have a phone number in a single list, if anything, switching to Signal is a general convenience. Besides, if a listing is not Signal-enabled, Signal still lets you exchange unencrypted messages with it, so there is really no reason to be concerned about the replacement.

At this point, Signal is ready to use. However, you might choose to install Signal Desktop, which is not capable of sending messages by itself but offers the convenience of a larger screen and the use of a mouse.

Signal Desktop is also available as a Chrome app [7]. So far, at least, it does not run on any web browser except Chrome or Chromium, although it can be used with other Android or iOS phones.

Signal Desktop is installed via a wizard (Figure 3). At the end of the installation, the wizard displays a QR code (Figure 4). For Signal Desktop to function, you must link it by selecting on a device *Setting* | *Linked devices* from the menu in the upper right corner, and then scanning the QR code that displays from your phone. When the desktop recognizes the QR code, encryption keys are generated for communication between the phone and the desktop. If you add or delete contacts when using the linked phone without Signal Desktop, the next time you use it, select *Settings* |*Contact* |*Import Now* to resync.

Using Signal

Whether you are using the desktop or a phone, Signal is much the same. The main differences are that the desktop has fewer

settings and, in the beta version, has three restrictions: It can delete but not add contacts, shows only contacts with which you have interacted, and can only place a call with phone or voice if you have already done so at least once from the linked phone.

On a linked phone, you can still use the original apps for contacts and phone calls without using Signal, but any missed messages from Signal display in them. Additionally, the phone has options for setting notifications. On both the desktop and the phone, you should add a passphrase to Signal – after all, it hardly makes sense to go to the trouble of setting up encryption, and then having encrypted messages accessible to anyone who reaches your desktop. Start Signal Desktop from the Apps icon in the upper left corner of the browser.

To communicate, either click the phone icon in the title bar of a contact or use the text field at the bottom of the screen. You can also add an image or audio file, a shot from the camera, your location, or another contact to a message by selecting the paper clip at the bottom right of the screen.

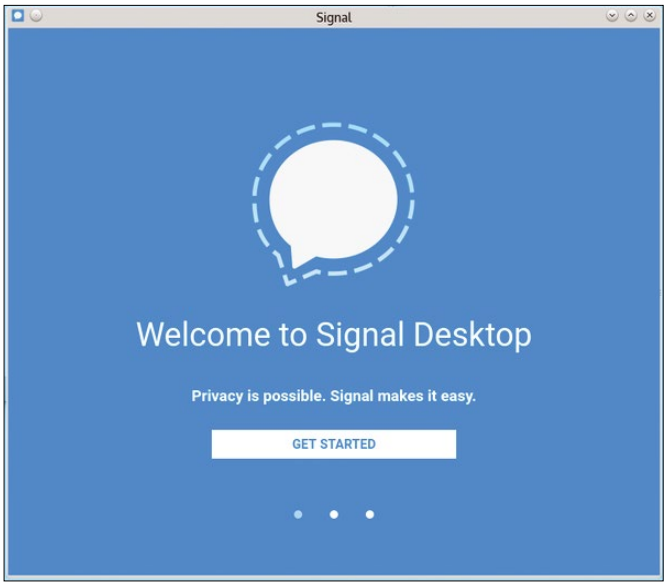


Figure 3: A wizard guides users through the installation of Signal Desktop.

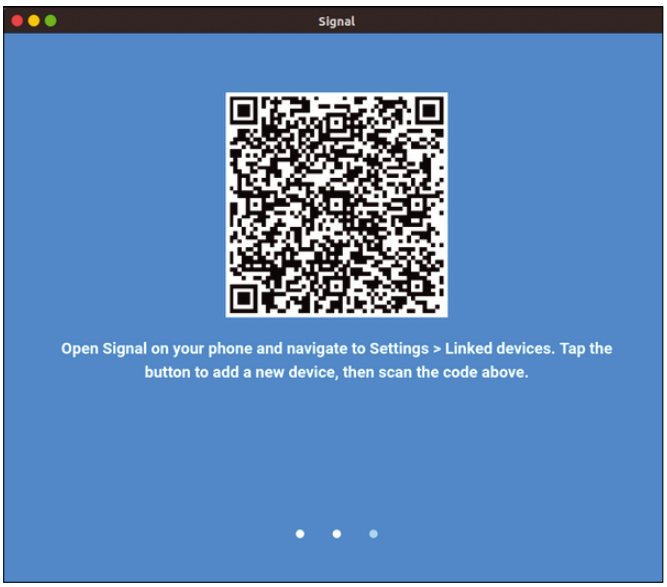


Figure 4: Signal uses QR codes to transmit encryption keys between Signal Desktop and a phone.

If the phone number you are contacting is not already Signal-enabled, you can still send to it. However, when you call unenabled numbers, an option displays below the title bar that gives you an option to invite your contact to join Signal. In any other app, this option might seem like blatant opportunism, but because all parties in a conversation need to use Signal for encryption, in this case, the advertising seems forgivable.

Figure 5: Although Signal handles the exchange of encryption keys automatically, you can verify them for yourself.

From each contact, you can also manage your exchanges using the menu at the upper right in the title bar. As you might expect, you can delete the log of your exchanges or change the color-coding for the contact. More unusually, you can set the time from the present that the log expires, display all exchanged images, or verify safety numbers with the link provided (Figure 5). Should a contact become a nuisance, another option is to block them via the Conversation settings submenu.

An Example for Security

Signal does have a few limitations. In particular, contacts must have a phone number, not just an email address. Perhaps the most serious limitation is that it must run on specific equipment and operating systems. However, given that the necessary conditions, hardware, and software are readily available, these limitations are mostly matters of preference and are seldom a barrier to using Signal.

The greatest barrier is undoubtedly convincing others to use it, and even that is changing with the current political and social climates.

Even so, Signal is gaining popularity with a speed that few comparable apps can match. I suspect that the secret of its success is that it hides the complexity of encryption from users who simply want its services. Just as importantly, even without encryption, Signal is an efficient messenger, replacing preinstalled apps without a problem, and placing all communications in a single display. Through these tactics, Signal makes encryption a feature that anyone can use – and, in doing so, sets an example for the entire industry. ■■■

INFO

[1] Open Whisper Systems: *https://signal.org/*

[2] Signal Desktop: *https://whispersystems.org/blog/signal-desktop/*

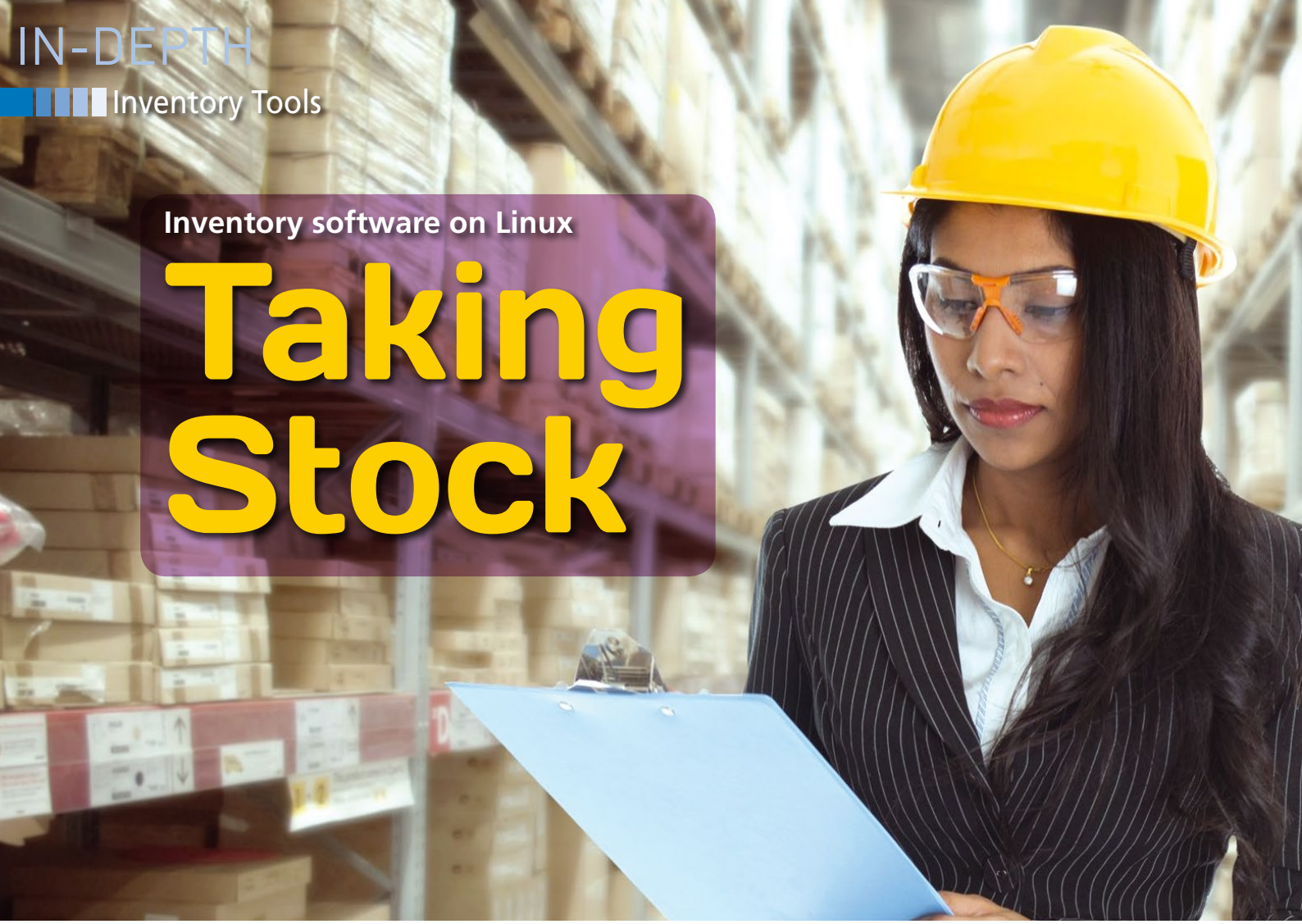
[3] Pretty Good Privacy: *http://www.pgpi.org/*

[4] Safety numbers: *https://www.whispersystems.org/blog/safety-number-updates/*

[5] Signal at Google Play: *https://play.google.com/store/apps/details?id=org.thoughtcrime.securesms*

[6] EFF on installing Signal with Android: *https://ssd.eff.org/en/module/how-use-signal-android*

[7] Signal on Chrome: *https://chrome.google.com/webstore/detail/signal-private-messenger/bikioccmkafdpakkkcpdbppfkghcmihk*



Inventory software on Linux

Taking Stock

As a network grows in size, it becomes increasingly difficult to keep track of hardware, software, licenses, and infrastructure. Inventory solutions can provide significant relief.

By Erik Bärwaldt

IT administrators are responsible for computers working smoothly. As a company expands, keeping an eye on individual hardware components, cabling, software packages, licenses, and operating systems becomes increasingly time consuming. Having multiple locations aggravates the situation.

Inventory solutions that deliver a precise overview of the installed hardware and software and take into account the entire lifecycle of an installation (including maintenance, support contracts, and planning) promises some relief. This month's column looks at three of these software tools.

Standards

The most professional inventory solutions are geared toward the IT Infrastructure Library (ITIL) standard. The ITIL standard, developed in the 1980s, originated in the United Kingdom. It comprehensively describes the components of IT service management [1]. Meanwhile, the requirements have evolved constantly to take account of technical development. Professional inventory and monitoring packages are therefore often ITIL certified.

The data stored in the databases of a configuration management database (CMDB) system are referred to as configuration items (CIs); this does not mean technical data, but it includes information relevant to controlling and accounting. Therefore, the databases also often contain the time values of objects and scenarios for planned purchases.

Functions

To cope with the diversity of inventory information in complex networks, the programs in question work with database back ends. To allow cross-platform use of the information, the front ends are usually browser-based and require no dedicated client software. However, client agents are occasionally used to provide the client data to the server for automated inventorying.

Vendors usually offer several variants of their solutions, which are often implemented as modules and extensions: Simpler versions for small networks thus do not usually integrate license management, although this plays an important role in IT infrastructures that rely on proprietary components. Also, IT security solutions designed to detect vulnerabilities in larger intranets are only available in premium packages or even as separate variants.

However, the range of basic information the packages measure is the same for all solutions: In addition to the hardware, the operating systems and installed software packages and services (including the versions) are listed. Inventory thus

also partly covers configurations and – in the case of business solutions – rights, users, and folder structures.

Additionally, the Layer 2 network topology can typically be visualized. Many of the inventory applications can be used in virtual environments, because the dedicated solutions are only certified on a few Linux distributions. The data obtained can be converted to other formats – primarily the widespread CSV format – and thus also archived to meet legal requirements. The server systems are implemented on a LAMP basis.

Some variants of the solutions tested here offer facilities for administrators that go well beyond simple stock taking: They can be used on larger networks to handle fully automated updates of individual workstations and, if needed, install operating systems from scratch.

Additionally, the applications support automated software package distribution. This software deployment function removes the need for time-consuming installation and configuration of software on individual workstations and thus saves money.

I looked at three inventory solutions, taking into account both ergonomics and feature scope. All the solutions presented here are based on Linux servers, but they can naturally also be used in heterogeneous environments.

I-doit

Developed and distributed by Synetics from Düsseldorf, Germany, since 2005, i-doit [2] sees itself as an all-around solution for the fields of CMDB, IT documentation, and computer infrastructure security. The software is suitable for companies of all sizes; both smaller environments and extensive infrastructures, as found in large-scale organizations, can be mapped and managed. I-doit can be deployed either as a virtual appliance for the VirtualBox, VMware, and Virtual PC virtual environments or on a dedicated server system.

In addition to simply documenting the hardware components, the application is capable of mapping complex networks on the basis of Layer 2 to Layer 7 connections, including IP address management. For users in heterogeneous environments, it comes with integrated license management that ensures legally compliant licensing of proprietary software at all times. The ability to display the acquired data visually, including individually generated reports, rounds out the feature set.

The software also impresses with centralized rights management and automated workflows that significantly facilitate the administrator's everyday life. Of course, the product is suitable for use in large organizations with multiple locations that may have their own infrastructures.

Costs

I-doit is available in a number variants; a 30-day trial period is available on registration [3]. The trial version can be hosted on-premise or online as a web service. A demo system with prefabricated datasets also is up for grabs. The commercial Pro version is based on a subscription model, wherein the yearly charge is calculated on the basis of the number of objects.

All systems on the intranet (i.e., both server and client machines, but also temporarily connected laptops, smartphones, printers, and IT infrastructure components) count as objects. Licenses or software packages also count as objects [4].

Modules that supplement the selected basic package's feature scope can be purchased; again, the number of objects on the intranet defines the price. Synetics bills you separately in the scope of support agreements, and the company offers training, for which they charge somewhere between EUR2,200 (~$2,300) and EUR4,600 (~$4,900) for one, two, or three days, respectively.

Documentation

Detailed software documentation is important, particularly for newcomers without previous CMDB systems experience because of the many tasks and, in part, strict legal requirements that need to be considered. The vendor takes various approaches to handling this in i-doit: In addition to a software installation and configuration manual [5], which is implemented as a wiki, they also provide information on special functions and modules on the company's website in the form of the doIT Better series.

User groups, which offer conferences and meetings, provide users the opportunity to network. For licensing questions, a dedicated FAQ page [6] is available; its technical counterpart is implemented as a wiki [7]. The content is available in English and German.

Installation

Installing i-doit is not easy because of the many requirements. Thus, the developers provide detailed documentation that lists the numerous services and applications required before the installation. However, a few stumbling blocks remain. The current version of Ubuntu 16.04 LTS (Xenial Xerus) is not supported, and the latest PHP version 7.0 thus remains sidelined. You might also need to install the recommended database back end, MariaDB, from third-party sources, depending on your choice of distribution. Older web browsers on the clients can also cause serious problems, and you might need to update.

Because i-doit runs on a LAMP system with Apache, MariaDB, and PHP, these servers need to be installed before installing the i-doit server. You then copy the i-doit ZIP archive to a directory that can be accessed by the Apache web server. After adjusting the permissions that grant the web server read and write access to the directory, you can call the graphical setup menu in the web browser; it then guides you through the configuration in a few steps.

If you decide to deploy i-doit as a virtual appliance, make sure the system resources are sufficiently dimensioned. As the minimum hardware requirements, the vendor cites 2GB of RAM and 10GB of mass storage, but the recommendations for smaller organizations are 8GB of RAM and 50GB of disk space. Also, a dual-core processor is imperative. If you are running i-doit as a virtual application, of course you need to add to the resources required for the host system.

A bug in the virtual test appliance was reproducible on all test systems using several versions of VirtualBox: The virtual machine is configured with the wrong operating system ID and with an insufficient graphics memory allocation of just 4MB, thus making it impossible to launch.

To remedy this, access the *General* section of the VirtualBox configuration set-

tings and, in the *Type* and *Version* selection fields, replace the existing values with *Linux* and *Debian (64-bit)*. Additionally, you need to increase the value for graphics memory to at least 8MB in the *Display* section. I-doit only launched after we did this, taking us to the initial configuration in a simple ncurses screen immediately after logging in with the authentication data *idoitadmin* as the username and *idoit* as the password.

After adjusting all the options, you can then access the i-doit dashboard using any web interface. Note that after entering the CMDB server's IP address in the browser, the default authentication credentials are *admin* (username) with a password of *admin*. Once you see the dashboard, access *Manage* | *Manage License* to load the license key. The software is then ready for use.

Being Objective

I-doit groups the inventory in objects that it stores in the CMDB. You will find matching object types in the objects view on the dashboard to help you with this, pre-sorted by categories such as software or infrastructure. The GUI displays the objects in an intuitive tree structure on the left of the program window in the *Object view* tab. You can add a new object by left-clicking on the appropriate type in the object view.

A detailed view of the selected type's existing objects then opens on the right in the large panel of the program window. This view is initially empty. To add a new object of the active category, you need to click on *New* above the view, and a corresponding entry screen then appears. Use this to enter the required data.

Many selection fields are filled with default options. But because these often fail to reflect object-specific data such as the manufacturer name, serial number, or type designation (especially when you launch the software for the first time), you can add these data by pressing the *Add new value* button, which you will find on the right of the active selection field in each case. Use the clear-cut dialog to enter the new options and then save your changes (Figure 1).

I-doit offers the option of considering device locations in the *Location view*. Thanks to this option, you can centrally enter and manage the systems for a site (e.g., a branch office of a larger company) in this context using the reporting function. For components that correlate to other objects of the CMDB system, you can also create appropriate mappings as required: For example, when entering the data for computer systems, you can select the operating systems or application software from the existing *Software* object group, including license and service management. Of course you can archive the data – say, on retired hardware – to comply with legal requirements.

Import and Export

Basic data resources from the intranet can be automatically loaded using the Nagios interface and the Check_MK [8] plugin, which you can access via the Extras menu. The individual data records can be retroactively modified after pressing the *Edit* button. I-doit also offers blank text fields in virtually every category so that you can add further explanations.

To import data, the *Extras* | *CMDB* | *Import* menu gives you the option of loading lists in various data formats. It also supports CSV and XML data. Individual data records can also be prepared for use in third-party applications by pressing the *Export as CSV* button.

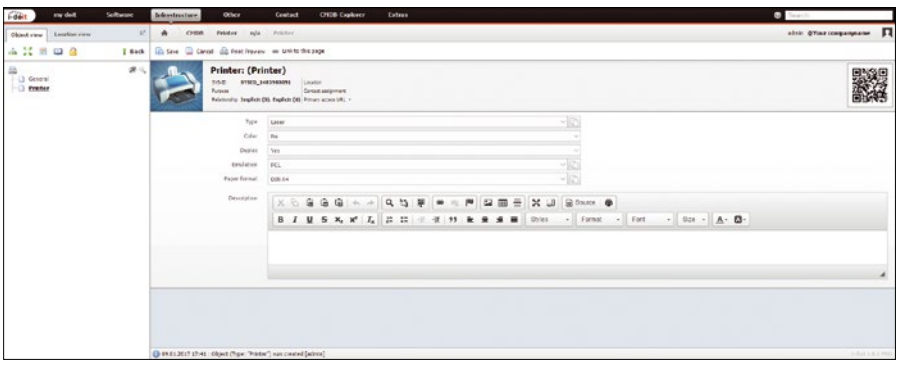


Figure 1: I-doit enables highly detailed data entry.

Reporting

Like any professional CMDB system, i-doit offers a reporting function that lets the administrator define and output individual reports. The corresponding dialog for creating the report structures can be found in the *Extras* | *Report Manager* menu. If needed, individual components included in an object group as object components can be queried in a report. This makes it easier, for example, to inventory additional hardware or software licenses. You can also add contract and vendor data.

The system can also map the geographical situation so that administrators, especially in larger organizations, will quickly have an overview of the existing structures (Figure 2).

opsi

Developed and maintained for years by uib GmbH from Mainz, Germany, the opsi (Open PC Server Integration) [9] client management system is available as free software under AGPLv3. The package can be considered a genuine all-arounder in the field of intranet client management: In addition to inventorying hardware and software, the application can also handle the installation of complete operating systems, taking into account the typical Windows versions as of Windows XP. Opsi does not support other systems such as Mac OS X and BSD derivatives, but it does at least support Linux through the use of netboot.

Additionally, the package offers software deployment, even across different sites, and can host software repositories for installing clients at remote locations. Some of opsi's functionality (e.g., a local image backup or the Nagios interface, as well as license management) is still in development and is programmed as a cofunding project.

uib GmbH offers software updates and patch solutions for opsi, partly in the form of commercial subscriptions based on the client count on the respective intranet. Also, a commercial support model can be tailored to customer needs on the basis of detailed service descriptions [10]. Various workshops and training courses complete the support offer.

Installation

The opsi server can be installed in different ways: The manufacturer offers its

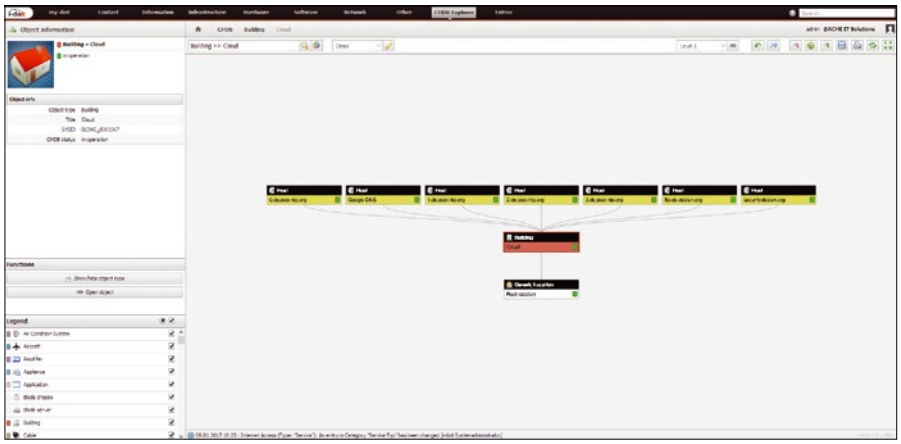


Figure 2: Without further ado, i-doit immediately gives the admin an overview of the existing resources.

own images for VMware Workstation Player or VirtualBox and VMware ESXi virtual environments, and the manual contains a list of server operating systems on which the opsi server runs, according to the manufacturer. In addition to the latest Ubuntu LTS version, they include Debian, openSUSE, Red Hat Enterprise Linux, SUSE Linux Enterprise Server, CentOS, and Univention Corporate Server. The developers also say you need 2GB of RAM and a dual-core processor as further conditions for using the server.

Problematic

For this article, I tested the VirtualBox variant of the server on a dedicated Ubuntu 16.04 system. Although the installation of the pre-built virtual machines was completed in a few minutes, the complex and time-consuming installation procedure of server version 4.0.7 on a freshly installed Ubuntu failed because of missing dependencies and a resulting termination of the routine, which was reproducible on two different computer systems. It appears that the documentation has not kept pace with the technical developments.

In tests with other Linux derivatives not on the manufacturer's compatibility list, including the Ubuntu-based Linux Mint 18 "Sarah" and Mageia 5, we were unable to talk the opsi server into installing or cooperating; administrators would thus do well to adhere strictly to the manufacturer's specifications. To compensate for this, the vendor offers exemplary manuals, sometimes in multiple formats, in which the individual installation steps are described [11].

Unfortunately, when it comes to installing the server on a dedicated machine, it is described relatively late in the manuals, and not very prominently, that a Java runtime environment version 7 must be installed if you want to use the management interface directly on the server. Opsi is not picky and also works with the OpenJDK run time.

Virtual Machine

If your users rely on something other than the certified Linux distributions, you can still use opsi in a virtual machine on one of these systems. In our lab, the current version ran without any problems in VirtualBox on a host computer with Linux Mint 18. However, pay attention to installing the network correctly: The VirtualBox LAN options must be set for network bridge operation.

On the VirtualBox machine, the opsi server is set up in a general way when first launched and adapted to the network infrastructure using multiple ncurses dialogs. Authentication credentials need to be created for the users *root* and *adminuser*.

After rebooting the virtual machine, *adminuser* logs on to the system, launching a fully customized, very lean Lubuntu system. In a browser window, which also opens, you see instructions and links to the very useful and comprehensive 350-page manual, which explains the further configuration steps and use.

To call the opsi client management system, you need to enter the address *https://<your_Opsi_server>_IP:4447/configed.jnlp* in your browser. It should be noted that the corresponding website is only rendered correctly on any computer on the intranet if the IcedTea-Web extensions are already installed; as an OpenJDK plugin, these extensions allow the execution of Java applets in your web browser. These can be found in the software repositories of popular Linux distributions; however, distributors tend to use different names, so more research might be required depending on the distribution.

It is easier to call the page on the opsi server: To do so double-click the *opsiconfiged* launcher on the LXDE desktop. This opens the server configuration interface (Figure 3).

Windows Clients

As the first step for working with the opsi CMDB system, you need to acquire the client data. If your server lives in a Windows environment, then you first need to install the opsi client agent on your Windows clients [12]. To prepare the client agents on computers without a PXE boot option, first download the opsi client boot CD-ROM as an ISO image and burn it on a CD-ROM [13].

During the download, keep the hardware architecture in mind: The ISO image is available in versions for 32-bit and 64-bit architectures. After launching the correct client agent CD-ROM, you can then integrate the respective Windows clients with your opsi universe.

Linux Clients

According to the manufacturer, an agent is under development for Linux that will support automated operating system installation, and the installation and configuration of software applications, in addition to the inventory function. Parts of the Windows agent are portable to Linux; other components need to be developed. This client agent for Linux will then be offered as a commercial extension of the opsi server that is being created in cofunding with customers. For users with up to 500 clients, the charge is around EUR2,200 (~$2,300) [14]. For larger installations, you need to ask the manufacturer about the price.

Inventory

For hardware inventory in Windows systems, opsi uses Windows Manage-

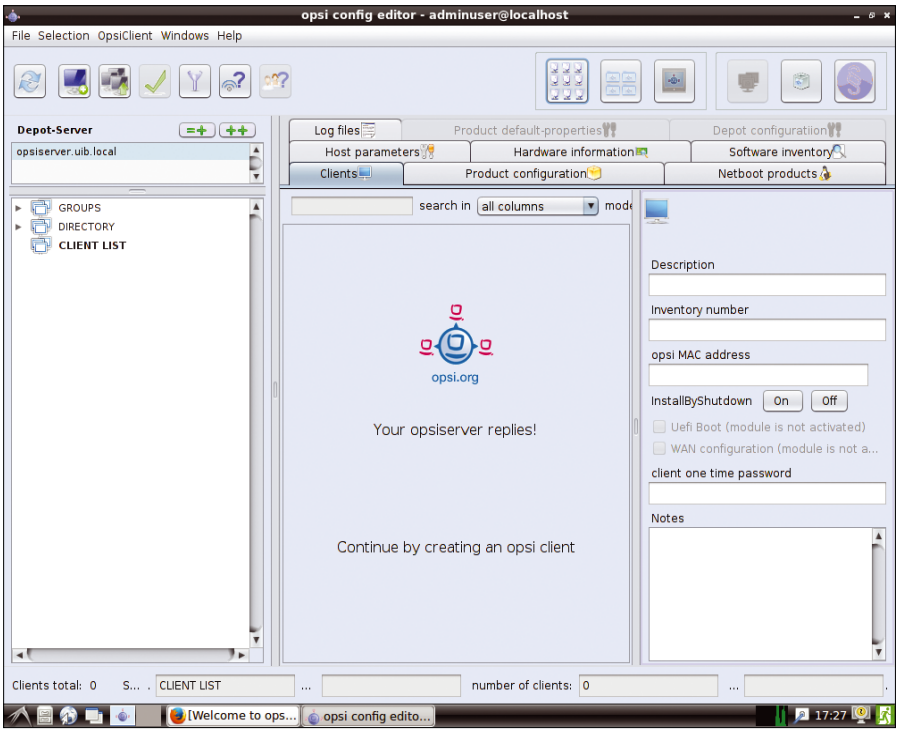


Figure 3: The opsi server's configuration interface looks somewhat cluttered.

ment Instrumentation (WMI) and reads the client software from the registry. On Linux, the software data is read from the distribution's package manager. Hardware inventory relies on boot image methods.

Functions

Like i-doit, opsi is also based on the ITIL standard and offers further options that go far beyond the actual inventory process: They include automated operating system installation, patch management, software deployment, and – as an add-on – user profile management. A software inventory feature rounds off the package. Cofunded extensions are EUR2,200 (~$2,300) each, license management and an Nagios integrated agent can be added to the software. An image backup routine is also available as a cofunded option for about the same price.

The opsi interface provides all the necessary information in different tabs; groups, directories, and clients are visualized in a list view on the left in the program window. Context-sensitive information is available in the right panel of the window. A reporting function queries the acquired inventory data, sorted by category, and processes the data. A CSV converter is also in place (Figure 4).

Open-AudIT

From Australia's Opmantek [15], Open-AudIT is one of the less well known IT inventory solution packages in Europe. The program requires a LAMP environment on Linux – as do both the other packages discussed in this article – and is available as an installation script of 35MB, which you can download from the Internet.

Like the other inventory programs, Open-AudIT will not currently run on Ubuntu 16.04 LTS (Xenial Xerus), because it is designed for PHP 5.x. Moreover, Open-AudIT requires a 64-bit operating system. In our lab, I was able to install Open-AudIT on Ubuntu derivatives such as Zorin OS 9.0, however. To use the web interface, make sure that the browser supports HTML5.

In addition to the community version, which only has a basic feature set and does not include any support, there is also an Enterprise variant, which is designed for organizations of various sizes depending on the subscription model. Customers also have access to commercial support service in the Enterprise version.

The prices for the Enterprise versions vary, as a function of the number of node objects, between $250 per year (up to 100 nodes) and $800 per year for a maximum of 500 nodes (US$). For large companies with an even larger infrastructure, individual packages can be put together on request. Node means a complete device, such as a workstation, but not a single component, such as a hard disk or a processor.

Installation

After downloading the script, first type the command

```
chmod 755 <script_name>
```

to make it executable. Then, you can call the script. One prerequisite for the installation routine is a previously installed functional LAMP environment. The script copies the files to the appropriate directories, sets attributes and parameters (if necessary), and launches the dae-

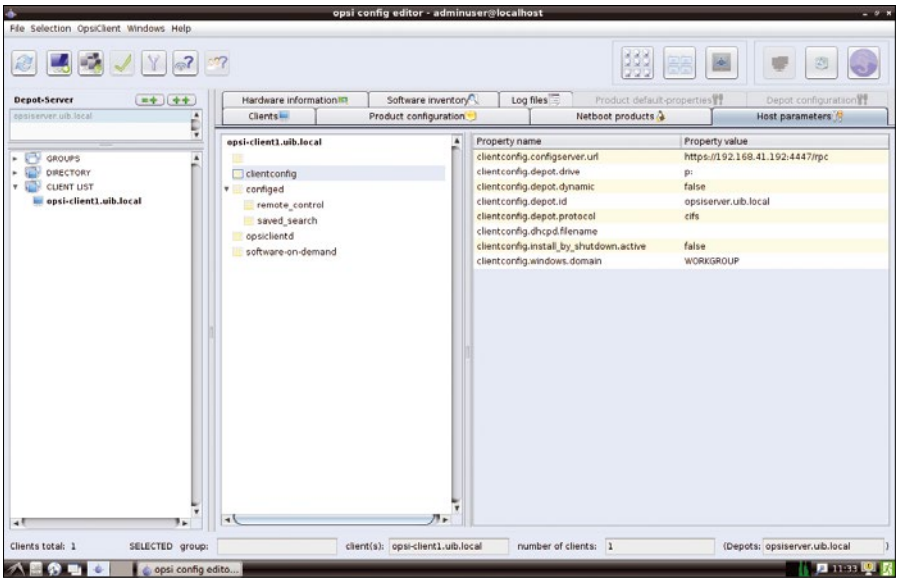


Figure 4: Opsi neatly groups data in tabs.

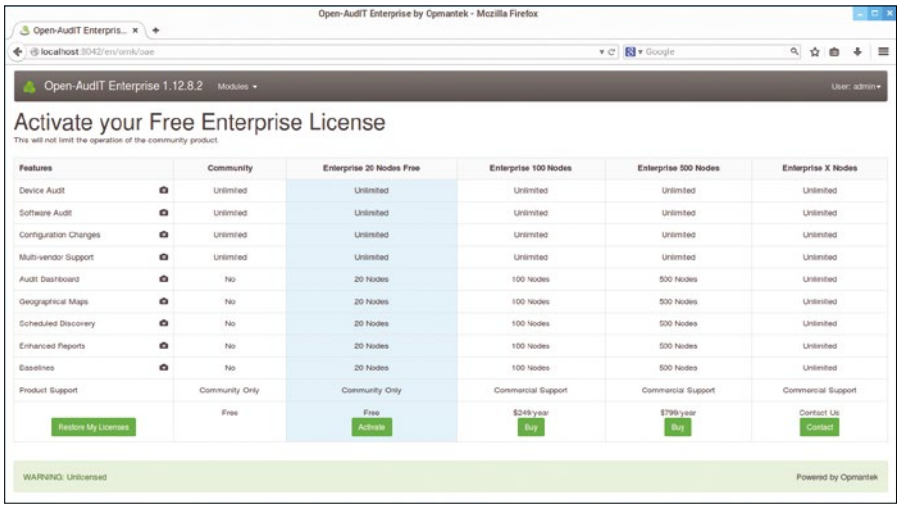


Figure 5: After calling the Open-AudIT dashboard, you will find an overview of all available versions of the program.

mon. The routine prompts you to approve the individual work steps in a user dialog in each case.

After a few minutes, the system is ready for use, and you can call the web interface in your browser by entering the URL *http://localhost:8042*. After logging in with username *admin* and password *password*, you reach an overview page showing the Open-AudIT variants. The user has the option of enabling the Community variant or the Enterprise version with a capacity of up to 20 nodes. It is immediately enabled after you enter some additional information, such as your name and email address (Figure 5).

Inventory

The software then goes to the main screen. Open-AudIT checks the server configuration during the call and immediately shows you any problems in the bottom left section of the window, if the check revealed any. In many cases, it will warn you that Nmap is missing; it needs Nmap to detect the network infrastructure. After installing Nmap and reloading the page, you can start acquiring data.

Open-AudIT supports both manual and automatic detection of existing components. To start the automatic scan, first click *Views* | *Discover a Subnet* in the menubar – which is horizontal with a black background – at the top of the program window. The software now prompts you for the of the subnet to be scanned in a clearly arranged dialog and then branches to the Log Viewer. In the Log Viewer, you can track the progress of the scan (Figure 6).

Acceleration

Depending on the size of the subnet, this initial scan can take some time to complete, but because the configuration mask also lets you input an IP address range, you can accelerate data acquisition, in particular on smaller networks, by specifying the first and last IP addresses to be scanned.

To enter components manually, go to the *System* | *Devices* | *Add a Device (manually)* menu. Then add the data in an input mask. Devices added in this way are also included in the software's reports and are incorporated into the graphs on the dashboard, as well.

Display Options

In many cases, it is not necessary to scan a complete subnet if you only want to retrieve some data on the installed software packages. Open-AudIT therefore focuses on the ability to modify the display below the *Queries* drop-down menu: This empowers users to retrieve many details on the hardware and software. As a result, it is easier for the administrator to plan changes to the hardware and software resources of individual computer systems.

Visualization

Open-AudIT visualizes the collected data in various forms: In the dashboard, the user sees an initial overview of the scan results, each grouped by periods. You can thus display all the devices detected by the last scan, in the last seven days, or in the last 30 days. This information can be broken down further by detected devices, operating systems, or installed software.

The *Map* item under Menu in the dashboard informs the user about locations. The *Logs* item contains the logfiles, which can help with troubleshooting. The user can also access these views via the menubar by selecting the *Views* entry; the other functions here include automated scanning of a subnet or – for Windows environments – an Active Directory (Figure 7).

The *Reports* drop-down in the menubar let you show and print detailed lists on demand; on request, the pre-built templates will even show you devices that have not appeared in scans for some time. Such reports can therefore provide information on lost or stolen components.

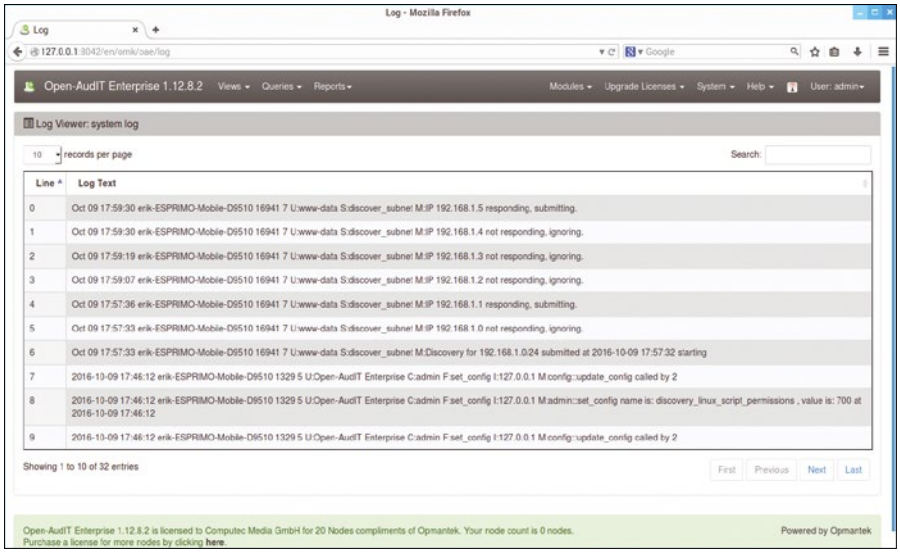


Figure 6: Open-AudIT informs you of the nodes it has found in an intuitive list.

Conclusions

The tested inventory solutions left me with a mixed impression. Although they fulfill the expected tasks, they have limitations. All three solutions suffer from a cumbersome and error-prone installation that is not state of the art in any way. I-doit and opsi provide detailed installation instructions (sometimes several pages long), and Open-AudIT comes with a usable script. However, the risk of errors is great – especially when configuring the LAMP system. In the case of Open-AudIT, the additional installation and customization of SNMP and Nmap are not properly documented.

It is incomprehensible that the developers have not adopted the far more elegant solution of a central installer, in that this solution has already been implemented for many free software packages by the Bitnami project [16].

Opsi has some catching up to do in terms of Linux: The cofunding model for the Linux agent involves significant costs, which makes it uninteresting for small businesses who want to use the system with Linux desktops. The same applies to opsi license management if you use proprietary software and to the planned use of Nagios and the image backup routine. If you need several of these modules, costs can very quickly amount to five figures, and this does not even take support services into account. The concept of client agents deserves some criticism, because it causes additional configuration overhead on the client systems.

The other two candidates show that inventory management can work without agents and that free software can offer strengths and benefits. One positive factor was the clear-cut interface in i-doit that enables immediate access and takes the network infrastructure into account. Also, i-doit provides the most detailed documentation options by far, so the tool covers the entire lifecycle of a complete enterprise IT environment. This makes i-doit the best choice for professional users with a heterogeneous IT infrastructure who view an inventory system as a data source for the accounting department. ■■■

INFO

[1] ITIL: *http://www.itinfo.am/eng/information-technology-infrastructure-library-guide/*
[2] i-doit: *https://www.i-doit.com/en/*
[3] i-doit trial version: *https://www.i-doit.com/en/trial-version/*
[4] i-doit subscription model: *https://www.i-doit.com/en/products/pricing/*
[5] i-doit manual: *https://kb.i-doit.com/display/en/*
[6] FAQs on i-doit licensing questions: *https://www.i-doit.com/en/support/faq/*
[7] i-doit technical FAQs: *https://kb.i-doit.com/display/en/FAQ*
[8] Check_MK: *http://mathias-kettner.com/check_mk.html*
[9] opsi: *http://uib.de/en/opsi/about-opsi/*
[10] opsi service descriptions and prices: *http://uib.de/en/support-training/prices-support/*
[11] opsi manuals: *http://uib.de/en/opsi-documentation/documentation/*
[12] Installing Windows clients in opsi: *http://download.uib.de/opsi_stable/doc/html/en/opsi-getting-started/opsi-getting-started.html#opsi-getting-started-firststeps-osinstall*
[13] opsi download: *http://download.uib.de/opsi4.0/boot_cds/*
[14] List of cofunding amounts: *http://uib.de/en/opsi-cofunding/prices/*
[15] Opmantek: *https://opmantek.com*
[16] Bitnami project: *https://bitnami.com*

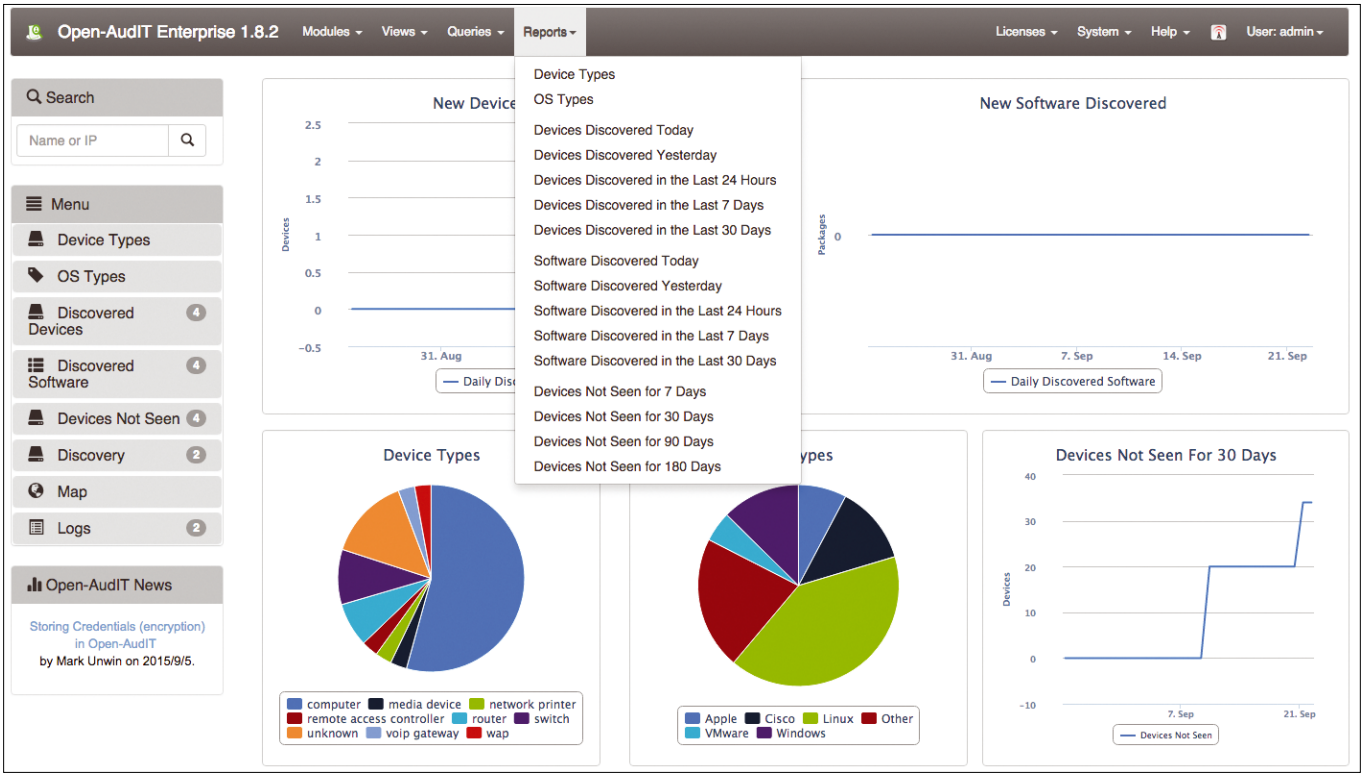


Figure 7: Open-AudIT's dashboard provides a visually appealing and clear-cut network overview.

The sys admin's daily grind: SparkFun

Trouble in the Air

Is your neighbor burning the wrong kind of wood or did a couple of VWs just pass by your house? Charly finds out with a sensor. For an attractive approach to visualizing boring measurement figures, you can either use your own web server or rely on a specialized service like SparkFun. *By Charly Kühnast*

I ordered a particulate matter sensor from smog-experienced China (Figure 1), connected it to a Raspberry Pi, and can now see with up-to-the-minute accuracy when a neighbor has a fire in their wood-burning stove and even tell if the wood was properly dried. I use RRDtool to create illustrative graphs from the particulate matter readings that I get once every minute and upload them to my web server [1].

I could even do without the web server, because number of services handle storage and visualization of measurement data for you, if so desired. One of them, SparkFun, lets you store your data simply with an HTTP call.

First, you have to register your project: Just click *Create* when you reach the website [2]. In the form, enter a title, a short description, and, most importantly, the names of the data fields you want to fill. In my case, I named them *PM10* and *PM2.5* for the number of dust particles below 10 micrometers (µm) or below 2.5µm in diameter. PM stands for particulate matter. Finally, you need to enter a web alias under which you will then view the collection of values later on.

Figure 1: The particulate matter sensor from the Far East that Charly connected to his Raspberry Pi.

Call the Locksmith

When you submit the form, you are given a public and a private key, both of which consist of random strings. You need these to transfer data to SparkFun via an HTTP call, which can be easily automated (use your keys without the angle brackets):

```
wget ➥
  "http://data.sparkfun.com/input/ ➥
  <RM736ga3vxHqMZ1qnMn2> ?private_key= ➥
  <lzaG5qwG9EIVdGRvxvx> ➥
  &pm10=$PPM10&pm25=$PPM25"
```

Then, add the line to a small script that reads the particulate matter values from the sensor once a minute.

I let 15 minutes pass and then called my SparkFun URL. The results were two neat rows of numbers – the transfer worked. To start the visualization, I then pressed the top-right button *Export to Analog.io*. On the next page, I then checked the values I wanted to display – *pm10* and *pm25* – and then finally pressed *Load All* in the top right-hand corner. After a pause for thought, the graph shown in Figure 2 appeared.

If you mouse over the graph, you can pick out some interesting data points. The lower, smaller graph is used for zooming. All in all, this method is a good alternative for people who occasionally find themselves out of breath, but do not want to run their own web server. ■■■

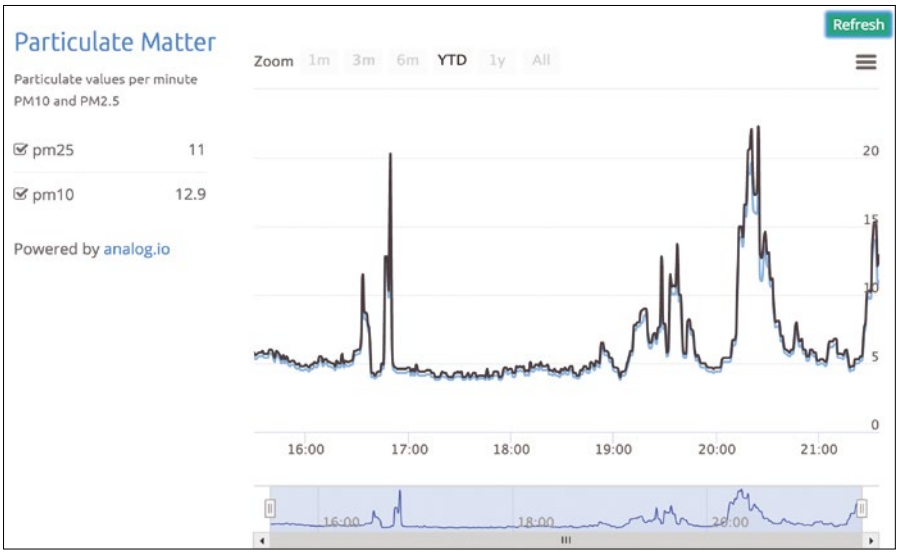


Figure 2: The free SparkFun web service visualizes the readings passed to it.

INFO

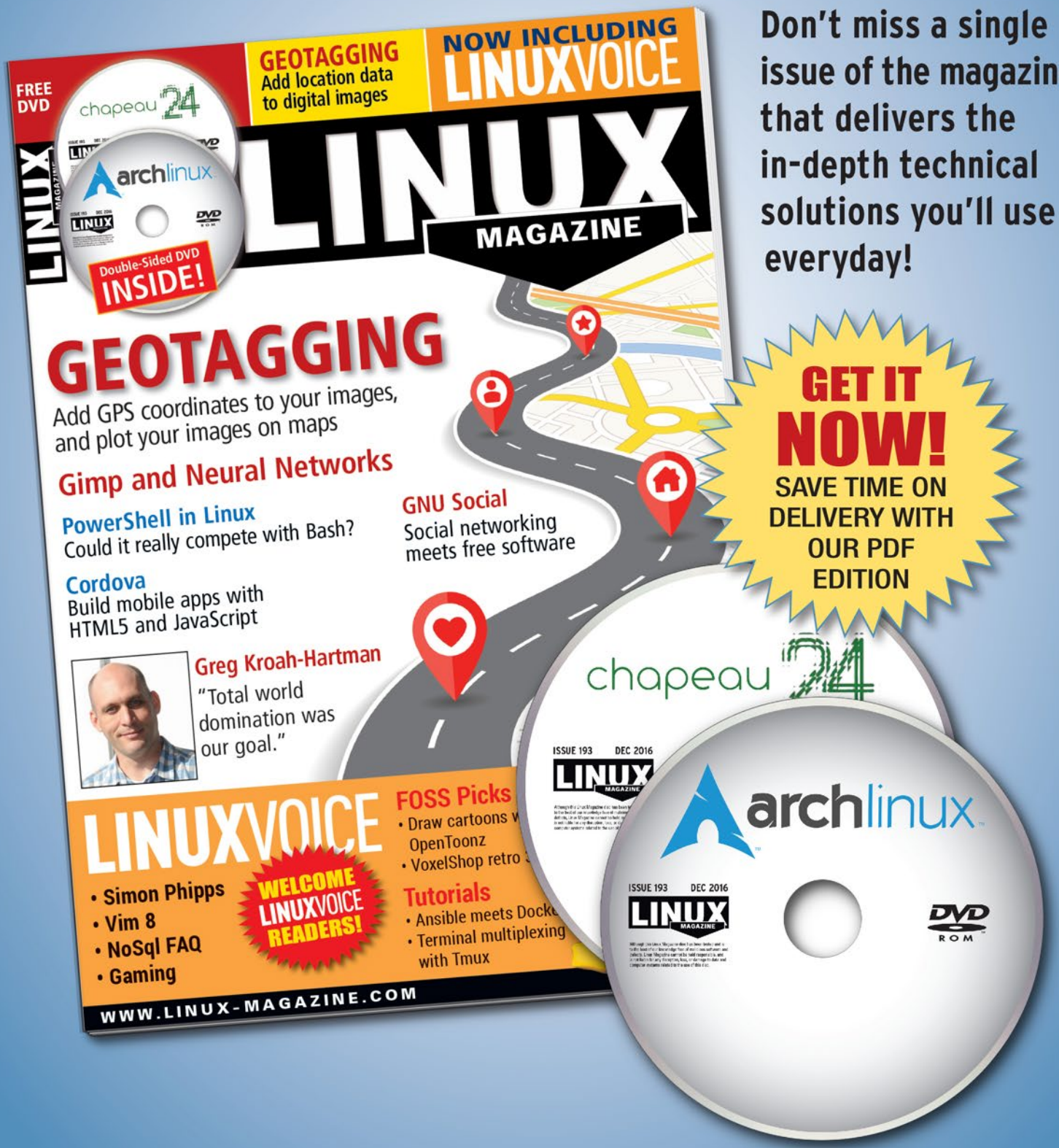
[1] Charly's "Particulate matter" page: *http://kuehnast.com/fs/*

[2] SparkFun: *https://data.sparkfun.com*

CHARLY KÜHNAST

Charly Kühnast manages Unix systems in a data center in the Lower Rhine region of Germany. His responsibilities include ensuring the security and availability of firewalls and the DMZ.

Subscribe now!



Don't miss a single issue of the magazine that delivers the in-depth technical solutions you'll use everyday!

shop.linuxnewmedia.com/subs

Social networking the FOSS way

The Email Upgrade

Forget email: Bitmessage harnesses the power of public key cryptography to create a decentralized, trustless P2P communications protocol. Messages are virtually impossible to spoof or tap. *By Nate Drake*

Users of the pseudonymous cryptocurrency Bitcoin will know that its strength lies in a blockchain – a decentralized ledger of transactions shared across thousands of computers. Since transactions are confirmed several times, it is highly unfeasible for anyone to forge an entry in the blockchain to give themselves a digital wagonload of Bitcoins. Nor is it very easy to steal coins from another user's digital wallet without their digital private key [1].

Like Bitcoin, Bitmessage uses a decentralized peer-to-peer (P2P) protocol. Instead of using a blockchain to record transactions, however, Bitmessage uses complex mathematics to validate and encrypt messages. In simplest terms Bitmessage works as a vast e-mail server, albeit one that is not controlled from any one central point [2].

Developer Jonathan Warren's official whitepaper on Bitmessage [3] goes into considerable detail on how this is achieved. As an average Linux user, it's sufficient to know that each user is assigned a virtual "address" (e.g., *BM-2cSpVFB6cDxLLGUeLRy3pZTwYsujmpRzP7*) that can be used to send and receive messages. Bitmessage users can have one or a number of these addresses (Figure 1).

As with Bitcoin, which works on the basis of "wallet addresses" to receive money, you only need to provide one of your Bitmessage addresses to a fellow user to communicate. This address is in fact a hash of a public key, and as such, it's much harder for a scammer to as-

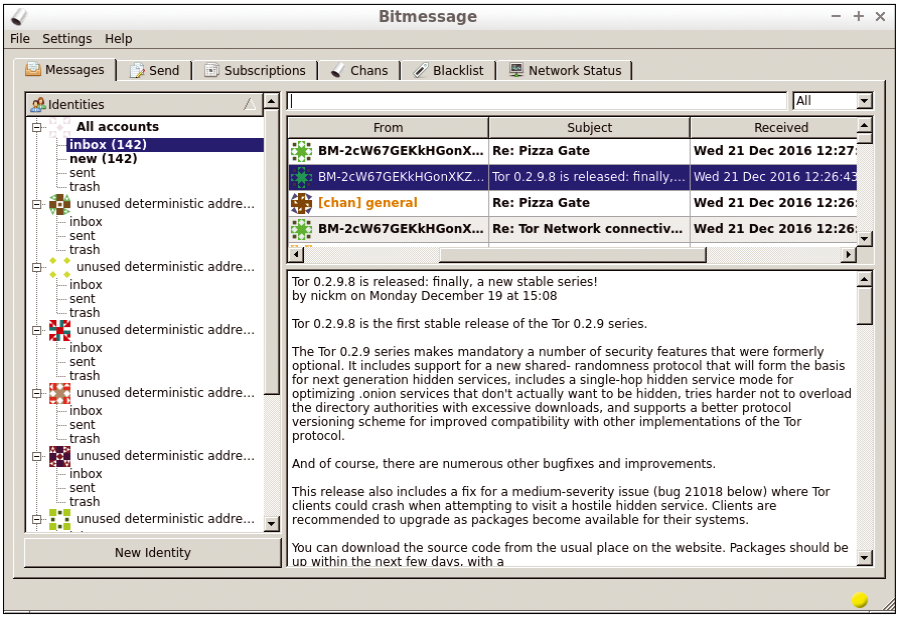


Figure 1: PyBitmessage, the official Bitmessage client in action. Users can exchange messages, as well as subscribe to news lists (*Subscriptions*) and discussion channels (*Chans*).

Lead Image © nasirkhan, 123RF.com

AUTHOR

Nate Drake is a freelance journalist specializing in cybersecurity and retro tech.

sume your identity by sending an email supposedly from your address.

Messages are transferred over a P2P network through users running the Bitmessage client PyBitmessage. The client's name is often shortened to just Bitmessage but is mentioned here to distinguish it from the Bitmessage protocol itself.

To prevent the network from being overrun by selfish users and spammers, a proof-of work must be completed for each message proportionate to its size. Just as Bitcoin users have access to all transactions, all Bitmessage users have access to all messages through their clients. However, they can only decrypt messages that have been sent to their own address.

Installing PyBitmessage

If terms like "partial hash collision" and "decentralized" fail to excite you, rest assured an in-depth knowledge of the protocol is not required to download and make use of Bitmessage's client.

Linux users can easily clone the Pybitmessage source code and run it in Python by following the instructions on the Bitmessage wiki [4]. You most likely will have the necessary prerequisites installed already on your system, such as `python` and `openssl`.

Once the software is downloaded, simply run the Python script with:

```
~/PyBitmessage/src/bitmessagemain.py
```

A pop-up appears explaining that PyBitmessage won't connect to anyone until you allow it. If you're happy to go ahead, click *OK* to continue. If you connect via a proxy or Tor, check *Let me configure special network settings first* before proceeding. (See the "Bitmessage + Tor" section for specific steps for connecting via Tor).

On the first run, PyBitmessage will generate a `keys.dat` file. By default, this is stored in your `~/.config/PyBitmessage` directory. Make sure to keep backups of this file or use deterministic addresses (Figure 2).

The Bitmessage Identity

Click the *New Identity* button at the bottom left of the PyBitmessage window to open the wizard to generate new addresses. These can be used both to send and receive messages. The key to Bitmessage's security lies here.

Users of Bitcoin will be familiar with the concept of generating new wallet addresses after each transaction to make payments harder to trace. The concept is similar to Bitmessage addresses. Creating and abandoning addresses is encouraged because it makes it much more difficult for an adversary to read your communications if they don't know from where they originate.

The only downside to this is that you will need a secure way to exchange your new Bitmessage address with all your contacts each time you generate them. This isn't very burdensome when you consider that you can make as many addresses as you like.

You can generate addresses either by generating random numbers or by using a passphrase. Take the time to read through this window (Figure 2) carefully about the pros and cons of such an approach. The advantage of using deterministic addresses (i.e., those protected by a passphrase) is that if anything happens to your machine, you can recreate your addresses and retrieve all messages. This is done by going to *File* | *Regenerate deterministic addresses*.

If you do decide to use a deterministic address, make sure to choose a strong passphrase. For extra security, use a string of random words generated by Diceware [5]. Store these safely on paper or in your password manager.

If this sounds like too much trouble, have the system generate an address automatically for you using random numbers. Make sure to keep your `keys.dat` file safe because, if it's lost or copied, your messages will be compromised.

Click *OK* when done to generate your addresses. By default, you will be assigned eight addresses, but you can change this as you see fit.

Ideally, have a friend go through the process separately on their machine at the same time as you, so you can send your first message.

Your First Bitmessage

The main PyBitmessage window will now appear with a number of addresses in the left-hand pane. The *All accounts* section aggregates all messages sent and received to all addresses. Below will be the unused addresses you generated earlier.

If you generated deterministic addresses, each will be listed as an *unused deterministic address*. If you generated random addresses, the Bitmessage address will display. Double-click on the name of one of these to give it a more human-readable name, such as *Jane – Work*. Click to highlight your name, and press Ctrl + C to copy your Bitmessage address to the clipboard.

If you want to set an avatar for your address, right-click on your name and choose *Set avatar*. From this menu, you can also disable an address, as well as set up an email gateway (see the "Email Integration" section).

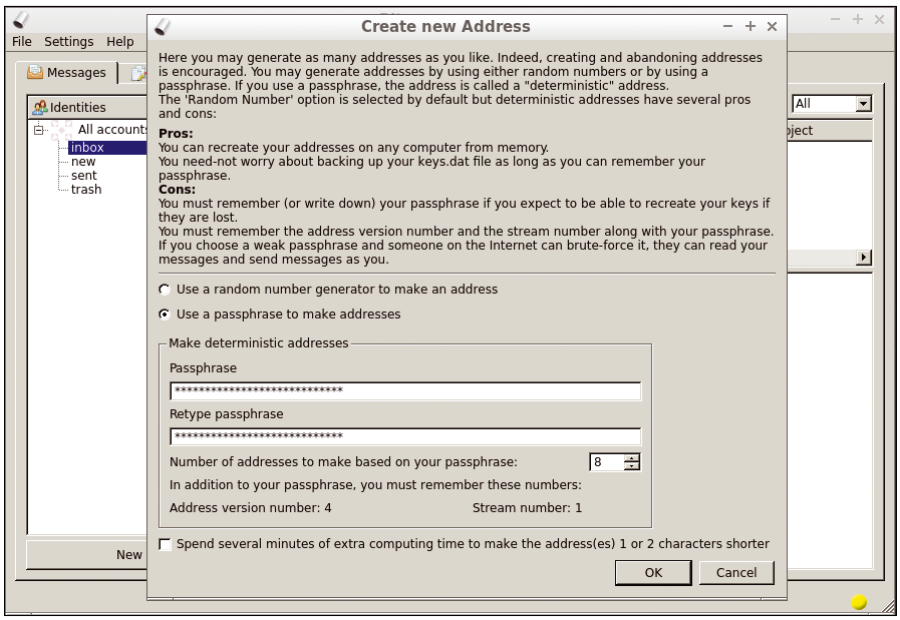


Figure 2: When creating deterministic addresses, note the *Address Version number* and the *Stream number*, as you'll need these if you have to regenerate them in the future.

Exchange your new Bitmessage address with your contacts and then click on the *Send* tab. Click *Add Contact* at the bottom left to add your friends. The *Label* field is used to provide a human-readable name (e.g., *Joe*), and the *Address* field holds the Bitmessage address.

Once your friend's address appears in the left-hand pane, right-click it for further options. You can set an avatar here if you want or stick with the one generated by PyBitmessage. Choose *Send message to this address* to prepare your first message. If you are setting this up alone, do this with the *Bitmessage new releases* address, although you shouldn't expect a reply anytime soon.

Now, move to the *Send ordinary Message* tab in the right-hand pane (see the "A Time to Live" box). Your recipient's Bitmessage address will appear in the *From* field. In the *To* field, select the address you set up previously – this will be easy to identify because it will have a friendly name and possibly an avatar.

The remaining *Subject* and body fields are self explanatory. Click *Send* to queue your message for delivery.

Bitmessage Broadcasts

By now you will be familiar with some of the concepts underlying Bitmessage, and your first message will be encrypted and working its way through the P2P network to be decoded by your recipient.

This is fantastic for one-to-one communication, but in some situations you might want to message a number of people at once, such as when sending a newsletter. Broadcasts serve as a form of universal inbox – anyone who knows the correct name and Bitmessage address can receive messages.

Click on the *Subscriptions* tab to get started. One of the more useful subs is Timeservice, which posts regular updates about the Bitmessage network. WikiLeaks also broadcasts over Bitmessage.

Give Chans a Chance

As useful as subscriptions are, they are only useful for one-way communication. They also rely on a single central client, such as the WikiLeaks server, to be constantly online and broadcast messages.

Chans (short for Channels) are one solution to this dilemma. They are a form of DML (decentralized mailing list) – this involves sharing the private key for an address between multiple clients and means that anyone can send and receive from this address (Figure 3). By comparison, it's similar to an email address to which everyone has the password.

Click on the *Chans* tab to join or create a chan. PyBitmessage will explain that the decryption keys for the chan and the Bitmessage address are derived from a human-friendly word or phrase, such as the word *hello*.

A list of the more popular chans, such as *hello* and *general*, which are for new Bitmessage users and general chat respectively, is available from the Bitmessage forums [7].

One of the chans' abiding strengths is that, because anyone is in possession of the decryption keys, they can send a message from the chan address to itself without revealing their identity. To this end, chan users will sometimes post private encoded messages, knowing the recipient will see it.

Nevertheless, you can choose to message the chan from your personal Bitmessage address if you like. For this reason, messages will sometimes appear to be *From* the chan itself or from a individual address. Righ-click Bitmessage addresses to reply directly to a sender if you like.

In light of the above, chans have built-in protection against spam and illegal content besides the work involved to send larger messages.

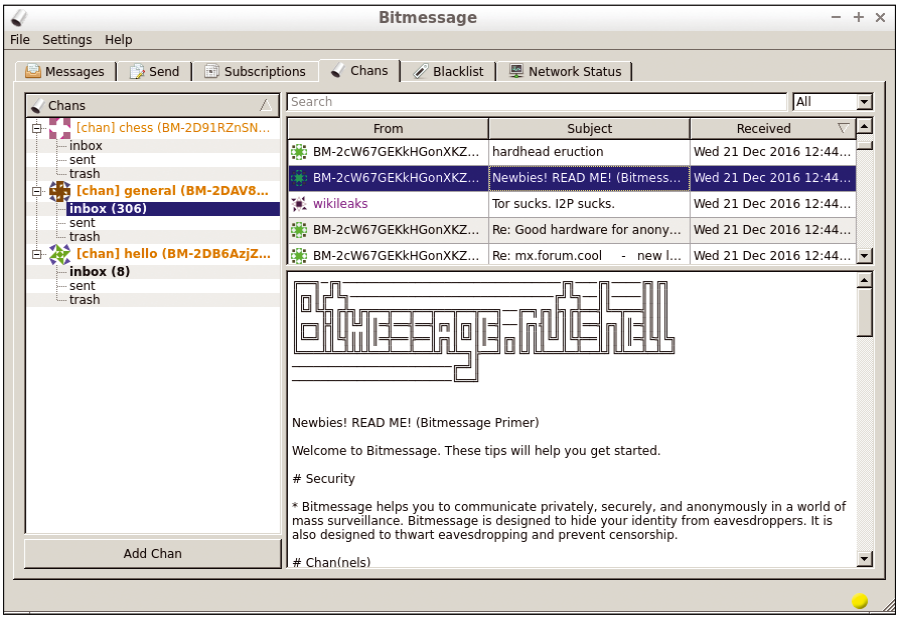


Figure 3: The *general* chan contains useful links and information for new subscribers. A full list of all registered chans is available online [6].

A TIME TO LIVE

Sharp-eyed readers may have noticed the *TTL* slider at the bottom of the *Send* pane. Time to Live (TTL) is the length of time that the Bitmessage network will retain your message. By default, that period is 102 hours. You can adjust this if you like, but the longer you want the network to hold the message, the more work your computer has to do. Once a message has been confirmed as delivered, your computer won't have to do anything further; it will be saved onto your device.

Bitmessage Blacklists

Although chans may be prone to spam, you can keep your Inbox clutter free by following a few best practices. Use the *New Identity* feature in the *Messages* tab as often as possible to provide each new party with your Bitmessage address. If the address then ends up in the hands of a scammer, simply right-click to disable.

In extreme situations where you are inundated with spam, or if you only want to use Bitmessage to communicate with a select few, then head over to the *Blacklist* tab. The interface here could not be simpler. First, decide whether you want to *Use a Blacklist*, which will allow all incoming messages except those from addresses you specify, or *Use a Whitelist*, which will block all incoming messages except those from people you have pre-

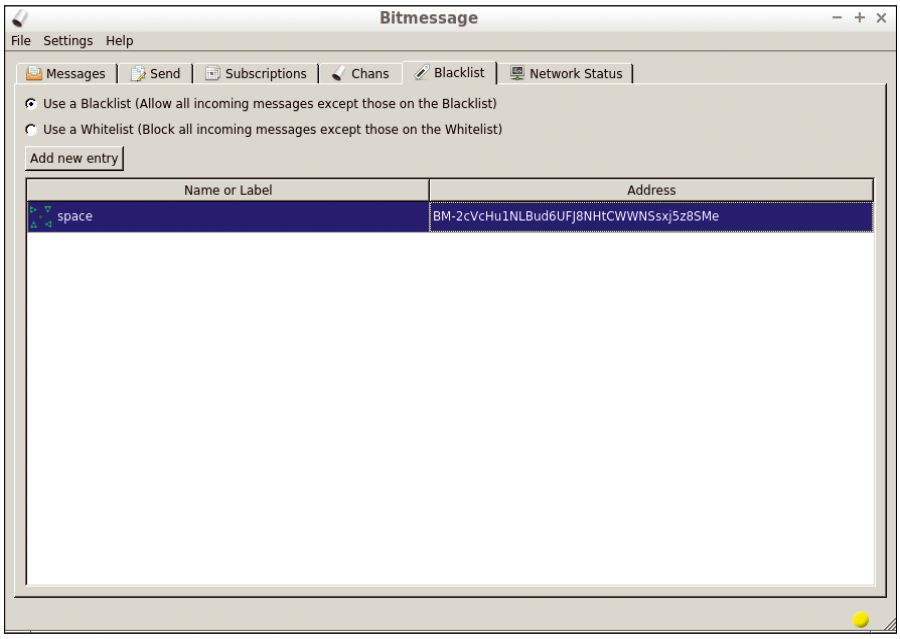


Figure 4: Click *Add new entry* to Blacklist/Whitelist specific addresses.

approved. Click *Add new entry* when you have decided to enter the label and address of the relevant users (Figure 4).

Sending messages requires computing power. The more messages and the larger their size, the greater amount of work must be done. To make the task even more difficult for those not already in your address book, head over to *Settings* and choose the *Demanded difficulty* tab. Alter the values here for larger and smaller messages to make it much harder for people who don't know you to send messages. Note that this doesn't apply to people already in your address book.

By default, the Bitmessage client will do as much work as is necessary to send a message. You and your contacts can change this from the *Max acceptable difficulty* tab under *Settings* if you want to save precious resources.

Bitmessage + Tor

As impressed as you might be with the ability to send messages easily and securely, network monitoring can still show that you are connected to Bitmessage's P2P network. Although it wouldn't be possible to know the exact content of messages you send, traffic

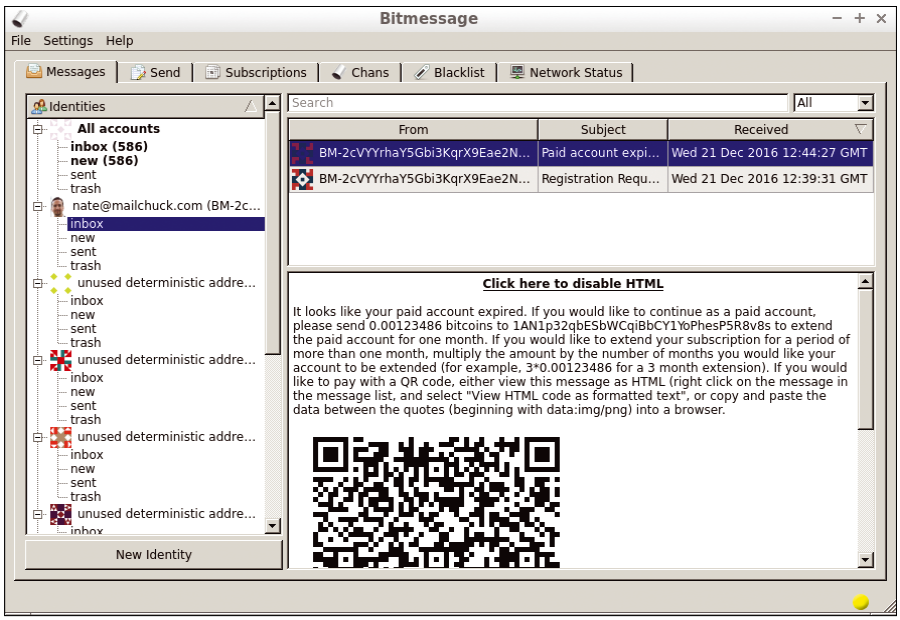


Figure 5: The first time you attempt to send a message, you will receive a Bitcoin payment address. Shortly after paying your dollar for one month, you'll be able to send messages.

correlation could be used to identify you as the sender of a message. Your location is also glaringly obvious.

If you have already downloaded and installed Tor, make sure it's running and then head to *Settings* and click the *Network Settings* tab. Choose *SOCKS5* from the *Type* drop-down menu. Leave *Server hostname* and *Port* at their default values (*localhost* and *9050*).

Press *OK* and restart Bitmessage to connect via Tor. This will naturally take longer, but it will also make your Bitmessages as untraceable as when sending email via the Tor network. For the ultra-paranoid, Bitmessage can accept connections as a hidden Tor (`.onion`) service. Specific instructions are available on the Bitmessage website [8].

Email Integration

Bitmessage is posited as a secure alternative to email. Speaking from experience, however, it's often difficult for privacy-minded individuals to bring others around to their point of view. As such, you have two ways to interface email with Bitmessage.

The first is easiest, but it does require some small expense. Right-click on any of your addresses in the *Messages* tab and select *Email gateway*. In the pop-up window, you will see the first option to register an email address. This is currently offered by the good people at Mailchuck. Enter your desired email address and click *OK*.

Bitmessage will send an *Email gateway registration request* to link your Bitmessage address with the email address you just created. With luck, you will receive a message to your Bitmessage Inbox stating that your registration request has been accepted. Make a note of the address to unregister the account.

Mailchuck also provides a relay address, explaining that you need to send email to that address, placing your recipient's email address in the subject line. Fortunately more recent versions of PyBitmessage do away with this. To send a message to an email address, simply head over to the *Send* tab. The *From* address is simply the Bitmessage address you registered with Mailchuck. Enter your recipient's email address in the *To* tab.

Although you are able to receive email free of charge, Mailchuck requires a small subscription fee of around one dollar a month, payable in Bitcoin, to send a message (Figure 5).

For those on a budget or who don't know how to get their hands on Bitcoins, the online Bitmessage Mail Gateway [9] offers a free webmail service for Bitmessage users. It allows you to create a human-friendly alias for your Bitmessage address and integrate with popular mail clients like Mozilla Thunderbird. More details are available on the site's FAQ.

Bitmessage Bummers

The moment any communications leave the Bitmessage network they are decrypted. This means sending email via the Mailchuck Gateway or receiving them via the Bitmessage Mail Gateway is no more or less secure than regular email. Try to encourage your contacts to join Bitmessage as well if you all want to communicate securely.

In terms of the PyBitmessage application itself, anyone in possession of the passphrase for your deterministic address or the contents of your `keys.dat` file can read your messages and impersonate you. Try to install the program to an encrypted volume. You can further increase PyBitmessage's security by heading to *Settings*, clicking *User Interface*, and ticking the *Run in Portable Mode* checkbox.

Portable mode ensures that messages and any configuration files are stored in the same directory where PyBitmessage is running. By default, this is the `PyBitmessage` folder in your home folder. Once portable mode has been enabled, you can then copy the entire folder to a separate device, such as a USB stick or an encrypted partition, and run it from there if you like.

Given how anyone running the PyBitmessage program can impersonate you and read your messages, one useful feature would be to protect the program and data files with a password. The developers have clearly focused on making sure that PyBitmessage is as functional as possible. As such, it may seem drab against more colorful messaging clients with downloadable skins. Head over to the Bitmessage Feature Request List if you have any suggestions [10].

Android users might want to install Christian Basler's Abit [11]. The app can recreate deterministic addresses from a passphrase or read the content of the `keys.dat`, but it must be set in *Full node* mode to work properly. The demands on data and system resources are quite extreme for a mobile phone, so do not expect this to run as well as on your computer.

Bitmessage is not and to some extent cannot be moderated. This means you may see links to harmful or even illegal content. Messages by default are shown in rich text, so links to other websites will work, but you will see a warning message. Other types of HTML, such as images, will only be shown if you click to enable it specifically.

Take time to work through Bitmessage and its features to see if it's right for you. If you run into any difficulties, in the first instance, read through the website's FAQ [12]. ■■■

INFO

[1] How does Bitcoin work?: *https://bitcoin.org/en/how-it-works*

[2] Coindesk: *http://www.coindesk.com/bitmessage-is-the-bitcoin-of-online-communication/*

[3] Bitmessage: *https://bitmessage.org/bitmessage.pdf*

[4] Bitmessage compiling instructions: *https://bitmessage.org/wiki/Compiling_instructions*

[5] Diceware: *http://world.std.com/~reinhold/diceware.html*

[6] BeamStat: *https://beamstat.com*

[7] Bitmessage address directory: *https://bitmessage.org/forum/index.php?topic=1689.0*

[8] Bitmessage as hidden service on Tor: *https://bitmessage.org/wiki/FAQ#How_do_I_setup_Bitmessage_to_work_with_Tor*

[9] Bitmessage Mail Gateway: *https://bitmessage.ch/faq.html*

[10] Bitmessage Feature Request List: *https://bitmessage.org/wiki/Feature_request_list*

[11] Abit – Android Apps on Google Play: *https://play.google.com/store/apps/details?id=ch.dissem.apps.abit&hl=en*

[12] Bitmessage FAQ: *https://bitmessage.org/wiki/FAQ*

KDE Connect links Android with the Plasma desktop

Building Bridges

KDE Connect bridges the gap between mobile devices and the KDE desktop, allowing the exchange of notifications, files, and URLs between devices. *By Ferdinand Thommes*

KDE developers have been working for more than a year to extend KDE Plasma to the mobile world. The current project, under the name Plasma Mobile, is geared to provide a free platform for mobile devices some time in the future, thus acting as an alternative to existing platforms [1]. Since Google Summer of Code 2015 (GSoC 015), an application has connected Android and BlackBerry devices with the Plasma desktop and supported some reciprocal functional control.

```
root@siductionbox:/media# kdeconnect-cli --help
Usage: kdeconnect-cli [options]
KDE Connect CLI tool

Options:
  -l, --list-devices          List all devices
  -a, --list-available        List available (paired and reachable) devices
  --id-only                   Make --list-devices or --list-available print only
                              the devices id, to ease scripting
  --refresh                   Search for devices in the network and re-establish
                              connections
  --pair                      Request pairing to a said device
  --unpair                    Stop pairing to a said device
  --ping                      Sends a ping to said device
  --ping-msg <message>        Same as ping but you can set the message to
                              display
  --share <path>              Share a file to a said device
  --list-notifications        Display the notifications on a said device
  --lock                      Lock the specified device
  --device, -d <dev>          Device ID
  -h, --help                  Displays this help.
  -v, --version               Displays version information.
  --author                    Autor-Informationen anzeigen.
  --license                   Lizenz-Informationen anzeigen.
devil : bash | () 192.168.178.20 | Downloads : bash | devil : bash
devil : bash - Drop-Down Terminal
```

Figure 1: KDE Connect from a terminal with `kdeconnect-cli`.

Spanish developer Albert Vaca aptly dubbed the application KDE Connect [2], which is available for Linux and FreeBSD, with clients for Android and BlackBerry. An iOS version is currently being built, as well. Users can meaningfully connect PCs, notebooks, tablets, and smartphones on the home network. Additionally, extensions for Firefox [3] and Chrome [4] send URLs from the desktop to Android devices. With the `kdeconnect-cli` command, you can control KDE Connect in a terminal (Figure 1).

In September 2016, KDE Connect reached version 1.0, which incorporates several important innovations. Most distributions are not packaging

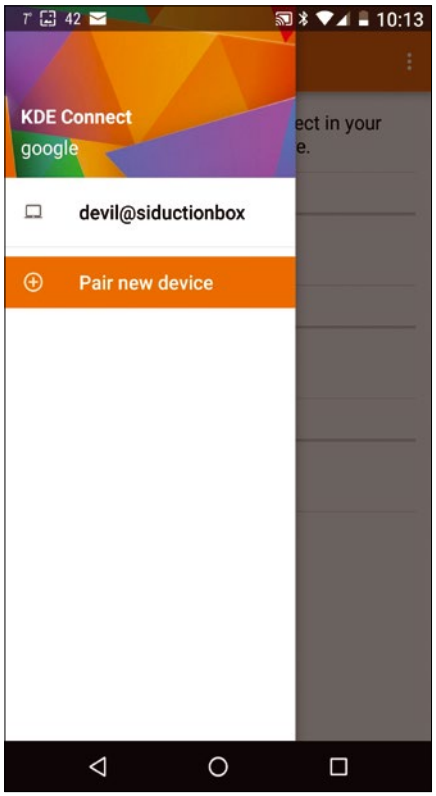


Figure 2: In the Android app, pair your mobile device with the desktop.

this version right now, but that is likely to change in the near future. The associated packages in the archives of the distributions are typically named *kdeconnect*; for Ubuntu – depending on the version – this is *kdeconnect-kde* or *kdeconnect-plasma*. If necessary, you can build the latest 1.0.1 version from source code.

Ubuntu, Linux Mint, and Elementary OS can use KDE Connect with desktop environments like Gnome, Cinnamon, Unity, Maté, Xfce, and LXDE/LXQt. In these cases, you install using the `indicator-kdeconnect` personal package archive (PPA) [5].

To install the Android app, go to either F-Droid or the Google Play Store. You need at least Android 4.1 (Jelly Bean) to use all the features. If you have a firewall, you need to open ports 1714 to 1764 for both TCP and UDP. To integrate the data of the mobile device with the desktop file manager, you also need the *sshfs* package.

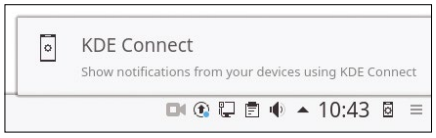


Figure 3: A plasmoid gives direct access to all the KDE Connect features.

Getting Started

After installing KDE Connect on all the devices to be connected, you should launch the Android app. From there you can pair the devices, much like you would using Bluetooth (Figure 2). Instead of Bluetooth, D-Bus, its Media Player Remote Interfacing Specification (MPRIS) [6] interface, and the Avahi Zeroconf implementation [7] connect the devices. The MPRIS interface provides an API for controlling media players, including functions for identifying, querying, and play.

The app offers to ping the respective device to test the connection; then, you can use the three-item menu at top right to enable the desired plugins. On the Plasma desktop, you will want to install the plasmoid for KDE Connect first, so you always have quick access to the application (Figure 3).

Key Features

You can now right-click on the KDE Connect plasmoid to access the configuration on your desktop (Figure 4) and enter the following settings:

- *Battery monitor*: Display and warn in case of low battery power.
- *Receive notifications*: Forward notifications on the mobile device to the desktop.
- *Remote filesystem browser*: Browse the remote file system with SSH Filesystem (SSHFS).
- *Pause media during calls*: Stop music or movies in case of a call.
- *Multimedia control receiver*: Remote control for media running on the desktop.
- *Telephony integration*: Send alerts for calls and text messages to the desktop.
- *Share and receive*: Send files from the mobile device to the desktop.
- *Remote input*: Use the smartphone as a touchpad or keyboard for the desktop.
- *Clipboard*: Share the clipboard with other devices.

In the lab test, I quickly switched off the notifications from the mobile device to the desktop, because the number of messages was annoying.

In version 1.0, KDE Connect saw the introduction of some long-awaited features. For example, you not only see a notification for incoming text messages on the desktop, you can also respond directly; conversely, the mobile device now receives notifications from the desktop. You can also define commands to the desktop, which you can then call from the mobile device. In this way, for example, you could send a notebook to sleep. Security benefits with the switch from RSA keys to TLS for encryption, ensuring that no external devices can hijack the connection.

One of the most popular features includes pausing music or movie applications, such as Amarok (Figure 5),

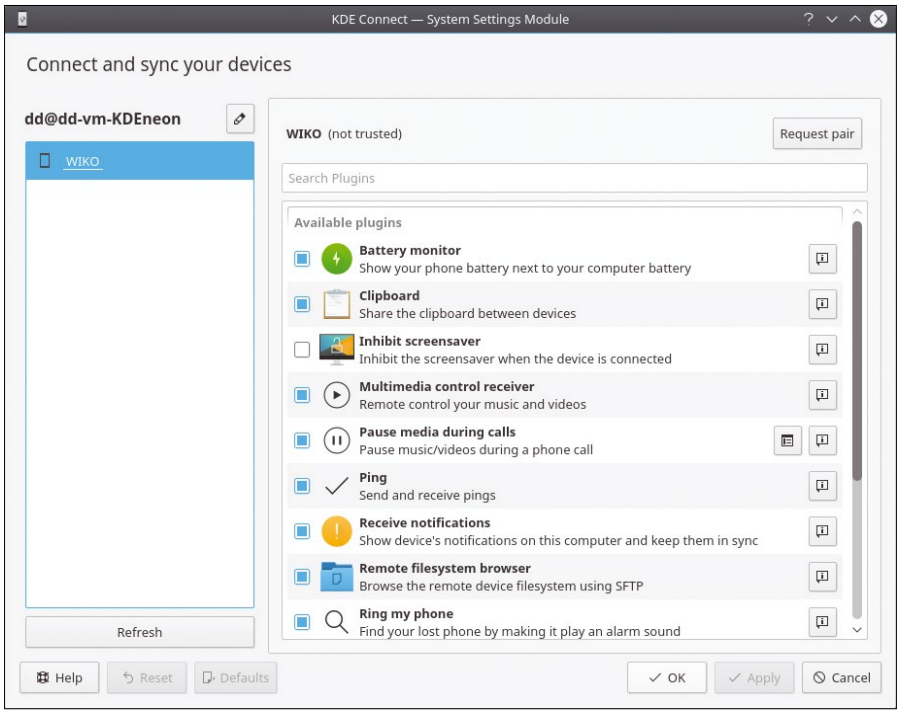


Figure 4: Right-clicking on the KDE Connect plasmoid calls the configuration on the desktop.

Clementine, VLC, or SMPlayer, when a call comes in on the smartphone. After the end of the call, the respective media continue to run. Integration with a file manager like Dolphin, Nemo, Nautilus, or Thunar is one of the much-used functions. You can use SSHFS to display the contents of the mobile device on the desktop or copy and move (using drag and drop). Dolphin automatically integrates mobile devices (Figure 6); in other file managers, you need to add them manually.

From the file manager, you can send a larger number of images for editing to the desktop quickly. This removes the need for apps like AirDroid [8], and the data does not leave the home network. Sending data, images, or URLs from the mobile device works in the respective share pages in the corresponding applications (Figure 7); transmitted pictures or data are displayed on the desktop and URLs are opened in the browser. The battery indicator on the desktop for the mobile device is very useful, as well.

The exchange of clipboard content keeps the clipboards of all connected devices synchronized in real time. However, the function is dangerous when you are outside the home LAN or a secure network, because the clipboard could contain sensitive data such as passwords that you do not want to share with others (e.g., at the workplace).

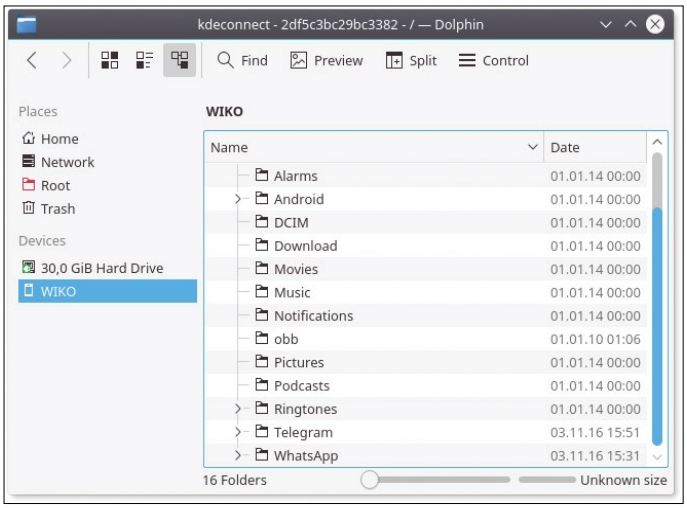


Figure 6: You can use SSHFS to display content or copy and move mobile device items from the desktop.

Conclusions

KDE Connect is literally addictive. It sets up a secure bridge over the short distance between the Plasma desktop and mobile platforms such as Android, BlackBerry, and iOS. The carefully considered application incorporates several features – especially for the KDE desktop – to which you become accustomed after just a short time.

The software is stable and constantly evolving. I can therefore wholeheartedly recommend KDE Connect for the Plasma desktop. Users of other desktop environments have to decide whether they want to accept the many dependencies in the form of KDE packages to access the abundance of features in KDE Connect. ■■■

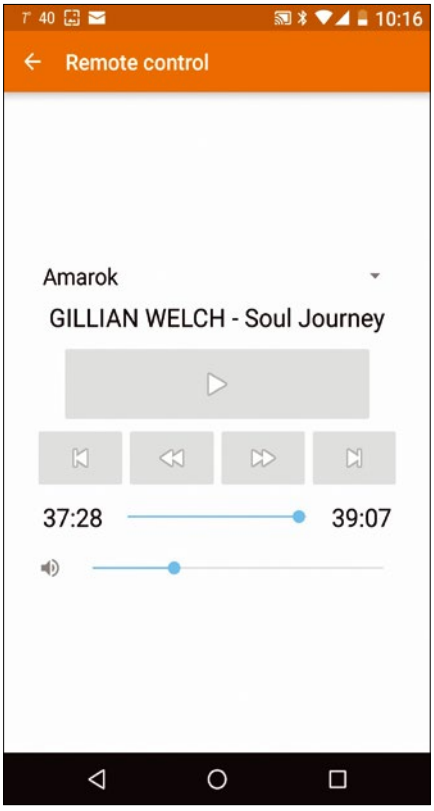


Figure 5: If necessary, you can use KDE Connect to control the desktop via the mobile device.

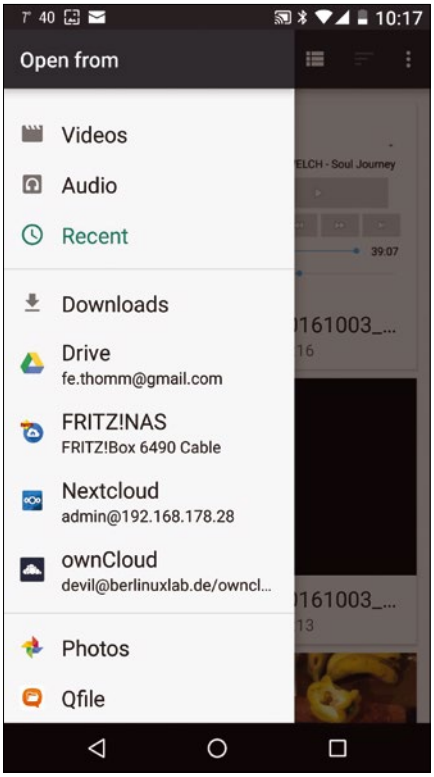


Figure 7: Sending data, images, or URLs from the mobile device to the desktop relies on the respective share pages in the corresponding applications.

INFO

[1] Plasma Mobile: *https://plasma-mobile.org*

[2] KDE Connect 1.0 announcement: *https://albertvaka.wordpress.com*

[3] Firefox: *https://kamikazow.wordpress.com/2014/11/22/send-firefox-tabs-to-your-phone-via-kde-connect/*

[4] Chrome: *https://chrome.google.com/webstore/detail/jniioigoopmlbeceondbcpgnbimeghhj*

[5] PPA: *https://code.launchpad.net/~vikoadi/+archive/ubuntu/ppa/*

[6] MPRIS: *https://www.freedesktop.org/wiki/Specifications/mpris-spec/*

[7] Avahi: *https://en.wikipedia.org/wiki/Avahi_(software)*

[8] AirDroid: *http://web.airdroid.com/*

14th Annual
2017 HPC for Wall Street – Cloud & Data Centers
Show & Conference

April 3, 2017 (Monday) **Roosevelt Hotel, NYC**
Madison Ave & 45th, next to Grand Central Station

Plan to attend, Free Show and Low Cost Conference at $295.

Plan to Attend: Featuring Leading Capital Markets providers for FinTech apps.

Keep Up with Capital Markets IT at the 2017 HPC for Wall Street.

All-Star Conference program for 2017. Cloud, Data Centers, Network Systems, Machine Learning, Virtualization, Software to speed applications, HPC, Linux, and FinTech solutions.

The FinTech marketplace will assemble at HPC's convenient and affordable one-day show and conference.

Location. Location. Location. The Roosevelt is next to Grand Central Station and within walking distance of JPMorgan Chase, Deutsche Bank, Morgan Stanley, NASDAQ – all in midtown.

Register online: Save $100. Conference only $295. $395 on site, luncheon included in the full day program.

Don't have time for the Conference? Register for the Free Show at www.flaggmgmt.com/linux

2016 Sponsors

IBM, Cisco, redhat, hp, lenovo, ORACLE, Juniper Networks, ARISTA, NYC NoSQL, Linux New Media – The Pulse of Open Source, Linux Journal, datanami, Securities Technology Monitor, Traders Magazine, Integration developer news, HPCwire, EnterpriseTech, Money management executive, WSTA Wall Street Technology Association

www.flaggmgmt.com/linux

Show Hours: Mon, April 3	8:00 - 4:00
Conference Hours: Mon, April 3	8:30 - 4:50

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

The all-star lineup of speakers from HPC 2016


Dave Weber
Global Financial Services Director, Lenovo


Ken Barnes
SVP Corp Dev, Options Information Technology


Bernard S Donefer
Associate Director, Baruch College


Mike Blalock
Global Sales Director, Intel


Andy Bach
Chief Architect, Financial Service, Juniper Networks


Jeffrey M. Birnbaum
Founder and CEO, 60East Technologies


Dino Vitale
TD Securities


Harvey Stein
Head of Credit Risk Modeling, Bloomberg


Fadi Gebara
Sr Manager, IBM Research


Terry Keene
CEO, iSys


Rob Krugman
VP Digital Strategy, Broadridge Fin Sols


Lee Fisher
VP Marketing, Redline Trading Solutions


Jeremy Eder
Perf Engineering, Red Hat


Matt Smith
Sol Architect, Red Hat


David B. Weiss
Sr Analyst, Aite


Rick Aiere
Architect Specialty, AIG


Shagun Bali
Analyst, TABB Group


Jeffrey Scheel
Senior Technical Staff, IBM Linux Tech Center


Ed Turkel
Mgr WW HPC Mkting, Hewlett-Packard


Charles Milo
Enterprise Technical Specialist, Intel


Alex Tsariounov
Principal Architect - Adv. Platforms, London Stock Exchange


Ugur Arslan
Quantative Analyst


Davor Frank
Sr Solutions Architect, Solarflare


Phil Albinus
Editor, Traders Magazine, SourceMedia

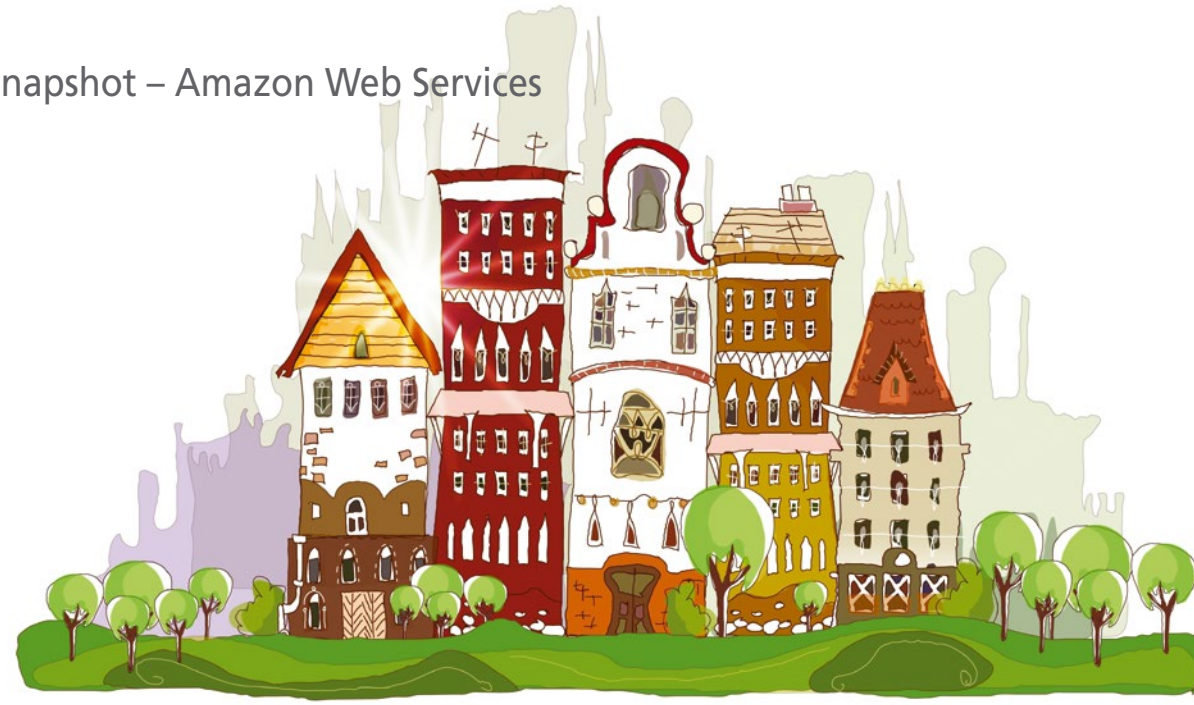

David Malik
Sr Director, Advanced Services, Cisco Systems


Russ Kennedy
SVP of Product Strategy, Cleversafe


Ryan Eavy
Exec Dir, Architecture, CME Group


Markus Flierl
VP Software Dev, Oracle


Nick Ciarleglio
Distinguished Syst. Engineer, FSI Product Mgr
Arista Networks



Set up Amazon Web Services

New Home

When applications run in a cloud system on Amazon Web Services, operators can forget management worries and concentrate instead on the essence of the app. Codemeister Mike Schilli performs the basic setup of the web service in the first part of this workshop. *By Mike Schilli*

Start-up companies attempting to shake the market in a flash and preparing for the onslaught of millions of happy users usually won't spend time or resources tending a server farm whose operation needs a knack for patches, reliability, and scaling. Streaming services such as Netflix and Spotify make no secret of the fact that large parts of their infrastructure run on rented clouds operated by Amazon Web Services (AWS). Although that makes them dependent on the operator, apparently even industry giants gain advantages by outsourcing infrastructure.

Choice

If you want to start off on a small scale and take your first few steps in the direction of cloud deployment, you first face a tangled mess of different service offerings and the emotional hurdle of credit card-based server operation. Amazon only takes your money, however, if you go beyond the scope of their free tier [1].

When I recently decided to make my surveillance video motion detection method [2] publicly available in the cloud, After reading about event-driven serverless applications [3] and building single-page apps on AWS [4], I was surprised, on the one hand, how quickly you can set up a web service at the command line and, on the other, the amazingly confusing number of configuration tweaks you need to adjust.

Storage by the Bucket

Amazon stores everything that defines a web service – the code that runs when a browser points to it and the configuration of access permissions – in its Simple Storage Service (S3). Because AWS also can deliver files to a web server as static content on request, this is often the first step into the world of clouds; more complex tasks then follow, such as setting up databases, operating back-end servers, or verifying user IDs.

At *console.aws.amazon.com*, the console is the central entry point (Figure 1) where you enter your Amazon ID; online shoppers will probably already have registered a credit card there. The *Services* tab then takes the newcomer to a page that lists a couple of dozen Amazon server offerings; you need to select *IAM* (identity and access management) here to be able to create a new user and assign the necessary permissions for cloud operation.

If you checked the *Programmatic access* box in Figure 1, you are given an access ID and a secret access key, which you can use later to configure the cloud working with your account via the `aws` command-line tool.

On the following page, you grant rights by assigning policies. From the

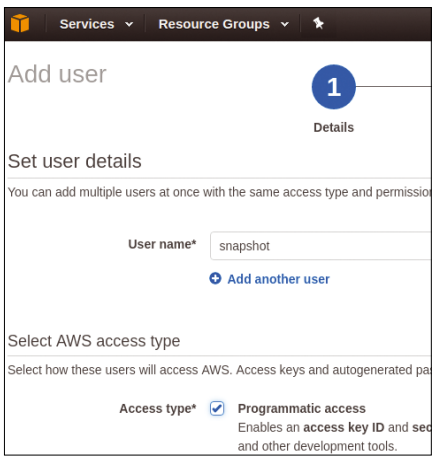


Figure 1: A new user on AWS is given the golden key to the city.

MIKE SCHILLI

Mike Schilli works as a software engineer in the San Francisco Bay Area. He can be contacted at *mschilli@perlmeister.com*. Mike's homepage can be found at *http://perlmeister.com*.

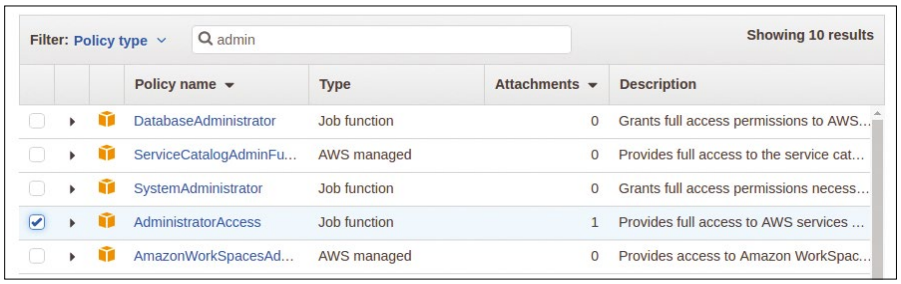


Figure 2: A user with *AdministratorAccess* rights can set up new S3 buckets.

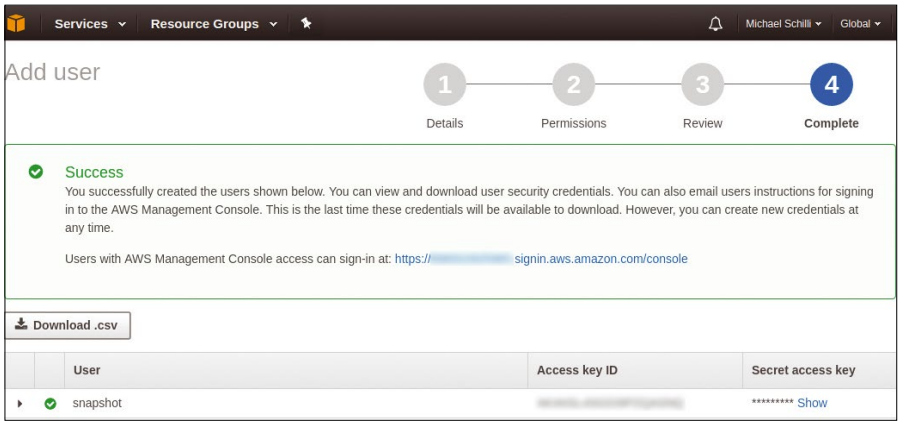


Figure 3: Using the access ID and secret access key, users can access your account with scripts.

```
$ aws configure --profile snapshot
AWS Access Key ID [None]: XXXXXXXXXXXXXXXXXXX
AWS Secret Access Key [None]: YYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYYY
Default region name [None]: us-east-1
Default output format [None]:
$
```

Figure 4: The `aws` command-line client receives the key ID and secret key for future online access and saves it locally.

tangled mess of a few dozen boxes, you will want to pick the *Attach existing policies directly* option and select the *AdministratorAccess* policy so you can create new S3 buckets in which to store your code files (Figure 2). Later on, as a good security practice, installed application policies with less far-reaching rights are available (i.e., for going live with new releases). Policies can also be combined strategically with the help of roles.

When you press the button to confirm, the console outputs an access ID for later identification and a secret access key, much like a password (Figure 3). The `aws` command-line tool itself is written in Python. On Ubuntu or Debian, you can then use:

```
sudo apt-get install python-pip
sudo pip install awscli
```

to install the tool. For it to be able to access the cloud servers later, the command-line call in Figure 4 configures the local `~/.aws/credentials` file with the values received earlier; subsequent calls to `aws` functions later on will find the access parameters there and pass them into the AWS gatekeeper.

The call also defines region `US-east-1` as the data center closest to you from among Amazon's selection of data centers worldwide. Not all data centers offer the same services; thus, you need to clarify in advance whether the nearest one meets all your requirements.

Server Trick

The sequence in Figure 5 creates a new S3 bucket to hold files (e.g., `index.html`) as a static test page and uses the `sync` command to drop the file into the bucket. The sub-command `website` then defines `index.html` as the web server's entry point and `error.html` as the error message file. After that, Amazon will happily serve up the page at the designated entry point, as if it were behind a normal web server (Figure 6).

It is your own responsibility to define easier-to-remember DNS records outside the `amazonaws.com` domain; a `CNAME` entry with your choice of provider will then point users to the cloud server.

Hidden Lambda

If you want to run programs on Amazon's back-end servers instead of just serving static web pages, you can opt for the Lambda offering. It runs JavaScript, Python, and Java functions in isolated con-

```
$ aws --profile snapshot s3 mb s3://snapshot.linux-magazin.de
make_bucket: snapshot.linux-magazin.de

$ echo '<h1>Well, hello there!</h1>' >public/index.html

$ aws --profile snapshot s3 sync public/ s3://snapshot.linux-magazin.de --acl
 public-read
upload: public/index.html to s3://snapshot.linux-magazin.de/index.html

$ aws --profile snapshot s3 website --index-document index.html --error-docum
ent error.html s3://snapshot.linux-magazin.de

$ curl http://snapshot.linux-magazin.de.s3-website-us-east-1.amazonaws.com
<h1>Well, hello there!</h1>
```

Figure 5: The `aws` tool creates an S3 bucket, copies `index.html` to it, and configures the static website.

Figure 6: An `index.html` file in a newly created S3 bucket is used for a test web server.

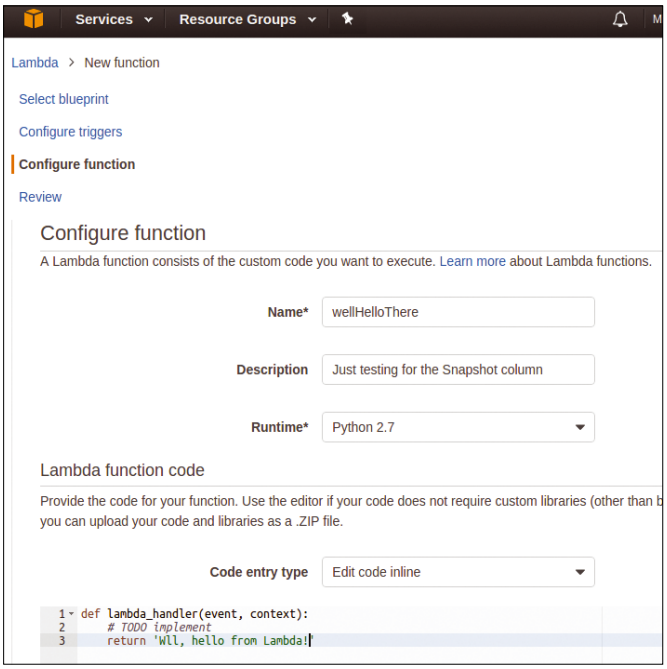


Figure 7: Preparation of the test script in Python as a Lambda function.

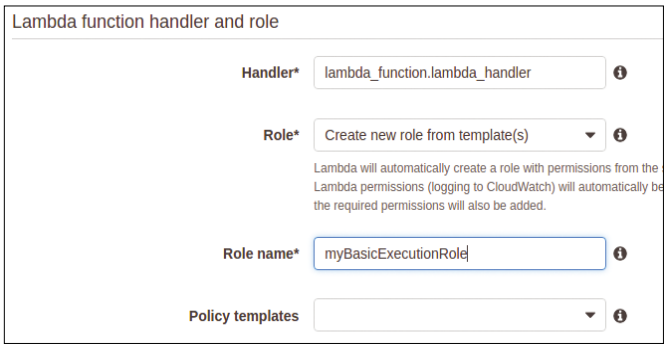


Figure 8: AWS assigns access rights as a role.

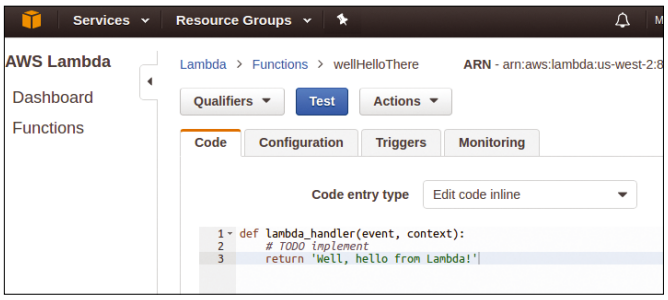


Figure 9: The test console launching a Lambda function programmed in Python.

tainers, triggered either via a web API or associated with events from other services.

For example, Amazon's dynamic database Dynamo can generate an event if a new data record arrives; this in turn triggers a Lambda function that performs further steps in the workflow. A cloud application defined this way thus does not consist of a flow that is orchestrated by the program logic but is formed by linking individual components and their events to create an overall architecture.

In the world of Lambda, the Python script in Listing 1 [5] provides a test function. On the web console, you need to press the *Lambda* option in the *Computing* section to do this. After showing an overview, the service prompts you to *Select blueprint* for the test function. For your tests, select the *Blank function* and skip the next page, *Configure triggers*; then, enter a name for the Lambda function on the following page (as shown in Figure 7; *wellHelloThere* in this case) and copy the code from Listing 1 to the text box shown below the *Edit code inline* drop-down.

On the following page (Figure 8), the console has already entered the name of the handler function. Because a run-time environment might include multiple files with many functions, you need to specify both the filename and the function it contains here.

Leave the default *Role* for execution rights – that is, *Create new role from template(s)* – and specify a suitable name later (*myBasicExecutionRole* in this case). After confirming, Amazon installs the Lambda function in the cloud and lets the user test it (Figure 9). You can add some parameters in JSON format, which the script dynamically evaluates later.

The `aws` command-line client also has access to the Lambda script. As the call in Figure 10 shows, the tool takes the name of the function previously defined in the Web UI (`wellHelloThere`; i.e., not the name of the Python function) and a JSON hash with input parameters as `--payload`; this remains empty in the test case. Later, any parameters provided by the web server end up in the `event` parameter of the Python function that then interprets them dynamically.

Now you need to teach the Lambda script to accept the URL with a video as input, fetch it off the web, and run the motion analysis program [2] on it. This requires more than a simple Python script without dependencies; in fact, you need a run-time environment with the OpenCV library and a precompiled static binary installed. To discover how this works and how to package the results, feed them to the cloud, and define a web service that both triggers the procedure and returns the results as an image file, tune in to next month's column. ■■■

LISTING 1: greet.py

```
def lambda_handler(event, context):
    return "Well, hello from Lambda!"
```

```
$ aws lambda invoke --function-name wellHelloThere --payload "{}" output.txt
{
    "StatusCode": 200
}
$ cat output.txt
"Well, hello from Lambda!
```

Figure 10: The command line can call the Lambda function.

INFO

[1] AWS usage rates for free operation: *https://aws.amazon.com/free*

[2] "Video Preview" by Mike Schilli, *Linux Pro Magazine*, issue 195, February 2017, p. 52, *http://www.linuxpromagazine.com/Issues/2017/195/Perl-Video-Preview*

[3] Poccia, Danilo. *AWS Lambda in Action*, Manning, 2017

[4] Rady, Ben. *Serverless Single Page Apps: Fast, Scalable and Available*, The Pragmatic Bookshelf, 2016

[5] Listings: *ftp://ftp.linux-magazine.com/pub/listings/magazine/195*

Fixing broken packages in Debian systems

Package Repair

When human error stumps the Debian package manager, familiar tools like apt-get, aptitude, and dpkg can help restore functionality. *By Bruce Byfield*

The Debian package manager pioneered automatic dependency resolution during software installation. However, like any software, it cannot protect against human error. Maybe you installed the wrong package from Testing or Unstable repositories or gambled on Experimental. Maybe you installed a flawed third-party package or mixed packages from different Debian derivatives. Or maybe the maintainer made a mistake or a major technology change has happened, and you are not to blame at all. But in all of these cases, you either receive an error message (Figure 1) or a ranked list of possible solutions (Figure 2), and suddenly you are unable to install, remove, or update anything until the problem completes its efforts and returns you to a waiting command prompt.

If you are patient, a new version of the problem package will be released that fixes the problem. The only trouble is, the new version might not be released for weeks, depending on where Debian, or your Debian derivative, like Linux Mint or Ubuntu, happens to be in its development cycle. Even after filing a bug, it can sometimes take time to resolve the problem. Probably, then, you want to take more active steps.

Fortunately, the tools you need are ones you are likely already be familiar with: `apt-get` [1], the package manager's front end; `aptitude` [2], the popular command-line interface; and `dpkg` [3], the basic package tool. All three have the structure

```
COMMAND SUBCOMMAND PACKAGES
```

as well as many of the same features for installing and removing packages.

Apt-get and `dpkg` are installed by default on any Debian or Debian derivative system. However, if you have risky habits, like constantly taking the latest package versions from Unstable, you should make

```
subprocess pre-removal script returned error exit
status 1
/var/lib/dpkg/info/phpmyadmin.postinst: line 35:
/usr/share/dbconfig-common/dpkg/postinst.mysql: No
such file or directory
dpkg: error while cleaning up:
 subprocess post-installation script returned error exit
status 1
Errors were encountered while processing:
phpmyadmin
```

Figure 1: A dependency problem with an error message, but no suggested solution.

```
The following packages have unmet dependencies:
  phpmyadmin: Depends: php5-mcrypt but it is not
installable
            Depends: dbconfig-common but it is not
installable
            Depends: libjs-mootools (>=
1.2.4.0~debian1-1) which is a virtual  package.
The following actions will resolve these
dependencies:

Remove the following packages:
phpmyadmin

Score is 121

Accept this solution? [Y/n/q/?] n
```

Figure 2: A suggested solution for a dependency problem. Notice the score assessing the problem.

stylephotographs, 123RF.com

sure that `aptitude` is installed, as well as other useful tools such as `script`, which can log your recovery efforts, or `equivs`, which builds packages that contain only dependency information. You should also bookmark the Debian bug tracker [4], so you can begin your troubleshooting by seeing whether others have faced the problem on which you are working.

In all these tools, you should gather whatever information you can about the state of the packages involved. However, actually fixing the problem is likely to take you far beyond the usual internal commands like `install` and `remove`.

Making Repairs with apt-get

When `apt-get` announces that you have broken dependencies and suggests solutions, very occasionally, removing problem packages with

```
apt-get remove PACKAGES
```

can solve the problem. On the principle of starting with the simplest solution, try this command, but don't be surprised if it does not succeed.

Another relative long shot is editing package sources to get newer versions of the problem package(s), using `apt-get update` to make them available. In particular, search for a mirror site with more recent packages than your usual ones to add to the file `/etc/apt/sources.list`.

A more promising approach is running

```
apt-get dist-upgrade --no-upgrade
```

to upgrade all the packages installed on the system. Do not use `apt-get upgrade`, since the last thing you want to do is complicate the problem by adding more packages to the mix.

Another possibility is to force completion of an install with:

```
apt-get dist-upgrade -f
```

Sometimes, specifying some or all of the packages mentioned in `apt-get` messages will work instead:

```
apt-get install -f PACKAGES
```

Alternatively, try

```
apt-get remove -f packages
```

but read the summary of what will happen carefully before continuing the command. For some obscure reason, all these commands may work the second, third, or even the fourth time you run them, so run them several times before giving up on them. You can also try specifying the repository and full package name by adding the `-t` option to any of these commands.

However, if you try all these solutions and have no luck, you have exhausted the capabilities of `apt-get` and need to try another command.

Aptitude Dancing

When run without options, `aptitude` opens an ncurses interface to the Debian package manager (Figure 3). However, what many users do not know is that `aptitude` contains many of the same tools as `apt-get` and `dpkg` for fixing broken packages, as well as several extra of its own.

For example, you may be able to resolve problems by using the `markauto` command to mark packages as being automatically installed, or `unmarkauto` to mark them as manual installations. Another useful command is `-t RELEASE`, which specifies which release version to use, or its counterpart `forbid-version` to specify a version not to use.

Another useful pair of tools is `why` and `why-not`. Both are followed by a dependency. The `why` command shows why a dependency would be required, whereas `why-not` shows why a dependency produces a conflict. The results of both can indicate how a subset of broken pack-

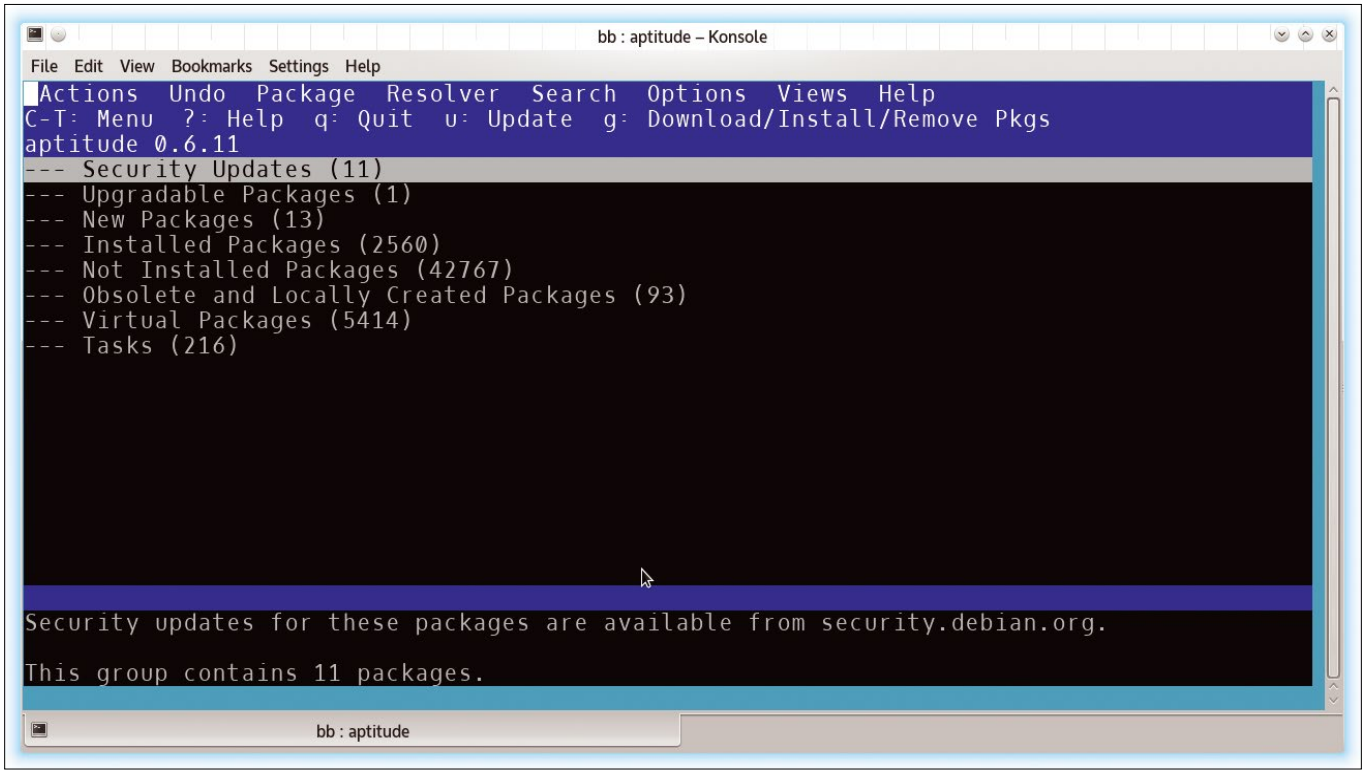


Figure 3: Aptitude is a popular ncurses interface for package management.

ages involving conflicts with another package can be resolved.

However, the most popular feature of `aptitude` is the *Resolver* menu. The menu lists the package manager's suggested solutions to dependency problems and allows you to approve and reject them. Often, this menu alone solves problems that `apt-get`, `dpkg`, or other features of `aptitude` cannot, although at the cost of hiding exactly what it is doing.

Escalating to dpkg

Because `dpkg` is a lower-level package than `apt-get`, it includes many features that `apt-get` and `aptitude` do not. As its man page shows, `dpkg` is especially useful for reading detailed information about packages, including the state it is in, and for filtering the information displayed. You might be able to run `dpkg` with the option `--forget-old-unavailable`, or `--clear-selections` to remove problems, or `--audit (-C)` to receive advice on what actions to try. However, more often, `dpkg` options or commands, such as `dpkg-query`, will be most useful in filtering or gathering background information that can help you develop a solution. The `--yet-to-unpack` option can be especially useful when you have been looking for solutions for some time and don't care to scroll back in your history for the names of the problem packages.

An especially powerful `dpkg` option is `--purge (-P)`. `--purge` is a more powerful version of `remove`, deleting not only the package, but all records of it, including the configuration files. In addition to removing the package, `--purge` also runs its `postrm` (post-removal script). While you are troubleshooting, this thorough deletion can simplify the problem's background and sometimes even solve the problem itself. The `dpkg` man page will give you more information.

Another important option is:

```
dpkg install --ignore-depends=PACKAGE
```

This option is misnamed, since it does check for dependencies but only reports conflicts between packages. Often, it can be the solution for which you are looking.

An equally powerful solution is:

```
dpkg --configure -a
```

which configures all partially installed packages. In my experience, this command fixes more broken dependencies than any other option mentioned in this article, although it is not infallible.

If not, then take a detailed look at `--force-things THING`, as well as `--no-force-things` and `--refuse-things`. Just as `--purge` is an enhanced version of `remove`, so `--force-things` is a fine-tuned version of the `apt-get --force` option. Probably, you probably want to avoid completions of these commands such as `bad-version`, `remove`, or `overwrite` unless you are absolutely confident of what you are doing. However other completions, such as `downgrade`, `configure-any`, and `remove-reinstreq` may provide solutions. But `--force-things` can bring your system down when used carelessly, so consult `--force-help`, as well as the man pages, before using it.

In fact, `dpkg` as a whole can be so deadly that you should use `--no-act [--dry-run, --simulate]` to do a dry run of any action, simply on the off-chance of unexpected effects. The simulation will not tell you in so many words that your system or desktop environment will crash, but studying the list of affected files should warn you that you risk making your situation worse.

Stepping Outside

The Debian package manager has other front ends, notably Synaptic [5], a desktop interface. However, if `apt-get`, `dpkg`, or `aptitude` cannot restore full functionality, then the chances are high that neither Synaptic nor anything else can do so.

That is not to say that finding a solution is easy. Resolving broken dependencies can take hours, and the complications are so numerous that, when you do find a solution, it can feel like luck. The real solution, though, is to work systematically through the possibilities.

All the same, if you regularly find yourself in dependency hell – as broken dependencies were once called – then maybe you should consider your computing habits.

While everyone is tempted by the latest possible release and can make mistakes out of enthusiasm, by stepping outside the safety of the package management system, you are striking out on your own. An expert can do that, but to do so requires caution every step of the way. Otherwise, you may be reduced to desperate efforts such as editing a package's scripts or fiddling with `/etc/apt/preferences` in the faint hope of changing results that have already failed.

Some users thrive on such challenges. Many even find solutions that fall short of reinstalling the entire system. All the same, you have only yourself to blame if you find yourself wasting your time trying to re-enable the package manager instead of being productive or enjoying yourself. ■■■

INFO

[1] apt-get: *https://wiki.debian.org/apt-get*
[2] aptitude: *https://wiki.debian.org/Aptitude*
[3] dpkg: *https://wiki.debian.org/dpkg*
[4] Debian bug tracker: *https://www.debian.org/Bugs/*
[5] Synaptic: *https://wiki.debian.org/Synaptic*

IT Highlights at a Glance

Too busy to wade through press releases and chatty tech news sites? Let us deliver the most relevant news, technical articles, and tool tips – straight to your in box. Subscribe today for our excellent newsletters:

- ADMIN HPC
- Linux Update
- ADMIN Update
- Raspberry Pi

and keep your finger on the pulse of the IT industry.

Admin and HPC: www.admin-magazine.com/newsletter
Linux Update: www.linuxpromagazine.com/mc/subscribe
Raspberry Pi: www.raspberry-pi-geek.com/mc/subscribe

Static galleries with Expose

Exposure Time

Expose offers a wide range of configurable options for publishing static photo and video galleries in an easy-to-use tool. *By Dmitri Popov*

There are plenty of reasons to use a static website generator for web publishing instead of a traditional content management system. Serving static pages requires only a web server, which dramatically simplifies the required setup. This, in turn, improves security and reduces maintenance overhead, as the minimal stack has fewer potential vulnerabilities and is easier to troubleshoot and keep up to date.

Moreover, since publishing static content can be done using a lightweight web server, you can host your site on modest hardware or an inexpensive virtual private server.

That's all fine and dandy, but most static generators are designed to work with text-centric content like blog posts and long-form articles. But, what if you want to publish a photo essay or a photo gallery? Enter Expose [1], a Bash shell script for generating static photo and video galleries (see Figure 1). The script is less than a thousand lines long, but it's capable of generating rather impressive galleries and photo essays, and offers a wide range of configurable options to boot.

Getting Started with Expose

Expose has only two dependencies: ImageMagick and FFmpeg. The latter is required only if you plan to publish videos. Better still, being just a regular Bash

Lead Image © ssilver, 123RF.com

Figure 1: Expose generates polished and user-friendly static photo galleries.

shell script, Expose requires no installation. Start with installing the required packages. To do this on Debian or Ubuntu, run the command:

```
sudo apt install imagemagick ffmpeg
```

To install the packages on openSUSE run the command:

```
sudo zypper in ImageMagick ffmeg
```

To deploy the script on your machine, clone the project's Git repository using the following command (make sure that Git is installed on your system first):

```
git clone https://github.com/Jack000/ ➥
          Expose.git
```

Open then the `~/bashrc` file in a text editor and specify an alias that points to the `expose.sh` script:

```
alias expose=/script/location/expose.sh
```

Before you start using the script, you might want to edit some basic settings. You can do this either by modifying the defaults directly in the `expose.sh` script or creating a separate `_config.sh` file in the directory containing the photos you want to publish.

In the latter case, configuration may look something like this:

```
site_title="Title Goes Here"
theme_dir="theme1"
social_button=false
backgroundcolor="#ffffff"
```

Expose comes with two themes: the default theme presents photos as a gallery, while the second theme is more suitable for photo essays featuring a mixture of text and images.

Choosing the theme you want is a matter of specifying its name using the `theme_dir` configuration option. The `social_button` option lets you enable or disable sharing buttons, while the `backgroundcolor` option specifies the background color of the gallery.

Expose supports plenty of other options, too (Figure 2). If you want to allow visitors to download the published photos, use the `download_button=true` option. In this case, you also need to install the zip package using the `sudo apt install zip` command on Debian and Ubuntu or `sudo zypper in zip` on openSUSE. With the download option enabled, Expose conveniently bundles a readme file with the default copyright notice, but you can change it using the `download_readme` option, for example:

```
download_readme="CC BY-SA-NC 4.0"
```

By default, Expose reduces image quality to 92%, but you can override this using the `jpeg_quality` option as follows: `jpeg_quality=99`. Want to give your visitor the option to add comments? You can enable support for the Disqus commenting service by specifying your Disqus shortname: `disqus_shortname="shortname"`. Expose also provides a range of video-related options, and you can find them along with their brief descriptions in the `expose.sh` script.

Before you run the script, you need to do some preparatory work. Expose sorts images in alphabetical order, so to arrange photos, you might want to rename them. One way to do this is to use numeric prefixes (e.g., `001_foo.jpeg`, `002_bar.jpeg`, etc.). You can group photos in folders, and Expose generates a navigation menu based on the folder structure. To organize the folders, add numeric prefixes to them. The script strips these prefixes when generating a gallery, so they don't appear in the navigation menu. If you want to skip a certain folder, prefix it with _ (e.g., `_private-photos`), and Expose will ignore the folder when generating a gallery.

Using Expose couldn't be easier. In the terminal, switch to the directory containing the photos and videos you want to publish, and run the `expose` command. This generates a complete static gallery in the `_site` directory. If you want to generate a preview, run the `expose -d` command to create a gallery with low-resolution images. Keep in mind, though, that Expose doesn't overwrite existing images when you run the script again, so you need to delete the `_site` directory to regenerate the gallery with high-resolution photos.

Adding Descriptions and Settings

Adding a text description to a photo is as easy as creating a `.txt` file with the same file name as the photo. For example, to add a description to the `001_foo.jpeg` photo, create the `001_foo.txt` file with the desired text in it.

In the default theme, the specified description appears as a text overlay in the lower part of the photo. But this might not always be the most optimal text placement. Fortunately, Expose makes it possible to define text position and flow by specifying YAML configuration settings in the description file. You can specify the exact text position as well as the width and height of the text box by adding the following configuration at the beginning of the description file (all values are expressed in percent):

```
---
top: 15
```

Figure 2: You can use the available options to toggle sharing buttons, enable downloads, and configure other settings.

```
left: 5
width: 25
height: 20
---
```

In this example, the `top: 15` and `left: 5` options instruct the script to place the text 15% from the top edge and 5% from the left edge of the photo. In addition to position, you can also specify text color using the `textcolor` settings:

```
---
top: 15
left: 5
width: 25
height: 20
textcolor: #ffffff
---
```

By default, the text is placed in a rectangular text box of specified width and height, but it's also possible to make the text flow around shapes using the `polygon` option. Using this option, you can specify multiple points using the X and Y values, thus turning the rectangular box into a polygon.

Here is how this works in practice. Use the `width` option to specify the width of the text box, for example:

```
---
width: 50
---
```

The X and Y coordinates of each corner of the box are as follows:

```
x:0,y:0-------x100,y:0
   |             |
   |             |
x:100,y:100-----x:0,y:100
```

The rectangle can be defined using the `polygon` option with a JSON-formatted list of coordinates:

```
polygon:[{"x":0, "y":0}, ➥
         {"x":100, "y":0}, ➥
         {"x":100, "y":100}, ➥
         {"x":0, "y":100}]
```

Knowing this, you can adjust the coordinates of each corner as well as introduce additional points with specific coordinates. For example, to add a left-to-right slope to the right side of the text box, adjust the X value of the top-right corner as follows:

```
polygon:[{"x":0, "y":0}, ➥
         {"x":25, "y":0}, ➥
         {"x":100, "y":100}, ➥
         {"x":0, "y":100}]
```

Expose uses ImageMagick for all image manipulation tasks, and you can add processing instructions to the description file for on-the-fly image adjustments. For example, if you want to watermark a specific image add the following processing instruction to its description file:

```
---
image-options: /path/to/watermark.png ➥
  -gravity SouthWest ➥
  -geometry +10+10 ➥
  -composite
---
```

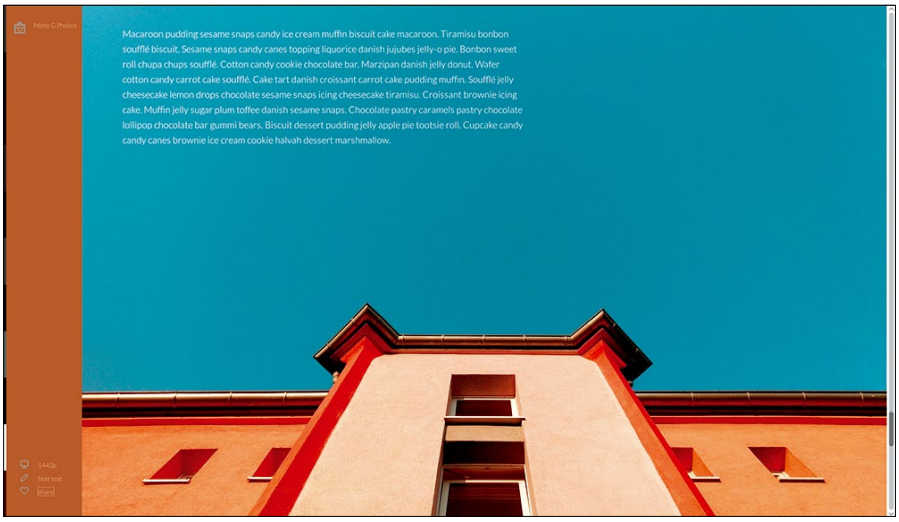


Figure 3: Expose makes it possible to add a description to each photo.

This uses the `watermark.png` image from the specified location to apply a watermark at the lower-right corner of the image, with a 10-pixel margin from the edge. Need to sharpen a photo? The following option does the trick:

```
---
image-options: -sharpen 0x1.5
---
```

You can use practically any option supported by ImageMagick, so you might want to peruse the Command-Line Options page [2] if you want to make the most out of this functionality.

Instead of specifying options for individual photos, you can apply them globally to all published images. To do this, create the `metadata.txt` file and specify the desired options in it.

If you shoot time-lapse images, you'll be pleased to learn that Expose can automatically encode them as videos, no manual work required. Add the `imagesequence` keyword to the folder containing time-lapse images (e.g., `001 tokyo nightscape imagesequence`) and put in the directory with the photos you want to publish. Then run the `expose` command, and the script transforms the image sequence into a video.

In Conclusion

Despite its simplicity, Expose is a rather capable and flexible tool for generating static photo and video galleries. It gives you full control of the publishing process and allows you to configure a wide range of options.

More importantly, the Expose tool produces galleries and essays that look exceptionally good and are easy to navigate. ■■■

INFO

[1] Expose: *github.com/Jack000/Expose*

[2] ImageMagick options: *www.imagemagick.org/script/command-line-options.php*

Ben Everard

What does the word Linux mean to you? To some people, Linux is an open source kernel initially developed by Linus Torvalds; to others, it is a desktop and server operating systems built on this kernel; to others, the term "Linux" refers to anything that has the Linux kernel at the base – from Android and embedded systems to Ubuntu and Fedora on the desktop. It's hard to say which definition is categorically correct, and the question has even become a political issue for many. In the interest of harmony, Linux Voice has articles that are about whatever definition of Linux you prefer. For the only-a-kernel stalwarts, Valentine Sinitsyn looks at how to customize the kernel for your personal use, which helps your squeeze every drop of performance out of your machine. For those who follow the view that Linux is a desktop and server OS, you can find Graham's selection of the latest software for these systems in FOSSPicks. For the extremists like myself who willfully include everything that's ever touched the kernel under the Linux moniker, I've taken a look at the latest introduction to the wider-Linux ecosystem, Lineage OS, in this month's FAQ.

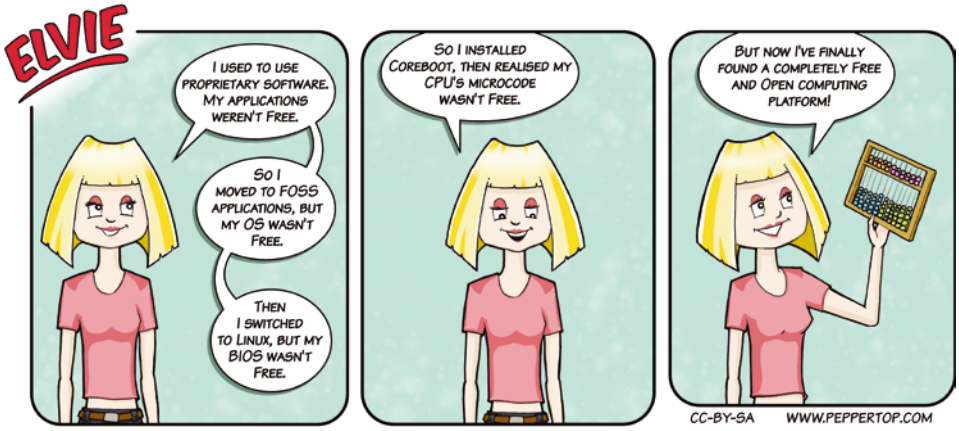
Let's not let our etymological differences drive us apart. The start of a new year is about hope and looking forward to a better tomorrow, not about bickering over small details. The future, as one famous song puts it, is not ours to see, but the evidence is that it's likely to be more penguin-friendly than the present. If you just want some positive news to help you through the darkest days of winter (or the hottest days of summer for our antipodean readers), flick straight to Mike Saunders explaining why 2017 will really be the best year for Linux so far. Let's band together and look forward to some great developments in Linux, whatever that word means to you.

– Ben Everard

Andrew Gregory

Graham Morrison

Mike Saunders



LINUXVOICE

Why 2017 Will Be Awesome

2016 was a wild ride – and 2017 promises to deliver even more FOSS goodness.

BY MIKE SAUNDERS

It's something of an in-joke to say "`$CURRENT_YEAR+1` will be the year of Linux on the desktop." This started back in the early 2000s as a more serious statement, reflecting the optimism at the time. Linux was on the brink of major success, after all; just a few things needed to go right, and Microsoft's days of dominance on the desktop would be over. Many of us had been using Linux as our daily desktop OS for years, but we were still waiting for that final big breakthrough.

So why didn't it happen? Some would argue that it took a long time for desktop distros to become really polished and user friendly, while others would point to the relative paucity of commercial triple-A applications. I think, however, that the biggest factor involved was simply users' reluctance to change. Consider how many Windows users clung on to XP while Vista, 7, and 8 came out and how much the Microsoft Office Ribbon interface was (and still is) hated by a large number of people.

No matter how much Linux improved, most people stick with what they know. Now, in the meantime, Linux has come to dominate smartphones, tablets, embedded devices, and the cloud, building upon its success in the server and networking spaces. So as 2017 gets underway, can it finally crack that especially difficult desktop nut? Surveys point to Linux attaining a 2%-3% desktop market share, which may seem tiny but is impressive growth from the 1.5-ish% we were used to. And there are many things coming up in the next 12 months that could really help. So let's see what's in the pipeline ….

Wayland

It feels like we've been talking about Wayland forever, and, indeed, it has been in development for almost a decade now. In case you're not familiar with it, Wayland is a display server protocol and implementation that serves as a replacement for the old X Window System. If this all sounds like gobbledygook to you, consider this: the X Window System is the base graphical layer on your Linux desktop, which talks to the graphics card, draws dots on the screen, handles mouse input, and does other jobs. On top of that, you have graphical widget toolkits such as Qt and Gtk, which provide buttons, menus, and other facilities, and then apps and desktops such as KDE and Firefox use those toolkits to create shiny apps.

Now, the X Window System (aka X) has been around since the 1980s and been the de facto standard graphical layer on desktop Linux distros since forever, so why is it being replaced now? It's true that X has done a decent job over the years and offers some nifty features, such as network transparency, where you can easily run an app on one machine and display it on another. (Yes, you can do this with VNC and the like, but those just copy pictures over the network, whereas X actually sends specific drawing commands in a more elegant fashion.)

X had its limitations, though. There was a lot of cruft in the codebase, and many of us were still dealing with performance issues and "tearing" when dragging windows around on the screen. There were security concerns as well, where one program could capture keyboard input from another. Wayland was conceived to replace this old codebase with something newer, smaller, and smoother, and has been "just around the corner" for a while.

Until now, though, only one distro actually used it by default – and that was RebeccaBlackOS, a novelty distro that nobody took seriously but actually did a good job of showcasing Wayland (see Figure -1). It did highlight, however, that Wayland – or more specifically, programs running on it – still had issues. There are many graphical toolkits (and different versions of those toolkits) in use, along with graphical apps, that use the base X libraries, so getting everything running smoothly on Wayland has been a gargantuan task.

2017 might be the year of Wayland. Fedora 25, released in November last year, was the first major distro to use Wayland by default (with an option to switch back to X if users experience problems). Fedora has long been a cutting-edge showcase of new technologies, but it's still used for serious work, so this was a major step forward. I expect some of the other mainstream

desktop distros to follow suit in 2017, with the exception of Ubuntu, of course, which is due to adopt the Mir alternative display server at some point.

LibreOffice 5.3

LibreOffice 5.3 is due for release in late January or early February, so it might already be available by the time you read this. As usual with any LibreOffice release, much work has gone into improving file filters – especially for compatibility with Microsoft Office – along with bug fixes and performance improvements. But there's plenty to talk about in terms of new features, as well. Writer, the word processor, now includes proper table style functionality, letting you apply and switch between styles just as with any other part of a document.

In previous LibreOffice releases, you could only apply some "auto-formatting" to a table in a document (i.e., adding colors, bold, italics; changing font sizes; etc.). But this was a one-off job, so when you modified the table, you'd have to apply the formatting all over again. It was tiresome and prone to problems. With table styles, you can apply a specific style to a table, modify that table, and the style effects will remain in place.

In the spreadsheet Calc, wildcards are enabled in formulas for fresh installations (i.e., new installations that don't inherit settings from a previously installed release). This improves compatibility with other office suites and makes it easier for new users to work with formulas, in that they don't need to mess around with regular expressions any more. Calc also includes a new set of default cell styles that are much prettier and have more descriptive names than in earlier versions.

Additionally, a "safe mode" has been introduced that aims to identify and fix problems with user profiles. If LibreOffice is behaving oddly, users can start the suite in safe mode and either create a fresh profile from scratch or disable certain settings and extensions to narrow down the cause of the problem. This has been requested by users for many years and should help cure many headaches where users can't use the suite properly because of a corrupt or misbehaving user profile.

The biggest change in LibreOffice 5.3 is the new NotebookBar (see Figure 2). This is part of a larger user interface concept called MUFFIN, the "My User Friendly and Flexible INterface." While MUFFIN is a jokey name, much like the code names for Ubuntu and Fedora releases (Beefy Miracle, anyone?), it's used to describe a design approach in which users choose what works best for them.

Many LibreOffice end users have requested a Ribbon-like interface for years, and indeed other combinations involving toolbars, menus, and sidebars. With MUFFIN, users can switch between four user interface setups: the standard dual-line

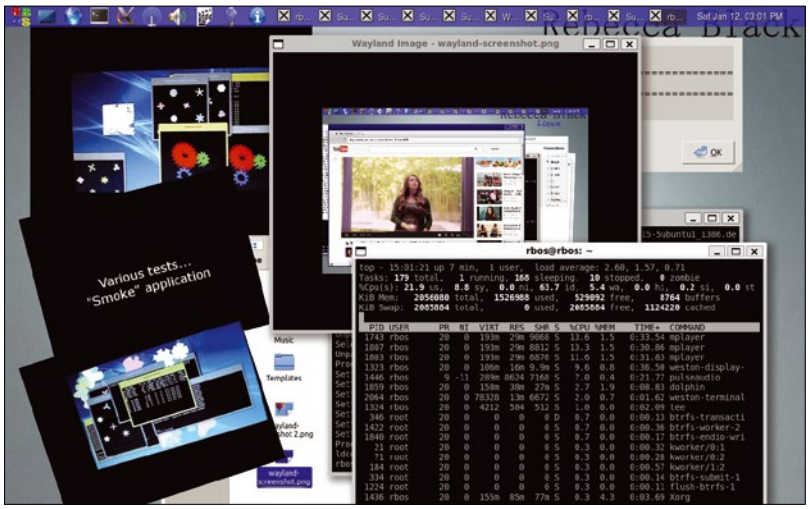


Figure 1: RebeccaBlackOS was the first distro to use Wayland by default, but now more mainstream distros should adopt Wayland.

toolbar, a single toolbar mode, toolbar(s) with the sidebar, and the new NotebookBar. It's important to note (no pun intended) that the NotebookBar isn't a clone of the Ribbon, but rather a new design that aims to organize buttons and operations by categories.

Currently, the NotebookBar is still very much in development and is marked as an experimental feature in LibreOffice 5.3. So, in other words, it's not being pitched to end users as a new default interface, but something to try and provide feedback on as the designers and developers iron out bugs. Whether it will become the default GUI layout in LibreOffice 5.4 or 6.0 remains to be seen, but in any case, it's good to see progress on this front.

Some pundits have been complaining that LibreOffice is going down this multi-interface route, saying that the dev team should focus on a single, all-encompassing GUI instead. There's some sense to that argument, but when you have some users clamoring for a shiny new interface and oth-

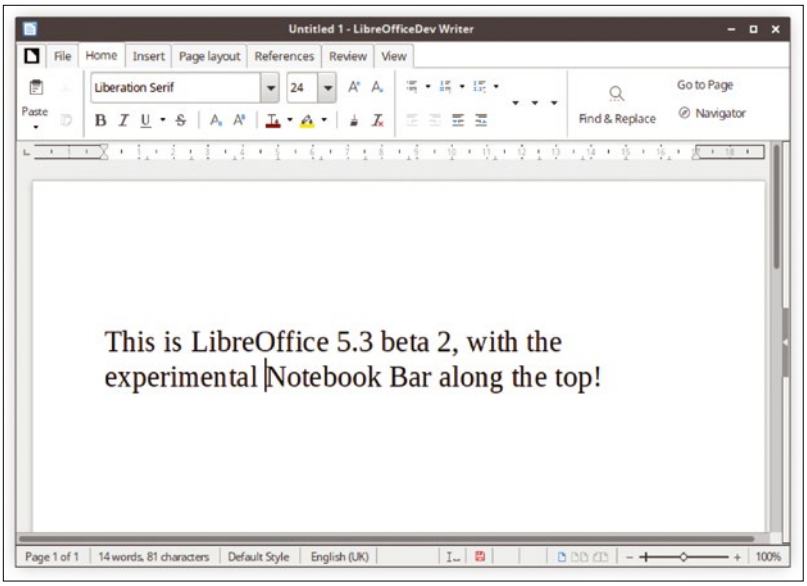


Figure 2: LibreOffice 5.3 will introduce an (experimental!) NotebookBar interface as an alternative to traditional toolbars.

ers that would rather die than give up their toolbars, this might be the best compromise to give the suite (and community) a healthy future.

Gnome, KDE, and Xfce

All being well, Gnome 3.24 will be released on March 22nd. It looks to be a largely evolutionary release, rather than revolutionary, which is expected at this stage of the desktop's maturity. A lot of work has been put into Epiphany, the web browser, which will have experimental support for HTTPS Everywhere [1] for improved security, along with syncing of bookmarks via Firefox Sync. Moreover, the bookmark management dialog has been redesigned to be easier to use, while the whole codebase has been relicensed from GPLv2 to GPLv3.

Gnome 3.24 will also include a new version of NetworkManager with improved Bash auto-completion and various fixes, while Gnome Music uses Cairo for cover scaling. Even the todo applet will see some new features, such as subtasks, while Gnome logs is faster when performing searches. If you spend a lot of time using mobile data, you'll be happy to see a new option in Gnome Software: "Only download updates on non-metered connections." In other words, Gnome won't try to pull hundreds of megabytes of data from the net when you're connected via your mobile phone or similar device.

Meanwhile, KDE Plasma 5.9 is due to arrive at the end of January, so it may well be available for your distro by the time you read this (especially if you run a rolling release distro). One of the most requested new features that will hopefully make it into 5.9 but may be pushed into the following release is a global menubar – as in Mac OS. The Kirigami UI has been bumped up to version 2 and now includes keyboard navigation for those who hate having to reach for the mouse, and Snap package support may get squeezed into 5.9, as well.

Over in Xfce land, no release date has been set for version 4.14 of this desktop yet, but it may arrive this year. By far the biggest change will be the porting of all core components from Gtk2 to Gtk3 – which may seem like a long time coming but will be a welcome update. Additionally, *dbus-glib* will be replaced by `gdbus`. The Mate desktop (a continuation of the Gnome 2 codebase) should also complete the transition to Gtk3, and version 1.18 should be released early in the year.

Gimp and Firefox

Gimp's current stable release series is 2.8.x, and can you guess when 2.8.0 was released? Way back in 2012. Yes, we've been waiting for almost five years for a new major update, and for many artists and designers, the delay has been agonizing. We can't point fingers, of course, because open source projects can only progress when there are enough developers to contribute, but we really hope to see Gimp 2.10 some time this year.

The changelog [2] is huge: Gimp 2.10 will use the GEGL image processing library for all operations, with higher bit depths and experimental hardware accelerated rendering via OpenCL, and

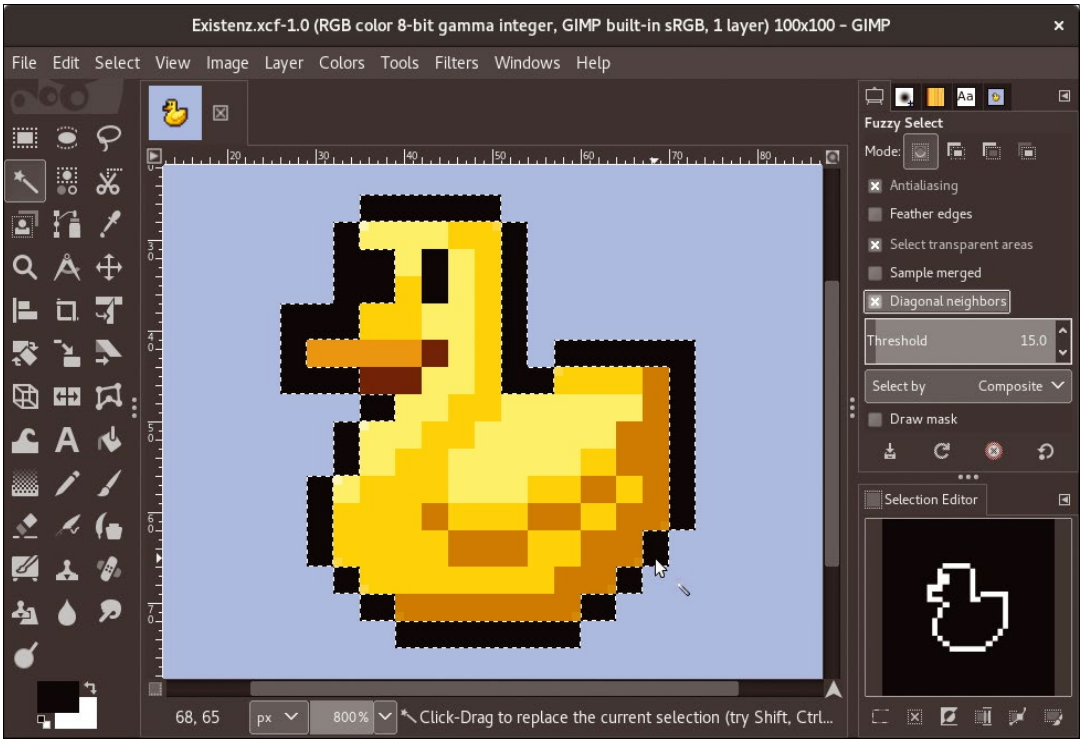


Figure 3: When Gimp 2.10 finally arrives, it will sport a bunch of new features, including diagonal pixel detection in fuzzy selects.

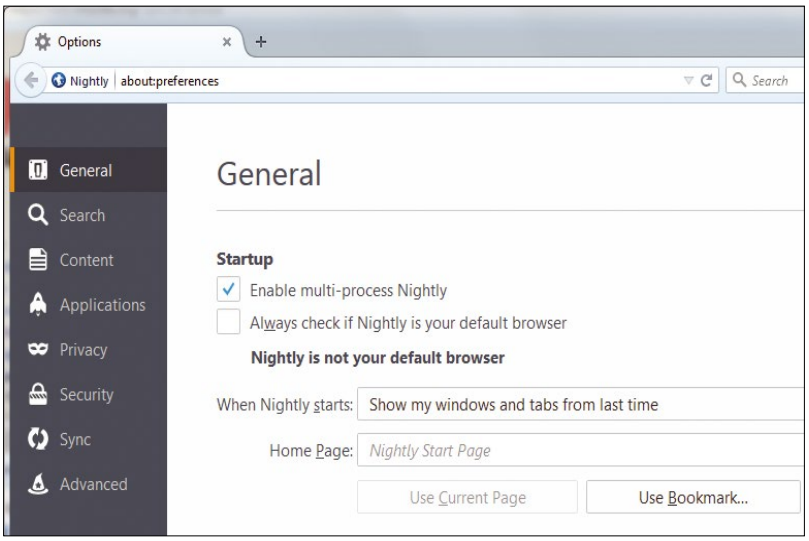


Figure 4: Users of Firefox nightly builds already can try the Electrolysis process separation system for better performance.

the position and content of items on the canvas can be locked to prevent them being edited (as often seen in desktop publishing applications).

Many new tools have been added (see Figure 3), including *Unified Transform* (which combines rotation, scaling, and skewing in a single tool), *N-point deformation* (to bend objects in a natural way), and *Warp Transform* (like the old *IWarp* plugin, but working directly on the image and not just in a preview window). Additionally, all filters can be previewed on the image canvas itself, and new icon sets have been added to make the user interface shinier.

Firefox, meanwhile, has a much more aggressive release schedule than Gimp, so we'll definitely see some major updates this year. The one we're all waiting for is "Electrolysis," the work to split Firefox up into multiple processes rather than running everything together as one big lump. This should drastically improve performance, because individual tabs, add-ons, and the user interface can all run in separate processes, so if one becomes particularly sluggish for whatever reason, the others won't be affected.

Some beta testing for Electrolysis is already underway (see Figure 4); all being well, it will become the default in Firefox 51 or 52 toward the middle of the year (or a bit later). Another possible new feature is Context Graph, a homepage replacement that recommends websites to you based on the context of what you've recently been looking at. This might be a controversial feature among FOSS purists, but if we can turn it off, we'll live with it. For more on Context Graph, see the wiki [3]. ■■■

Info

[1] HTTPS Everywhere: *https://www.eff.org/https-everywhere*
[2] Changelog: *https://wiki.gimp.org/wiki/Release:2.10_changelog*
[3] Context Graph wiki: *https://wiki.mozilla.org/Context_Graph*
[4] SQL Server on Linux: *https://www.microsoft.com/en-us/sql-server/sql-server-on-linux*

Android and Chrome OS To Become One?

Another thing we may see in 2017 is the fruit of Google's Andromeda, a project to merge the Android and Chrome OS codebases (see Figure 5). Although Google hasn't officially announced anything along these lines yet, rumors have it that the company is working on this internally and wants to unify the mobile operating systems into a single core. This makes sense to us: There's an increasing amount of overlap between the use cases of smartphones, tablets, and small laptops, especially as the former become bigger and more powerful.

However, the merge has to be executed well. Over the past few years, we've seen far too many clumsy attempts to shoehorn touch interfaces into more traditional pointer-driven ones, and vice versa. If Andromeda becomes public, it has to be more than just a mash-up of interfaces – it has to work seamlessly across different devices so that you don't even notice it.

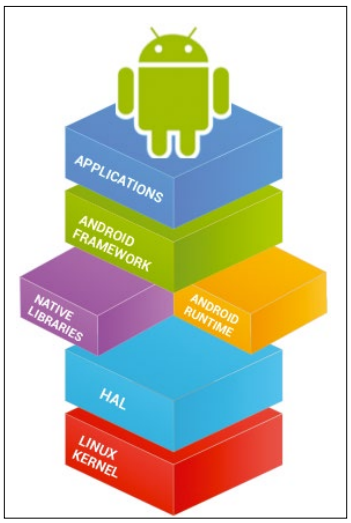


Figure 5: Will this be the year that Android and Chrome OS finally merge into the one mobile OS to rule them all?

Microsoft and Linux in 2017

Now that Microsoft has said it "loves" Linux and is porting various tools to the OS, what can you expect from the company in 2017? Around the middle of the year, you should see the public release of SQL Server for Linux. At the time of writing, a preview version for Red Hat Enterprise Linux, SUSE Enterprise Linux Server, and Ubuntu was available [4]. For developers, a new version of Visual Studio Code for Linux should also arrive early this year.

In more general terms, it will be interesting to see how Microsoft's attitude toward Linux and open source continues to evolve throughout the year, especially as the company jumped on board the Linux Foundation in November 2016 as a high-paying Platinum member. Many of us who've been using Linux since the 1990s remember the bad old "Linux is a cancer" days of ex-CEO Steve Ballmer, and while things have changed considerably under Satya Nadella, it's still best to be cautious. If Microsoft "loves" Linux, that's "great," but ultimately the company's main interest is making money for its shareholders.

Jon "maddog" Hall is an author, educator, computer scientist, and free software pioneer who has been a passionate advocate for Linux since 1994 when he first met Linus Torvalds and facilitated the port of Linux to a 64-bit system. He serves as president of Linux International®.

MADDOG'S DOGHOUSE

Cooperatives have a lot of power and flexibility that other business models lack. FOSS projects have leveraged this structure to create valuable and lasting products.

BY JON "MADDOG" HALL

At a GNU/Linux Meetup recently, I met a young woman who presented me a book that contained a chapter she had authored. The book is titled *Ours To Hack and Own* [1], and it discusses "cooperativism," the economic practice of running a company as a cooperative, wherein either the employees or the customers are the owners.

Many readers might stop at this point and ask, "How does this relate to Free and Open Source Software?" FOSS projects are typically the epitome of cooperativism, so please continue reading.

Many years ago, when I was a university student, there was a small grocery store in the heart of Philadelphia that was a cooperative ("coop" for short). The people who regularly shopped there joined the coop, and the money they paid for groceries went directly to buying the groceries, paying the salaries of the employees, and taking care of other expenses. Some of the people who shopped there also formed a board of directors, who helped govern the coop. These governors made the rules for running the coop, hired the Executive Director (who in turn hired the rest of the staff), and determined the Executive Director's salary. The Executive Director was responsible for running the coop efficiently and making sure the cooperative either put excess revenue back into the coop, to make it better, or lowered the prices to make sure the food was the best for the lowest price. People who actually belonged to the coop (and showed their membership card) got a small percentage off the already low prices and could attend the membership meetings once a year to vote for board members. The coop had no "owners" to siphon off "profits" to make the goods more expensive.

There are many examples of coops. I belong to a credit union, a type of bank that is owned by the depositors. Many rural electric companies are coops, owned by the people that buy the electricity. Early rural telephone companies were often coops, started by the users to provide telephone service when larger telephone companies did not see that segment of the population profitable enough to service. An association of these rural telephone companies helped coordinate their efforts to obtain low-cost loans to help establish their coops. Other examples include failing businesses that were purchased from the owners by the employees, who then turned around the business and made it self-sustaining, even after paying themselves better wages.

The essence of most coops is that the people who either work there or buy the products start the coop for their own needs – and not to generate money for remote stockholders.

So it is with many FOSS projects. Developers who start the project are usually passionate about some topic and start to write the code. In most of their minds, the object is to create "a really good piece of code" that they can use, not to create a product that they can sell to millions of people. Some projects become so large that the developers create a more formal way of driving development by establishing a board of directors who govern the project. The Debian distribution is a good example of this structure.

FOSS has inspired many other ventures, such as Creative Commons and the Open Hardware movement. Many of these projects appeared because people realized that when you "give away" certain things, it does not hurt the venture, and it often helps the project grow, either by attracting more developers or bringing more revenue into the organization from services. For more than 20 years, I have been talking about "making money with free software." In every discussion, I talk about the value of the cooperative model compared with creating a "sole proprietorship" form of business.

Cooperatives take more effort to start, but on the other hand, they allow the sharing of expensive resources over a (potentially) larger customer base. A single person will only be able to satisfy a certain number of customers, but a group of good people working together could satisfy a much larger number of customers.

More importantly, a coop would allow a greater diversity of expertise in the business, as well as a greater pool of talent to service customer needs. Many large companies will not do business with a single-proprietor business, fearing loss of data and service when that person dies, goes on vacation, or is overloaded with other work. A coop, because it is designed as a business from the start, can expand more easily in times of expanding business.

More importantly – particularly in the case of the employees (not the state or third parties) owning the coop – the decisions made by the coop are made locally, and not by a board of directors many miles away. ■■■

INFO

[1] Scholz, Trebor, and Nathan Schneider, editors. *Ours To Hack and Own*. OR Books, 2016

Habeas Video

BY ANDREW GREGORY

An opportunity missed for Free Software evangelists

As I write this there's a protest going on at the shooting of a man by armed police in West Yorkshire. Not many details have been released to the public, but 24-hour news has shown us plenty of images of the dead man's car with three neat bullet holes in the windscreen. We know that the man was killed at the end of what the police are calling a "preplanned operation." We also know that no body-cam footage will ever come to light, because West Yorkshire Police does not require its officers – even firearms officers – to wear body cams.

Now, the numbers of people killed by the police in the UK is extremely small compared with that in the USA, where things like having a broken tail light, driving without insurance, and even running away can get you killed. But each and every fatality at the hands of the police is a personal tragedy for the friends and family of the deceased. We owe it to them to make sure the police are investigated as thoroughly as possible. The police themselves should see it as in their interests to put themselves above suspicion at every turn, and we, the people, really need to have every confidence that the upholders of the law aren't turning into an extrajudicial death squad. Transparency brings confidence in the system, and we all benefit, which is why I find it odd that Microsoft will be hosting body-cam footage on behalf of the Metropolitan Police via its Azure cloud. Or should I say, its Azure "Other People's Computers."

Although Microsoft says it will store the police videos at its UK data center, in the end, Microsoft is not a UK company, but rather a US company with obligations that may not always be perfectly aligned with those of the UK. It's not hard to imagine a scenario in which Microsoft's contract with the UK government runs contrary with the US's demands to know more about, for example, a terror suspect. The Metropolitan Police could avoid these complications by simply storing the videos in house using Free Software. There are plenty of cloud hosting solutions out there that the police could use to keep the data in their own data centers in a way that is much less likely to compromise the rights of victims, the accused, and even the police officers.

Microsoft isn't cheap. It's daft for an employer as large as the Metropolitan Police to pay for a commercial solution when it has the resources to develop a solution itself that can then be rolled out across the rest of the country's police forces.

Finally, and most obviously I suspect, the chain of custody for any evidence that only exists digitally, stored by a private company using proprietary software, is irredeemably compromised. ■■■

Subscribe today and join the revolution!

Each issue of Raspberry Pi Geek is an adventure, with ingenious applications and cool projects for Raspberry Pi, Arduino, and other maker-board systems!

Discover the secrets that will empower you to envision and build your own Raspberry Pi inventions.

Features

Special articles on new technologies and topics of interest to the Rasp Pi community.

6 print issues with 6 DVDs or
6 digital issues for only

$59.95 £37.50 €44.90 shop.linuxnewmedia.com

Projects

Do-it-yourself, real-world projects that let you learn by doing.

DVD

Every print issue comes with a free DVD!

Kid Stop

Special projects for kids, including a new Scratch programming exercise in every issue.

Community

Interviews and reports on Raspberry Pi meet-ups around the world.

Skills

Timely tutorials to help you build your skills with Linux and other underlying technologies.

FAQ Lineage OS

CyanogenMod is dead. Long live Lineage.

BY BEN EVERARD

Q I'm going to take a guess from the name that this is some ancient operating system that's been around since before Tux was an egg resting on his father's feet in the frozen wasteland of Antarctica.

A I'm afraid that you're not even remotely close. Lineage OS is a new distro that's split from CyanogenMod following disagreements about the company's future.

Q Hang on, what's this Cyano-thingy you're talking about?

A CyanogenMod. It's basically a distribution of the Android OS with all the closed-source Google bits removed. It's available for a vast range of phones, tablets, and other devices that manufacturers claim are smart – or at least it was available, but we'll get on to that in a bit. While it was around, it was a great place to get the latest Android version for your phone even if the original device maker had decided not to update the software. It was also a great way to get an Android version minus the Google bits, so it was much closer to being a truly open source device (although not 100%, for that you needed Replicant), and you could be a bit more confident that your phone wasn't riddled with spyware.

Q Ah, CyanogenMod sounds like it was awesome! What happened to it?

A CyanogenMod started out as a community project lead by Steve Kondik (also known as Cyanogen). The project grew from a small one-man operation to a large open source project that took in contributions from lots of people and supported a bewildering array of devices.
In 2013, Kondik launched Cyanogen Inc. to commercialize the project. The core distro remained open source and free to use, but they worked with manufacturers to develop special versions of the OS for some phones, most notably the OnePlus One, but also some others.
We won't go into the various rumors of what happened internally at Cyanogen Inc., but suffice it to say that it didn't become the company that Kondik wanted it to be, and he left the company in November 2016. Around the same time, Cyanogen Inc. announced that it was shutting down the infrastructure around the CyanogenMod open source project. This infrastructure includes the wiki, developer's collaboration tools, and image hosting.

Q What, the company just deleted everything that the community had spent years building?

A Yes, but don't despair. The Internet Archive (see Figure 1) exists for situations exactly like this. All the information for users on the wiki [1] and the images [2] are still available online, so regardless of what happens with Lineage OS, that will all still be available.

Q Right. I think I'm up to speed on CyanogenMod now. Can we get on to Lineage OS?

A Yep! While Cyanogen Inc. can delete the data created by the community, they can't delete its spirit. All the people who made the original CyanogenMod are still around and still interested in making awesome mobile operating systems. Lineage OS is a continuation of CyanogenMod by many of the same developers, including Kondik – Cyanogen himself.

Q Hang on, if Steve Kondik is working on Lineage OS, and Steve is

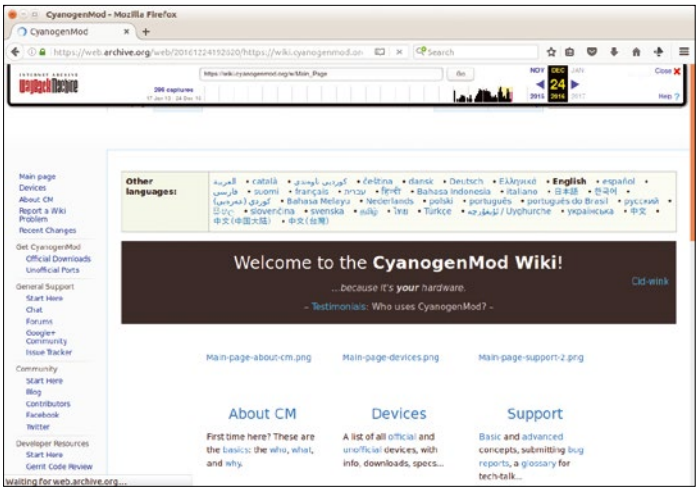


Figure 1: The Internet Archive is one of the great triumphs of the Internet age and exists to ensure important information isn't easily destroyed.

the original Cyanogen, and the company is discontinuing their open source efforts, why not just keep the name CyanogenMod and avoid any confusion?

A A couple of reasons. First, Cyanogen Inc. owns the name CyanogenMod, so it's up to them whether or not to let the new project use the name. Second, there's now quite a bit of bad will around the name, so the new project wants to start afresh. We can't say for sure which of these is the primary reason, but together they mean that the new project has had to come up with a new name.

Actually, here at *Linux Voice*, we were never that fond of the name CyanogenMod. It sounded weird and overly geeky to people outside the tech community. Lineage OS, on the other hand, is at least a word people can understand and spell, and the OS at the end of the name makes it sufficiently unique that web searches bring up the correct result, so we're all for it.

Q That sounds great. Cyanogen Inc. has shut down CyanogenMod, but nothing's gone away except the name.

A Well, not quite. CyanogenMod was the open source community project. In addition to this, Cyanogen Inc. provided some services that aren't necessary for the basics of a mobile operating system but are nice additions, for example, the Cyanogen app store (C-Apps). These have now been shut down. There are alternatives (including the F-Droid open source app store and of course Google's services), but this particular part of Cyanogen is unlikely to come to Lineage OS, at least in the near future.

Q What's happening with Cyanogen Inc. now that the community's left?

A Most of the staff have gone, many through redundancies, but the company is continuing, at least for the time being. The new strategy is to leverage the technology they developed and work with phone manufacturers to improve the Android experience on their handsets. In other words, they're no longer developing a whole mobile OS.

Q Is Lineage OS ready to use yet?

A Not really. If you're comfortable building an OS for your device, then you may be able to get something working from the source code [3], but this is more involved than a simple `./configure && make`, so for most people it's best to wait for official builds to be ready. At least, that was the status at the time of writing. This is a fast-changing situation, so by the time you read this, it might be possible to get pre-built images. Check the Lineage OS website [4] for the latest information (see Figure 2).

Q You mentioned Replicant earlier. What's that and how does it fit in with CyanogenMod and Lineage OS?

A CyanogenMod was, and Lineage OS will be, an open source OS in the same way Ubuntu, Fedora, openSUSE, and most Linux distributions are. That means they contain almost all open source software, but there are a few bits of precompiled software (known as blobs) that are closed source. Typically, these are device drivers. Use one of these systems, and you're using 99% open source, which is fine for many people, but some people are willing to sacrifice some functionality to run completely open source software. For these people, there are desktop distributions like Trisquel. Replicant is a project based on CyanogenMod that includes only completely open source software. This means that it can't run on as many devices (since there aren't open source drivers for all hardware pieces), but when it does run, it's more free. We strongly suspect that future versions of Replicant will be based on Lineage OS, but that remains to be seen.

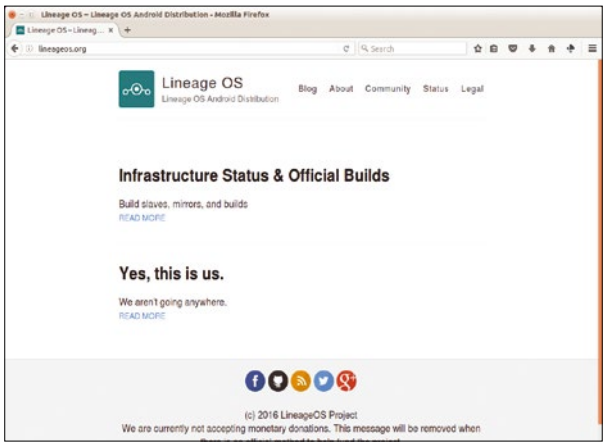


Figure 2: In this fast-changing situation, the Lineage OS website is the place to find the latest information.

Q CyanogenMod sounds like it was a huge project, and one that's hugely important to the general free software community. With its commercial sponsor dropping support, what can we in the community do to help it keep going?

A That's very community minded of you! With a project the size of Lineage OS, the infrastructure costs can get quite large. At the time of writing, the project doesn't have the legal setup necessary to take monetary donations. (Although, again, this could have changed by the time you read this, so head to their website [4] to get the latest information.) However, they do need servers. If you can spare a machine with a decent Internet connection, visit the donation page [5] to see if you can help. ■■■

Info

[1] CyanogenMod wiki: *https://web.archive.org/web/20161120060316/http://wiki.cyanogenmod.org/w/Main_Page*

[2] CyanogenMod images: *https://archive.org/details/cmarchive_snapshots*

[3] Lineage OS source code: *https://github.com/LineageOS*

[4] Lineage OS: *http://lineageos.org/*

[5] Lineage OS server donation: *http://lineageos.org/Infrastructure-Status-and-Official-Builds/*

Valentine Sinitsyn develops high-loaded services and teaches students completely unrelated subjects. He also has a KDE developer account that he's never really used.

CORE TECHNOLOGY

IPv6 is the future of the Internet, and it promises many goodies. Discover what your Linux box can do about it today.

BY VALENTINE SINITSYN

IPv6 in Linux

IPv4, today's Internet workhorse protocol, offers more than 4 billion IP addresses. This may seem like a lot, but in a world of 7 billion people and every single toaster seemingly wanting an Internet connection, it's actually not. In fact, IANA, the Internet's numbering authority, allocated the last block of IPv4 addresses about six years ago. That doesn't mean there are no spare IPv4 addresses left on the planet – regional operators still have some reserves – but the supply already has been exhausted.

IPv6 comes to the rescue. With 128-bit addresses – that is, about a quadrillion IPs per every human body cell in the world, we'll hopefully be on the safe side for some time. However, IPv6 offers much more than extra bits in the address; it fixes a 35-year irritant of IPv4 operation by allowing the network to function without network address translation (NAT) or DHCP. Moreover, it enjoys being a first-class citizen in your Linux box.

Back to Basics

In general, IPv6 addresses are shown as eight 2-byte, colon-delimited hexadecimal numbers (e.g., fe80:0000:0000:0000:eef4:bbff:fe29:873d). So many numbers are awkward, so a few simplifications are possible. First, you can omit leading zeros: 0001 is the same as 1, and 0000 is the same as 0. Second, you can drop running sequences of all zeros, which are quite common in IPv6 addresses. You can do this only once per address for the longest sequence, though. A double colon shows where zeros were: consider fe80::eef4:bbff:fe29:873d or even ::1, which is just a loop-back address, serving the same purposes as 127.0.0.1 in IPv4. This means a single IPv6 address may have several equally valid representations. If you think IPv6 addresses are more difficult to parse than IPv4, that's true. Luckily, libraries such as *glibc* handle this for you.

Recall that you also use a colon to separate hostnames and ports, as in 192.168.0.1:3128. To make this work for IPv6 addresses, enclose them in square brackets; for example, try entering *https://[2001:4f8:1:10:0:1991:8:25]:80* in your browser. Note this applies to raw addresses only. If a DNS name resolves to IPv6 address via AAAA (aka a "quad-A" record) [1], you don't need any brackets.

As in IPv4, IPv6 addresses are split into two parts. Higher order bits are known as a routing prefix, and there are some well-defined prefixes, as I'll explain shortly. The lower part, which is almost always 64 bits long, specifies the host or interface ID. Bits in between may encode a subnet or serve other purposes, depending on the address type.

A regional regulator could allocate a /32 prefix (e.g., 2a02:06b8::/32) to your provider. The provider then extends it to identify their customers. If it uses 16 bits, the provider can have more than 65,000 distinct clients. "Provider" bits may carry some information; for example, a most significant bit of *1* could indicate the customer is an organization, not a private individual. Or the provider could use the most significant byte to encode a city in which the customer is located. Although this bit affords many opportunities, ultimately the provider allocates a /48 prefix and hands it off to you.

This makes a difference. In the IPv4 world, you (a household or a small company) get a single IP

address or two, and that's it. IPv6 strongly encourages allocating prefixes smaller than /64. Address space is "cheap" in IPv6 now, and this affects network design. With a /48 prefix from your provider, you have another 16 bits left in the network portion. You could use a higher order byte to number rooms in your house (assuming you don't have more than 256 of them) and a lower order byte to represent a device. For example, :fc02 could be that IoT thing in the attic. In IPv4, all hosts in your house are usually on the same ("gray") subnet, unless you have a really big house. In IPv6, a fridge is often on a subnet of its own.

Addressing

Another difference is that you typically have several IPv6 addresses on your box, which I'll examine one at a time.

The first is the link-local address. It's present even if your ISP doesn't know anything about IPv6, as long as your Linux kernel supports the technology (most kernels today do). Link-local addresses begin with the fe80::/10 prefix, and the host part is calculated from the MAC address of the respective network adapter. Running

```
ip -6 address show
```

on your Linux box should reveal your link-local address. The `-6` switch tells the `ip` command to show IPv6 addresses only. Listing 1 shows an IPv6 configuration on a box with no global IPv6 connectivity.

Link-local addresses are valid only within the link (i.e., hosts plugged to the same Ethernet switch) and are never routed. In Listing 1, you see the real link-local address of my laptop, but you can't ping it by that address (sorry). They are still useful for cases like Neighborhood Discovery, which is link-local by its nature, or auto-configuration: Obviously, you can't use a global routable IPv6 address while you are trying to obtain one. Link-local addresses also are good for ad hoc networking.

Other "local" IPv6 addresses include a unique local IPv6 unicast address (ULA), or local IPv6 for short, is a private IPv4 address equivalent. The prefix is fd00::/8, which is followed by a random-generated global ID (Figure 1). Unique local addresses are provider-independent and shouldn't be routed over the Internet. A single prefix makes them easy to filter on border routers. You are free to use ULAs the way you want, on their own or together with global routable IPv6 addresses, which belong to the 2000::/3 prefix; that is, they always begin with 2. (See the "Troubleshooting IPv6" box.)

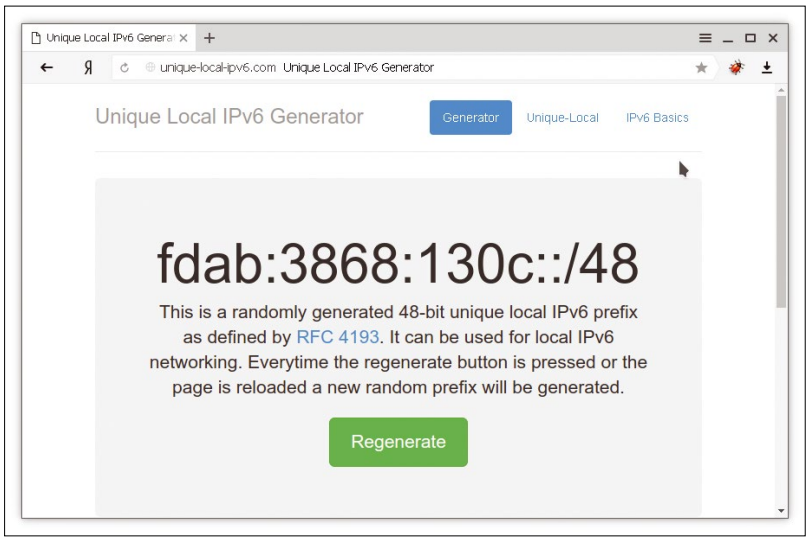


Figure 1: Websites like this help you generate a properly randomized, /48-bit-long ULA prefix.

DHCP, Begone!

An IPv6-enabled box gets one (link-local) address "for free," but what should it do with the rest? IPv6 was designed to benefit from auto-configuration mechanisms. With the Stateless Autoconfiguration (SLAAC) technique, you get a fully operable IPv6 network with no configuration except on routers.

A secret ingredient is ICMP6. You know and love ICMP because it makes `ping` and `traceroute` work, but IPv6 assigns ICMP

Listing 1: A typical IPv6 configuration

```
$ ip -6 addr show
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 state UNKNOWN qlen 1
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
3: wlan0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 state UP qlen 1000
    inet6 fe80::fef8:aeff:feeb:866f/64 scope link
       valid_lft forever preferred_lft forever
```

Troubleshooting IPv6

If you ever troubleshoot network problems, `ping` and `traceroute` (or `tracepath`) are your good friends. You can expect them to help you with IPv6 bugs as well, but wait!

```
$ ping 2001:4f8:1:10:0:1991:8:25
ping: unknown host 2001:4f8:1:10:0:1991:8:25
```

This return is bad news, really. How could it be that `ping` doesn't recognize a global IPv6 address?

In many Linux distributions, `ping` is an IPv4-only tool. The same holds for `trace`-whatever. What you want here is `ping6` and its friends. As the suffix suggests, they're IPv6-enabled. Ping your link-local address to check this:

```
$ ping fe80::fef8:aeff:feeb:866f
connect: Invalid argument
```

What's wrong now? Here, `ping6` isn't smart enough to deduce which link (or network interface) you are referring to, so you need to add an interface number (as reported by `ip link show`) as a suffix:

```
$ ping fe80::fef8:aeff:feeb:866f%3
```

Now everything should work as expected.

If your distro ships newer *iputils* packages, the above may not apply to you anymore. Congrats! Yet, you might want to invoke `ping -6`, just to be sure.

Listing 2: dnsmasq configuration snippets

```
# Do stateless DHCP and SLAAC, and generate DNS names for SLAAC addresses
# from DHCPv4 leases.
dhcp-range=1234::, ra-stateless, ra-names

# Do router advertisements for all subnets where you're doing DHCPv6
# Unless overriden by ra-stateless, ra-names, et al, ...
enable-ra
```

some extra duties. First, ICMP6 matches IPv6 addresses to Ethernet MAC addresses within the local network. This is known as Neighborhood Discovery, and a separate protocol (ARP) was in charge of it in IPv4. By the way, you can examine and change the local neighbors table both for IPv4 and IPv6 with the `ip neigh` command.

Then, there are router advertisements (RAs). When you were connecting an IPv4 box to the network, you had to supply the default gateway, or it wouldn't be able to communicate with hosts outside the local network. IPv6-enabled routers periodically advertise themselves to connected hosts, and these hosts may also solicit advertisements when needed. When you plug in the cable, your box generates a link-local address and uses it to learn where neighborhood routers are; then, it generates a global routable IPv6 address from the router-supplied prefix, and that's it. You are now connected to all the globe, no NAT required.

The Linux kernel doesn't handle router advertisements on its own, though. The good news is `dnsmasq` [2] speaks IPv6 and can do router advertisements for you. It also acts as a DHCP server and caching DNS server for your network. A few configuration samples related to RA and DHCPv6 are commented in Listing 2, and you can find more in the `dnsmasq.conf.example` reference configuration file [3].

Given SLAAC, you might wonder about the role of manual configuration and DHCP in the IPv6 world. Yes, it's certainly possible to assign an IPv6 address with the good old `ip` command:

```
$ sudo ip addr add fd5c:5053:5e0e::1/64 dev eth0
```

As for DHCP, it also exists for IPv6 under the name DHCPv6. Thanks to SLAAC, it's not mandatory anymore, even on corporate networks; yet, it's still deployed for several reasons. First, you might want to convey the name server, the hostname, and other configuration bits beyond raw addresses. Second, you might want a specific IPv6 address assignment procedure, so you always know who is who. The first is often called "stateless" DHCP, whereas the second is called "stateful" DHCP.

Digging Tunnels

To support the customer-faced features discussed here, IPv6 rethinks the on-wire protocol and helps routers parse it quicker. IPv6 drops the convoluted IP options mechanism, replacing it with the so-called "extension headers" that routers ignore most of the time. This means the IPv6 header is always 40 bytes, which streamlines network hardware design and eventually makes routing faster. Moreover, IPv6 routers don't need to update the checksum after they decrement the TTL. IPv6 simply reuses the checksum field in the UDP or TCP payload and excludes TTL from the calculation. Finally, intermediate routers don't fragment IPv6 datagrams anymore. All fragmentation occurs at the source, where the Path MTU (PMTU) discovery process is

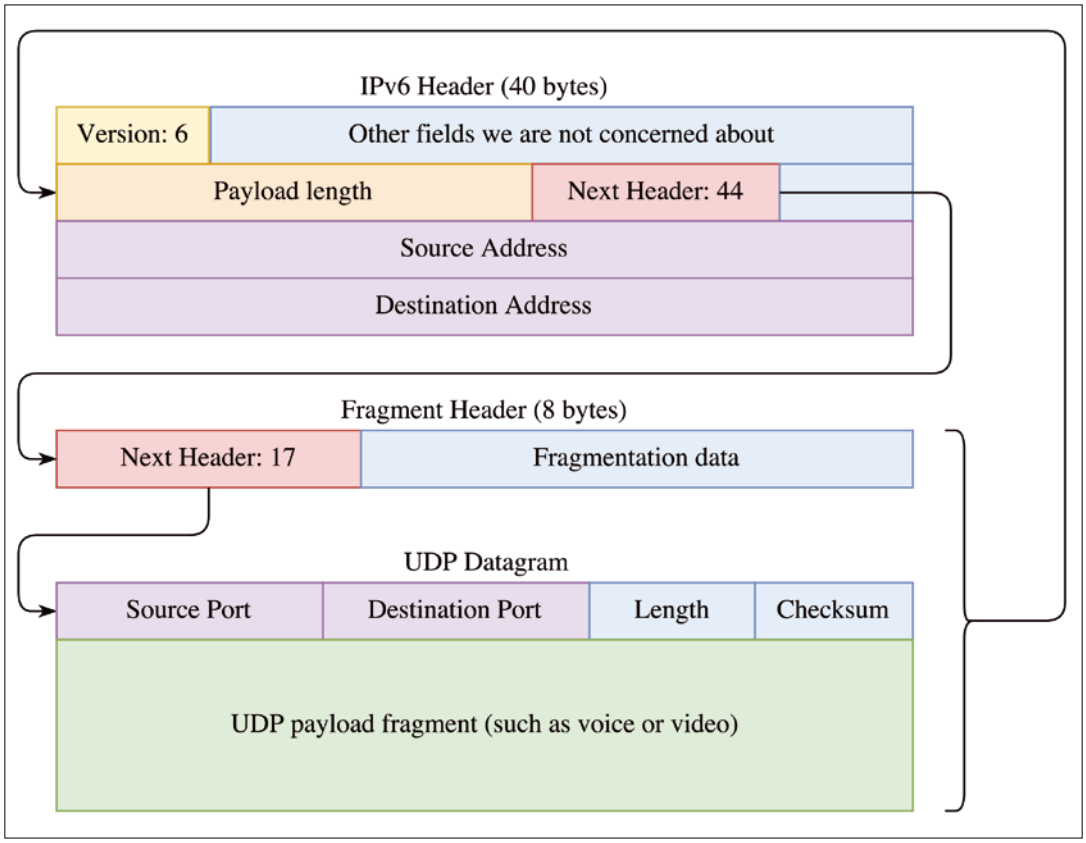


Figure 2: IPv6 header, along with some extensions. This example shows a fragmented UDP packet.

employed to determine the maximum datagram size. PMTU also relies on ICMP6.

Note the Next Header field of the IPv6 header depicted in Figure 2. Normally it stores a transport layer protocol identifier, such as *6* for TCP or *17* for UDP. If the datagram uses extension headers, Next Header tells the extension header type. Extension headers have their own Next Header fields that are used for "chaining," which is how fragmentation and security extensions (which were backported to IPv4 as IPsec) work in IPv6.

IPv4 and IPv6 also have their own protocol numbers (Listing 3; taken from Ubuntu 16.04). What happens if you put *41*, a protocol value for IPv6 itself, in the Next Header field?

You get an IPv6 datagram encapsulated in another IPv6 datagram. For IPv4 in IPv6, a value of *4* (IP-ENCAP or IPIP) does the trick. This is known as IP tunneling. It's not a technology you are likely to find in your living room, yet it comes in useful when you want to connect two IPv4 networks over an IPv6 transport network, or you could employ it as a simple network virtualization mechanism. You have a physical IPv6 network using one addressing scheme (often out of your control). When you create tunnels between hosts, you need to communicate and use the addressing you want. For instance, a load balancer can create tunnels to deliver traffic to your back-end servers as if they were connected directly. Despite the name, IPv6 tunnels don't add any security by themselves.

Linux supports IPv6 tunneling via the stock `ip6_tunnel` kernel module, and you create and manage tunnels with the `ip` command (Listing 4). Here, I create and set up the `ip6tnl1` network interface, which encapsulates both IPv4 and IPv6. This is known as an `any/ip6` tunnel. You should also tell `ip` the addresses for both the remote and local ends of the tunnel and then assign the `ip6tnl1` interface some address, as I showed earlier. You also might want to add an explicit network route because tunnel interfaces are point-to-point devices, so it doesn't happen automatically.

Listing 3: Selected protocol numbers

```
$ cat /etc/protocols | grep -E 'ip|tcp|udp'
ip      0       IP              # internet protocol, pseudo protocol number
ipencap 4       IP-ENCAP        # IP encapsulated in IP (officially ``IP'')
tcp     6       TCP              # transmission control protocol
udp     17      UDP              # user datagram protocol
ipv6    41      IPv6             # Internet Protocol, version 6
...
```

Listing 4: Creating IPv6 tunnels

```
$ sudo ip -6 tunnel add mode any remote <remote peer> local <local IPv6>
$ sudo ip link set up dev ip6tnl1
$ sudo ip addr add fd5c:5053:5e0e::1 dev ip6tnl1
$ sudo ip -6 route add fd5c:5053:5e0e::/64 dev ip6tnl1
```

Repeat the process on another box but reverse the local and remote peers and use fd5c:5053:5e0e::2 as an address. Now you should be able to ping both peers. If you spot an `ip6tnl0` interface, please ignore it; it is there for internal reasons.

The Other Way Around

Alas, not all of the world is IPv6-ready yet. This varies from country to country, and although my ISP has offered IPv6 for a long time, you might not be that lucky. Things change, fortunately, but transition takes time, and for a time, we'll have to live in a dual-stacked world.

There are several ways to "convert" your IPv4 address to IPv6, and most of them rely on tunneling techniques similar to (but different from) those I covered in the last section. The most common mechanism is 6to4 [4]. It implies you have a public IP address, which is not the case for many of us behind our NATs. My personal favorite is Teredo, an open protocol (RFC 4380) developed by Microsoft well before it open-sourced .NET Core and friends. The protocol encapsulates IPv6 datagrams in UDP and delivers

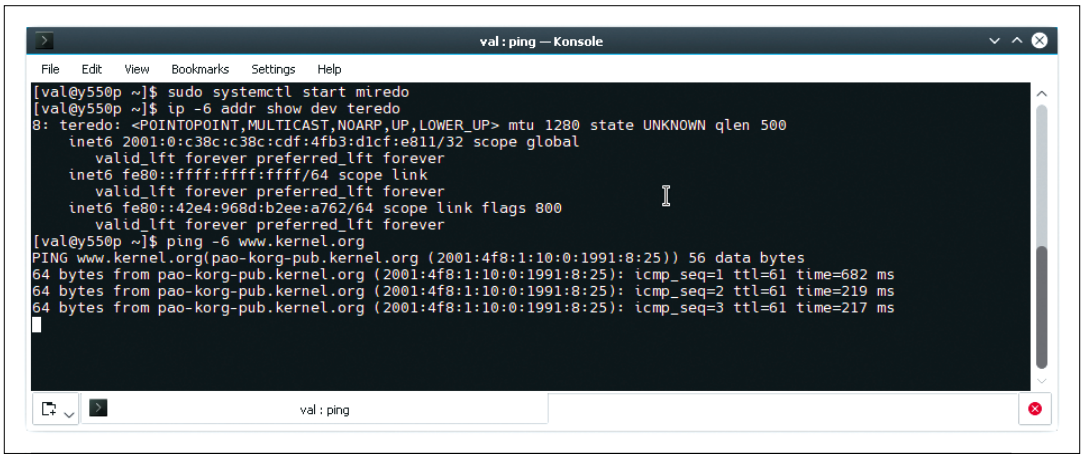


Figure 3: With Miredo, you can enjoy IPv6, even if your ISP doesn't support it yet. It also works great for NAT traversal.

Figure 4: Tayga is a quite popular NAT64 implementation for Linux. Like Miredo, it uses TUN devices and runs in userspace.

them over IPv4 networks to Internet servers, typically listening on port 3544. Teredo has its own IPv6 prefix (2001::/32). The other 96 bits encode client and server IPv4 addresses and some auxiliary information.

A de facto standard GPLv2-licensed Linux implementation is called Miredo [5], which you are likely to find in (maybe community) repositories, but it's equally straightforward to build yourself. Miredo is a pure userspace Teredo implementation relying on TUN devices for networking machinery. It usually installs itself as a system service and creates a `teredo` interface when you start it. The configuration file is usually at `/etc/miredo/miredo.conf`, but it's unlikely you'll have to adjust anything there.

You might have to make some adjustments if you try to use Teredo. Because Teredo is Microsoft technology, the company used to have a server farm to support it. Apparently, it's torn down now, because Teredo is considered deprecated (everyone should have native IPv6 connectivity today). The config file lists some other options (and you can google a few others), yet it points to *teredo.remlab.net* by default. This server is maintained by Miredo authors (if anybody), and it is for testing purposes only. Don't forget to use something else if you plan to deploy Miredo for real.

With `teredo` up and running (see Figure 3), your host now has global IPv6 connectivity, and it's reachable even if it is behind NAT, which makes Teredo a viable alternative to port forwarding (see the box "A Word on NAT"). If you drop a dynamic DNS into the mix, you can access your home NAS from wherever you are. Everyone else can do it too, though, so make sure you install the latest updates and use strong passwords. With great power comes great responsibility, you know. ■■■

A Word on NAT

In today's IPv4, NAT is almost a must. IPv6 also has several NAT flavors: NAT46 (translates IPv4 to IPv6), NAT64 (the reverse; Figure 4), and NAT66, but their usage is discouraged. With address space so abundant and multiple addresses on a box being a norm, there is no need for "gray" addresses in IPv6 networks anymore, and you have better ways to bridge IPv4 and IPv6 networks, as I explain here. Many NAT46 and NAT64 implementations (e.g., Tayga [6]) are userspace, which affects performance. Some might argue that NAT hides network topology and adds some security. That's questionable. The real use case for NAT is when you connect an IPv6-only box to an IPv4-only host (e.g., *github.com*) over the Internet. For everything else, there is often a better alternative.

Info

[1] AAAA record: *https://en.wikipedia.org/wiki/IPv6_address#Domain_Name_System*

[2] dnsmasq homepage: *http://www.thekelleys.org.uk/dnsmasq/docs*

[3] dnsmasq.conf.example reference configuration file: *http://www.thekelleys.org.uk/dnsmasq/docs/dnsmasq.conf.example*

[4] 6to4 overview: *https://en.wikipedia.org/wiki/6to4*

[5] Miredo homepage: *https://www.remlab.net/miredo/*

[6] Tayga homepage: *http://www.litech.org/tayga/*

Now Appearing on
APPLE NEWSSTAND

New age convenience...

Our inspired IT insights are only a tap away.

Look for us on Apple Newsstand and the iTunes store.

Download a FREE issue of each publication now!

FOSSPicks

Sparkling gems and new releases from the world of Free and Open Source Software

Graham tears himself away from updating Arch Linux to search for the best new free software. BY GRAHAM MORRISON

Music tracker and sequencer

Radium 4.3.5

Linux has become an amazingly creative environment, and if you take a look at some of our picks for this issue, it seems no more so than in the realm of audio and music. There are some real gems, and Radium is one of them. Radium is a tool for making music that's a hybrid of audio workstation and old school tracker, but it's also unique in both its fusion of cutting edge features and the way the user interface is designed. It's the user interface where all these ideas come together. The main view has vertical tracks for audio, just like a music tracker running on an Amiga in the 1980s. But instead of those columns behaving like a spreadsheet for musical data, which is how those old trackers worked, Radium includes rendered waveforms, clips of audio, and automation. It's like a merger between Cubase and Renoise.

Getting Radium is a little tricker than it should be. The project is proudly open source, with its code hosted on GitHub, but finding binaries for your distro is going to be difficult. This is because, much like Ardour, package builders have respected the developer's wish in asking users to pay for a binary download. That means that if you want a quick tryout of the software, you'll need to download and run the demo. This limits exports and the number of plugins. If you're not so short of time, you can install the required dependencies and try to build the project yourself. I gave up on an Ubuntu 16.04-based system, as it seemed impossible to track all the dependencies down, but Arch users fortunately have an AUR package, which is what I used eventually. Running the application is then just a simple case of disabling your system audio while you get JACK to work, which is never that simple.

1. Tracker. Radium can look and be used like an old-school tracker. 2. Audio. Samples can be automated along a vertical timeline. 3. MIDI. Control remote synths with a vertical piano roll view. 4. Mixer. Drop modules and pipe audio by drawing lines in the modular mixer. 5. Block view. See the note data and the play head. 6. Effects. Control parameters and import lots of different effects. 7. Sampler. Edit sounds without exporting them to a different tool. 8. Tracklist. Much of the UI is similar to a text-based sequencer.

It's worth the effort. Radium forces you to take a unique approach to making music. Much like a tracker, you start off with blocks of sequences and samples that you build into a song structure. Despite looking like something designed for Motif, the UI is fast and responsive. Many elements are updated in real time during playback. Drag a line to make tempo adjustments, for example, and block sizes change accordingly. Playback smoothly speeds up or slows down, and volume and CPU meters bounce around to give feedback.

Edit mode allows you to make changes, while numerous keyboard commands control playback. Effects and processing are accomplished with a unique modular mixer, which allows you to link up tracks with effects and processing, just as you would wire the real equipment in a studio, complete with real-time feedback. You can then construct complex effects chains with both LADSPA (Linux Audio Developer's Simple Plugin API) and Linux VST effects. The Linux version in particular is unique, because it lets you add elements you've constructed with Pure Data. Pure Data is the forerunner of Cycling '74's Max, a visual programming environment for audio and control data, and it lets you write any kind of process if you have the skills. It's a great addition that's only seen on Windows and OS X in something like Ableton Live, taking tinkering to a new level on Linux. ■■■

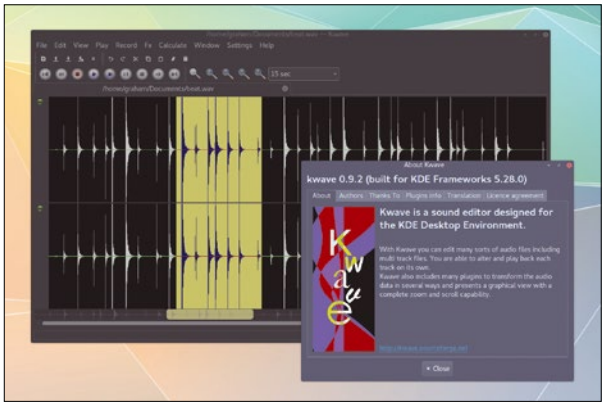
Project Website
http://users.notam02.no/~kjetism/radium/

Sound editor

KWave 0.92

Audacity is a brilliant audio editor, and it's now used by professionals on all kinds of operating systems for everything from cutting silence to preparing mastered audio files for compact disc. It's what open source software should be about. Before Audacity came along, the same functionality cost serious money and was inaccessible to Linux users. But Audacity is often too complex for simple jobs, such as adding a fade-in or fade-out or examining the contents of an audio file, which has left me often wanting a simpler, quicker alternative. Considering audio often goes hand-in-hand with images, there are very few alternatives. One of the exceptions is KWave, a sound editor that started life almost 18 years ago but, sadly, hadn't been updated for a while … until now.

After a string of updates, KWave is back. It's been ported to KDE Frameworks 5 and has had its 'K' position ratified by becoming a member of the official KDE Applications 16.12 bundle. This is great news for KDE users or almost anyone wanting a simple audio editor without needing to dive into Audacity, and KWave definitely doesn't compete with in terms of features. There's only a limited set of effects; for example, recording is very limited when compared with Audacity's multitrack capabilities, and there's no plugin support. But it does allow you to export an audio file with a simple click of *Save*, and cutting and pasting the audio itself is the same process. You can even zoom into a waveform to examine the specific bits within a wave, which is useful if you're editing out glitches. There's also some good options for saving a raw data file you know is audio, because KWave can change the sample format without forcing a conversion on the current data. ■■■



Finally, you have an alternative to Audacity if all you need is a quick and dirty tool for editing audio.

Project Website
http://kwave.sourceforge.net/

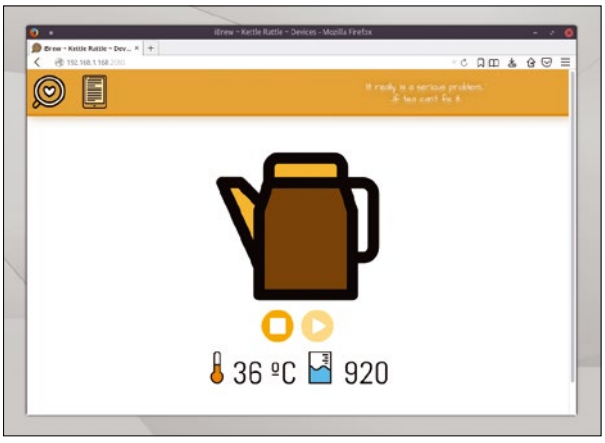
Smart kettle

iBrew

It may seem crazy, but you can buy a kettle that boils water when you press a large button on your phone. It's called the iKettle 2.0. What's even crazier is that the mobile app requires a remote user account, and more importantly, the connection to your kettle barely works. You may press the button, but when you walk all the way to the kettle, your water may be cold. Welcome to the Internet of Things (IoT).

Fortunately, like Linux itself, nothing exists for long without being deconstructed and rebuilt, and a brilliant little tool called iBrew rebuilds the iKettle into something much more functional. It's both a command-line utility and a web interface for all your iKettles and Smarter Coffee machines, and it does a far better job than the official tools. To start, you don't need an account. Just type *ibrew boil* and heat will be applied until your water is hot. It's as simple as that, and apart from needing the app to connect your device to WiFi initially, you no longer need accounts or proprietary software. Nor is this a simple tool. More than 50 other commands can be used to hack and better integrate these devices into whatever home automation system you're using. To keep the water hot, for example, you specify a specific target temperature (brilliant for green tea) and measure the water in the device. The kettle is protected from boiling empty, so these commands should be safe.

Another command will even launch a web interface that is much clearer and quicker to use than the app, and this can be used as both a conduit for the app itself, sidestepping multiple user issues with the original app, and as a RESTful API for easy integration with your other services. ■■■



The future is now – turn on your kettle from the command line and heat water to that perfect green tea temperature.

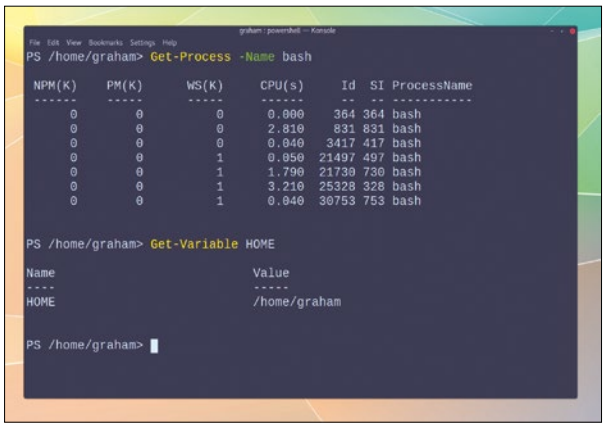
Project Website
https://github.com/Tristan79/iBrew

Frozen (s)hell

PowerShell 6 (alpha)

Yes, this is a Microsoft product. Yes, it's open source (MIT, with the exception of the Windows components on the Windows version). Yes, it uses Mono to some extent. And yes, you may think Microsoft releasing an open source shell that works on Linux is a little like Linux users porting Wine back to Windows. But there are some good reasons for PowerShell and good reasons for using it on Linux. The Linux Voice podcast team were reminded of this recently when we happened to make some throwaway comments about PowerShell, and one of PowerShell's engineers, Jeffrey Snover, was good enough to respond to our nonchalance in the podcast's comments. He pointed out that PowerShell's real benefit for Linux users comes when you're dealing with structured data and APIs, such as JSON and REST, running on different hybrid clouds – especially Microsoft's. PowerShell is designed to handle these transactions natively, in a similar way Bash is designed to handle POSIX natively. After all, this is how Microsoft's Azure is using Linux and is the reason for a lot of its kernel development and other investments. As Snover puts it in the podcast comments, "The goal is to make it simple and easy for customers to consume as much computing as makes sense for their business."

It may seem weird running PowerShell on Linux, but this makes sense because there isn't a shell I know of that's great at dealing with cloud-focused technologies. Instead of local scripting, for example, PowerShell scripts use .NET to pull in all kinds of external APIs, from Exchange and SQL Server to VMware vSphere. While there's no doubt these are Microsoft-centric, many are already industry stalwarts, and having a tool that talks to them natively from Linux is definitely a good thing. ■■■



Troll your friends by replacing Bash with Microsoft's PowerShell.

Project Website

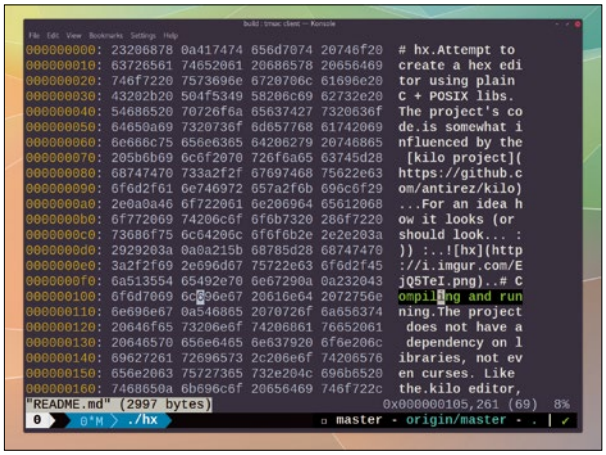
https://msdn.microsoft.com/en-us/powershell

Hex editor

hx

Even if you're not a developer or a "hacker," sooner or later you'll need to use a hex editor. This is primarily because a hex editor is a bridge between the worlds of code, binary, and content, allowing you to open and view a file regardless of its format or whether the file is corrupt or complete or not. The file could be an executable binary, or it could be a LibreOffice document corrupted whilst saving. Either way, a hex editor will gladly ignore the context of a file and happily display its contents. Because the context has been lost, that display usually defaults to hexadecimal values, or base 16, representing the raw binary contents of a file. Thanks to your computers' binary logic, this single hex value is a "nibble" of data, usually grouped into pairs to form a "byte." Bytes are also turned into ASCII text, so you can read raw data if necessary, and those bytes in turn are grouped into columns containing 64 bits per column. This makes finding a specific location or offset much easier, whether that's in your computer's raw memory or within a file.

As a new project, hx is the beginnings of just such a hex editor that runs from the command line. It's tiny and compiles almost instantly thanks to a single dependency on the ordinary C POSIX libraries. This is exactly what you need because you often use hex editors for trawling through large dump files, virtual devices, or executables, and you need great memory management and performance. Its editor is vim-like, where you can switch between normal mode and command mode. Navigation keys are also the same as vim, and you should be able to start editing without referring to the excellent man page if you're already familiar with vim. This means you can search, update, edit, insert, and replace right from the command line, working with binary just as you can with text. ■■■



Hex editors can reveal all kinds of information about binaries and system memory.

Project Website

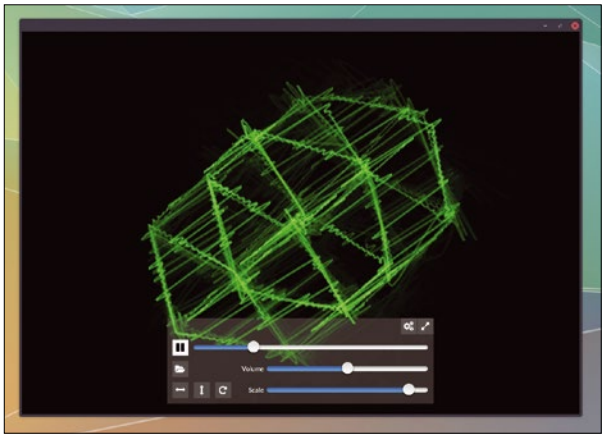
https://github.com/krpors/hx

Audio visualizer

Oscilloscope 1.0.7

The first audio I saw visualized on a computer screen was via a cheap digital audio sampler on the Commodore Amiga running Aegis Audiomaster II. Audiomaster had a function that would monitor the (stereo, 8-bit) live input on the sampler and use the audio signal just like source voltages for a software oscilloscope. Most of us think of an oscilloscope as a small, square CRT screen housed within a large brick-like box, typically found within a laboratory. The CRT would display the measured values from a couple of inputs mapped across the *x* and *y* axes in bright green, and while Audiomaster's oscilloscope also tracked the *x* and *y* (left and right inputs) in green, it couldn't be used to reverse engineer circuits. But you could see the effects of frequency modulation on a low-frequency sine wave, and the results were fascinating.

Over the years, there haven't been many purely software oscilloscopes. When they are developed, they're usually designed to go with hardware that's better equipped for high-frequency changes in voltage rather than changes in audio. But audio oscilloscopes also have a long tradition and are still used by modular synthesis geeks and audio engineers, and this software, called Oscilloscope, is the best one I've seen for years. What's brilliant about Oscilloscope is that it faithfully recreates the 1970s characteristics of those CRT oscilloscopes, particularly with the way it renders the waveforms



Oscilloscope music both sounds great and does clever things with an oscilloscope.

and the persistence of the screen. And because Oscilloscope is designed for audio, it supports high sample rates (e.g., 19,200), it will load audio files for playback, and recent versions will monitor audio input, too. With some clever audio routing, you can also use it to visualize real-time software synthesis and other software audio sources. Give it a try. ■■■

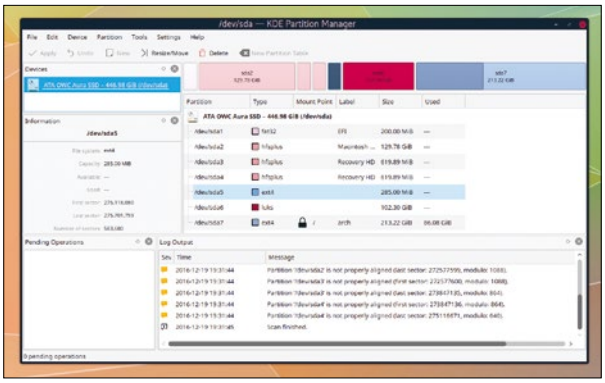
Project Website
https://asdfg.me/osci/

Drive partitions

KDE Partition Manager 3

If there's one thing you need when you write about Linux, it's a good partition manager, because writing about Linux gives you the perfect excuse to install new distros, break old ones, and update the un-updateable. Sooner or later, your drive will need tearing down and repartitioning. GParted is brilliant for this. It does everything you need and is available by default on almost every Live CD and USB stick. KDE Partition Manager, which is very similar to GParted, performs many of the same functions, although it also competes with GParted and as a result does a few things that GParted doesn't. After a few great recent updates, for example, it was the first GUI tool that could resize my encrypted partitions – a feature that should be at the top of everyone's list, now that we're all using encryption.

Recent updates have culminated in a major revision with the release of version 3. LVM specifically has received a lot of love. KDE Partition Manager now supports both LVM on LUKS (encrypted partitions) and LUKS on LVM; the creation of new LVM volume groups, along with the ability to resize both logical and physical logical volumes; and the removal of physical volumes from the LVM volume group. More can now be done without booting to an external USB stick, too. Ext4 partitions can now be grown, for example, whereas Btrfs partitions can be both grown and shrunk. Like GParted, KDE Partition Manager also works just as well without KDE. KDE in general has been



It may not be as well known as GParted, but KDE's equivalent has a host of additional features.

able to dump the huge library dependencies that used to blight its cross-desktop credentials, which may make KDE Partition Manager an even better match as a replacement for GParted, especially if you're dealing with encrypted data and LVM partitioning schemes. ■■■

Project Website
https://git.stikonas.eu/andrius/partitionmanager

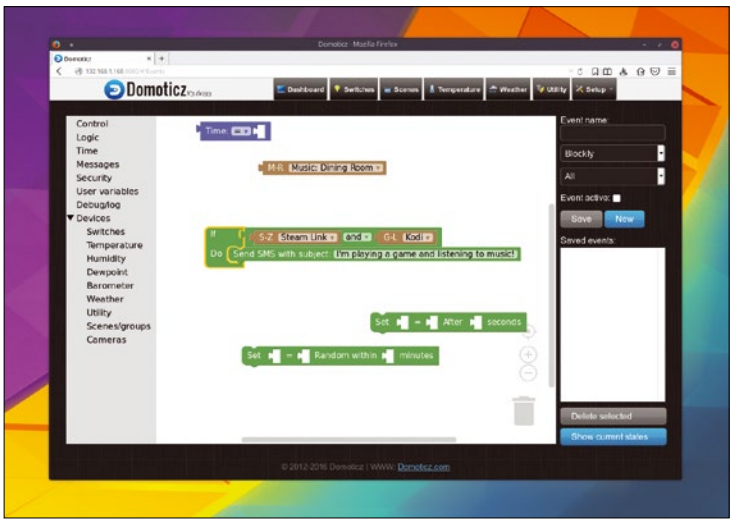
Home automation

Domoticz 3.6

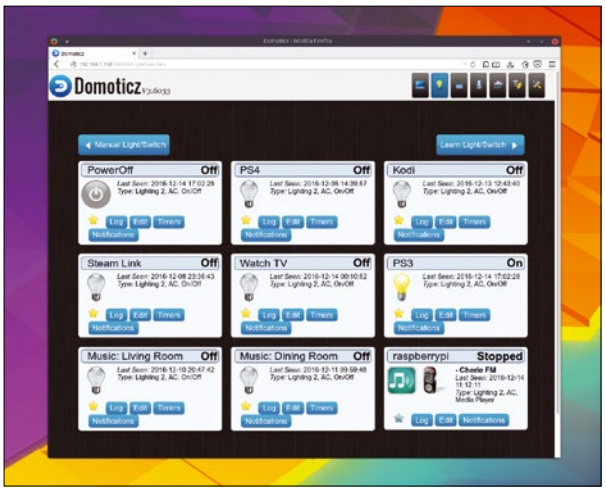
Despite many serious concerns about the wisdom of installing lots of Internet-enabled devices in our homes (proven, somewhat, by the IoT DDoS attacks of October 21, 2016), I'm certain home automation will continue along its path of massive growth. There's good reason for this – home automation is both magically addictive and genuinely useful. From reducing heating costs to creating dynamic entertainment environments, you can now geek out over your home in the same way you used to geek out over a MythTV configuration.

The best way to circumvent security and privacy issues is, of course, to use open source software. You don't always get the convenience of a one-click installation or an accompanying smartphone app, but you get control and configurability. Rather than create lots of different solutions for each of the devices you want to control, it's far better to use a single hub that can either be made to talk to these devices or has support built-in. And that's exactly what Domoticz is. It's a piece of software that's light enough to run on a Raspberry Pi and works with lots of hardware and software you might use when automating various parts of your home.

As you might expect from something that's capable of controlling both your thermostat and your amplifier, Domoticz is relatively complex. You start off by adding your hardware; one of its best features is that it communicates with dozens of devices without requiring any further configuration. Devices include OpenZWave USB, RFXCOM transceivers, Panasonic televisions, plenty of home brew adaptors for the Raspberry Pi, and lots of software like Kodi, Squeezebox, and online weather APIs. You can configure sensors, adding details like temperature or humidity to your system so that you can build events around their values and changes. The project also supports the widely used Philips Hue Bridge, which is great for lighting, but also because the protocol has been reverse engineered and re-implemented by many open source projects, making the Hue Bridge a great gateway through Domoticz to other services like Amazon Echo and IFTT.com.



Event scripting is accomplished with a visual editor that works much like Scratch.



With Domoticz and an open source Philips HUE-compatible bridge, you can even automate with voice via Google or Amazon Echo.

All of this is great for remote control, where you can build triggers, use your phone to switch things on and off, and link devices together into convenient groups. But the real power comes from creating your own events. You can scrape weather forecasts off the web for forward planning or heating schedules, for example. Set timers and triggers, and like a super-powered IFTT.com (If This Then That), you can build your own event-driven scripts using a brilliant visual editor that works just like Scratch. You can use this to send yourself SMS messages, push messages on iOS and Android when certain things happen, or turn off one set of switches when a door closes or opens or your phone gets 1km from the house. The potential is limitless, and it's from this point you slowly become addicted to the idea of home automation as you purchase more and more automatable products and devices. We also highly recommend looking at the offline PDF manual, which is another great addition to the project and the best way of familiarizing yourself with its capabilities before getting started. ■■■

Project Website
https://domoticz.com/

Strategy game

Widelands (19-rc1)

Settlers 2 was one of the best games on the Amiga. It was mostly a resource management game. You'd do things like instruct your four-pixel-high populace to do things like gather wood from an environment that resembles the lowland pastures of Germany's Black Forest or build a pig pen. As your group of settlers became more advanced, you'd build more infrastructure and set up trade routes. This inevitably led to conflict with neighboring tribes, but this isn't why most of us enjoyed the game so much. I enjoyed it because the environments were so well (randomly) crafted, with each element of the game masterfully drawn and augmented with sound. As you scrolled godlike across the landscape, it felt like you'd crafted your own Arcadia. No game since has quite captured that same feeling.

Widelands is an open source game that gets very close to those early Settlers titles. It's currently in a rapid state of development, but more playable than the majority of games still in development. There's an excellent tutorial mode, for example, to help get you started, with basic control, the economy, and even seafaring, as well as the inevitable warfare. There's also a campaign mode, where a story guides you through a series of maps and missions. And there are plenty of single maps to try, each with a different set of challenging conditions. The best thing about Widelands is that the pace remains slow, just like those early titles, and the immersion is helped a great deal by the wonderful artwork and sound. When you get good at the game, an editor will let you create your own levels, and you can even challenge players online. The depth and playability of Widelands is brilliant. ■■■

In Widelands, wind blows through the leaves in the trees, and wild animals follow your trails.

Project Website
https://wl.widelands.org/

Another strategy game

0 A.D.

If the pseudo-two-dimensional bitmaps of Widelands aren't your thing, but you still enjoy strategy games, then 0 A.D. is a must-play. Thanks to being a commercial game that was later released as open source in 2009, the game is amazingly polished, even in its current state of development. The audio, graphics, and presentation are exactly as you'd expect had you bought the game, and the 3D game engine runs brilliantly. The music is also wonderful for setting each scene. Gameplay is similar to other strategy games, in that you have your locations and resources, which you need to use to your advantage by building bases, creating an army, and researching new items to give you leverage over your competition. The landscapes are beautiful and wonderfully navigable in 3D, letting you move down hillsides, rotate views, and easily select the asset you want to control.

Unlike Widelands, the campaign mode isn't finished, and there are no single-player scenarios to dive into. New players will either have to play each other or the AI on a new map, but most of the hard work needed for these modes are complete, as many other aspects of the game have the depth of a fully fleshed out title. There are 12 ancient civilizations to choose between, for example, from the Athenians to the Mauryans and even the Britons. These can be selected when you start a new match, complete with many different kinds of maps for all different kinds of gameplay – straight out battle, for instance, or strategic defense. Each civilization has its own technologies, territory, buildings, and units, and a game will often evolve like a carefully played game of chess. Even without the campaign modes and missions, the game feels complete when playing other people or the AI, and it's a great testament to the hundreds of developers, designers, and artists that have poured their efforts into the title after it became open source. ■■■

Even the art content in 0 A.D. (November 9, 2016) is open source, released CC BY-SA, and I really wish other designers would do the same.

Project Website
https://play0ad.com/

Understanding System Services with lnav

See what's going on in the background of your Linux box by analyzing the logfiles.

BY BEN EVERARD

Broadly speaking, there are two types of software that run on a machine: interactive software that the user launches and controls and background software that runs quietly doing what it needs to do. The interactive software is what you think you're using most often – it's the web browsers and word processors and all the other bits and pieces you launch when you want to do something. However, if you take a look at the processes on your computer (just type `ps ax` in a terminal window), you'll see that most of the software running isn't anything you launched, but the stuff that runs in the background quietly getting on with what it's doing. For the most part, you don't have to think about this – the software just works, and you get to focus on the interactive software and use the computer in the way you want; however, every once in a while, you have to delve a little deeper and see how this background software is running.

Init Systems

When you turn on your computer, the boot system (either the BIOS or EFI) kickstarts a bootloader (e.g., GRUB) that loads the kernel into memory and launches the initialization system. In the past, this init system was a set of shell scripts that launched all the background processes and made sure everything was running as it should. It's the output from this operation that goes into the syslog. In the past few years, most Linux systems have started using systemd as the init system. One of the biggest complaints about systemd is that it doesn't log to text files, but rather to binary files that are difficult to read. That previous statement isn't completely accurate – in fact it doesn't automatically log to text files, but in most Linux distros that use systemd you'll still find everything that you're used to in `/var/log`.

You can still use lnav even if your distro exclusively uses the binary logging in systemd; you just have to use `journalctl` to convert the binary logs into text logs first. The most basic way to do this is to just pipe all the output into lnav with

```
journalctl | lnav
```

If you do find yourself using this method of getting data into lnav, you might want to take a look through the `journalctl` options, because you can select what output to send into lnav.

Because this software runs in the background, you should never see an error message, a progress bar, or any of the other user interface clues that tell you how interactive software is performing. Instead, it sends all the output to logfiles. These are text files that usually live in the `/var/log` directory (see the "Init Systems" box). These logfiles are designed to be readable by both humans and machines, so they follow a fixed text format, but one that's sufficiently verbose to understand. The main log on most Linux boxes is the system log, which is `/var/log/syslog` on Debian and Ubuntu-based machines and `/var/log/messages` on many others. This is a text file, so you could open it in a text editor. However, it can be very large, so it's usually best to just view the end of it. You can do this from a terminal by typing:

```
tail /var/log/syslog
```

On CentOS and other Red Hat-based machines, you'll have to change `syslog` for `messages`. You'll see that each line corresponds to a single event that a piece of background software wants to record. For example, it might be the software that handles your network connection logging some information about the interface, or `cron` running a periodic command. Each line can be useful, but it can be hard to discern what's going on by just looking at a logfile. Useful information can be hidden by pages and pages of routine output. The simplest way of pulling information out of these logfiles is to use `grep`, which searches text and only outputs lines that match. For example, If you want to find all the lines that contain the text "cron," you can run:

```
grep cron /var/log/syslog
```

There's far more power here than I have space to cover, but things quickly gets complex, especially if you want to analyze dates or numbers, because that isn't easy to do in Linux text processing (see the "Regular Expressions" box).

Regular Expressions

If you're going to be looking into logfiles, then learning regular expressions (also known as regex) is an absolute must. These can be fiendishly complex, but they don't have to be – learning just a few characters can give you quite a lot of power.

Put simply, regexes are strings of text that define a pattern that you want to search for. In the main text, I used the text "cron" to search for those four characters. This is perhaps the most basic usage, as any letter matches itself. The full stop (or period for our readers across the pond) is a wildcard that matches any single character, so `cr.n` will also match cron, but not croon. The asterisk (*) character is a modifier that matches the preceding character zero or more times. It's most useful when combined with the wildcard so that it'll match any block of text (e.g., `cron.*root` will match any line that contains the word "cron" followed by the word "root" regardless of what's between them).

These basics will get you quite a long way, but if you want a more detailed understanding of regular expressions, you can read about them on the Linux Voice website [1].

Alternative Options

Lnav is a great option for home computers and small servers. However, if you have a big setup, it can be a struggle because it's not really designed for handling huge amounts of data coming from multiple servers.

Logstash is an excellent open source log amalgamator that pulls together logs from many computers and pushes them into an ElasticSearch database that can then be viewed using a visualization front end, such as Kibana or Graphite. In many ways, this is the same approach that lnav takes, just at a high-performance level, and typically it requires a dedicated server just to analyze your other servers' logs.

A number of excellent closed-source options, such as Splunk, do a great job of showing what's going on if you're happy using proprietary software in the heart of your server setup.

While these are excellent options for large organizations, I'm not aware of any other system that works as well as lnav for small setups.

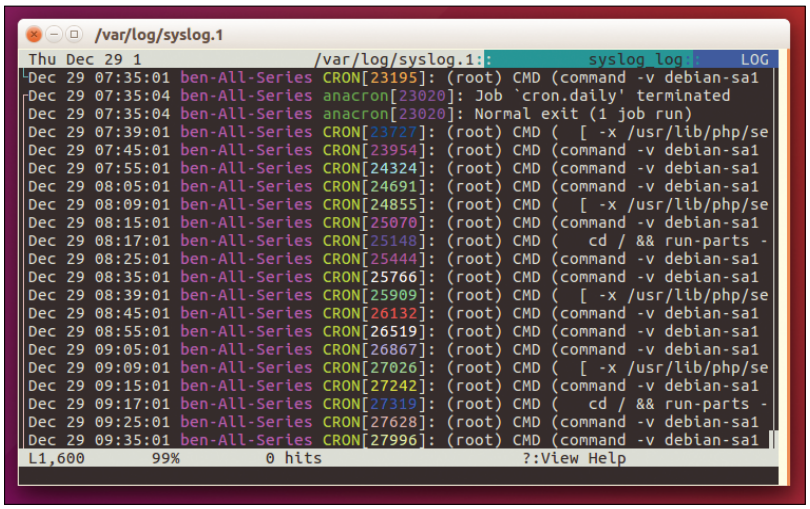


Figure 1: In addition to all the searching options, the colored highlighting in lnav makes logs easier to understand.

A New Hope

Lnav is a tool that hopes to make it easier to extract useful information from logfiles by making it easy to import them into a SQLite-based data store and then perform advanced queries on them simply and efficiently (for larger setups, see the "Alternative Options" box). You may find the tool in your distro's repositories; otherwise, you can grab a statically linked binary that should run on any Linux system from the lnav website [2]. Unzip this file, and you can run this tool from the command line by navigating to the unzipped directory and entering *./lnav*.

There's no setup or configuration for lnav, and it automatically detects most log formats, so you only need to point this tool at the logfile you want to investigate. For example:

```
lnav /var/log/syslog
```

If you're running the statically linked version that you've just downloaded, you'll need to include a path to the executable (e.g., prepend the above command with `./` if you're in the same directory as the file).

At first glance, it'll look just like you've opened the text file in a text editor, but unlike a general purpose text application, lnav understands a little about the logfiles, so it can help you find what's happening. For example, press Shift+W to jump back to the previous warning message and Shift+E to jump to the last error message (without the Shift, *W* and *E* move forward to the next message, but until you've moved back in the file, there won't be anything forward).

If you want to display only lines matching a particular pattern, you can do this with filters. First enter command mode by pressing the colon, and then enter `filter-in cron` to show only the lines containing the text "cron" (see Figure 1). Similarly, you can use `filter-out` to show everything except these lines. These filters will be applied to the

Log Rotation

Usually, programs continually add new lines to the bottom of logfiles as new issues come up that need reporting. Lnav will automatically add these new lines to your session, so you don't need to worry too much about this. However, it can explain why you might get slightly different results when you run queries at different times. Obviously, if programs keep writing data to logfiles, they would eventually fill up the hard drive, and the computer would fail to start (this happens to most sys admins at least once in their careers). To keep everything logged long enough to be useful, almost all Linux distros rotate the logfiles. The exact process varies a lot between different distros (and is often customized by sys admins on servers), but usually each logfile is moved at the end of the day, and a new logfile is started. A certain number of days worth of logfiles are kept, and old ones are deleted (older logfiles may or may not be compressed to save space).

If you have a look in `/var/log/`, you'll probably see that for each logfile, there are several archives (usually with a number in the file name to indicate how many days old that file is). You can bring all these into a single lnav session by using wildcard expansion. For example, to analyze all the syslog files, you can run:

```
lnav /var/log/syslog*
```

Lnav can read both plain text files and compressed files, so there's no need to decompress older files before running this command.

whole file, so until you cancel them (with Ctrl+R), any other action takes place on the filtered log, not the log as a whole.

Perhaps the biggest advantage of lnav over traditional text processing methods for looking through logfiles is that it makes it easy to search by date. You can scroll through by pressing *D* and Shift+D to move forward and backward by 24 hours (see the "Log Rotation" box if you can't find the day's data you are looking for), whereas pressing the numbers *1* to *6* go to the next *n*th minute past the hour (with Shift held down, they go to the previous *n*th minute).

As well as navigating by time, you can also amalgamate data by time. Press *I* to enter histogram mode (see Figure 2). Here, you'll get a line for each time period with the length of the highlighting varying depending on how many log messages there were at the time and the color of the highlighting showing the number of info, warning, and logfiles. In this mode, *Z* zooms in and Shift+Z zooms out. (See Figure 3 for other lnav hot keys.)

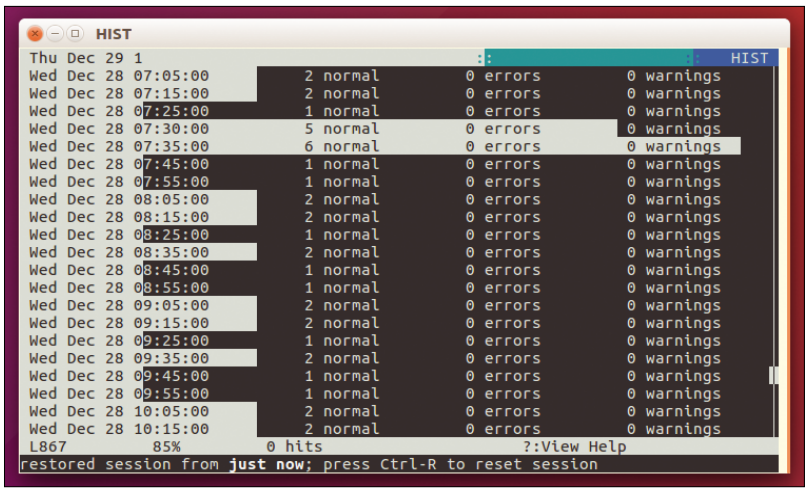


Figure 2: The highlighting in Histogram mode makes it easy to see how much activity has been going on over the log's period.

By far the most powerful part of lnav is its ability to query the logfile using SQL. Press the semicolon to enter query mode, and then you can analyze the logfile using SQL statements. Everything is held in a single table called `logline`, so you can get everything with:

```
;select * from logline
```

If you look carefully, you'll see that this doesn't return all the lines in the table. Since the text lines in a logfile don't all hold the same type of data, lnav can't put them all in the same SQLite table. For example, a `cron` line might include details of what is run and by which user, while a `dhclient` line might include details of what IP addresses are assigned to a particular network interface. Lnav gets around this by only matching lines that match the pattern of the currently selected line (i.e., the top line in the terminal window) when you run a command. Press *P* before entering query mode to see the available columns for this log type (see Figure 4).

I don't have time to delve fully into SQL here, but many of you will already be familiar with this database language, and SQLite supports all the major features, including aggregation, which can be a really useful tool for building up a high-level overview of what's going on in your logfiles. If you don't know SQL, then you can still find out most useful information using filters and the histogram mode in lnav.

One of the best features of the Linux command line is the ability to pipe the output of one command into the input of another. A really simple example of this pipes the bottom 10 lines of a logfile into a search for the text "cron," which you can get by running the command:

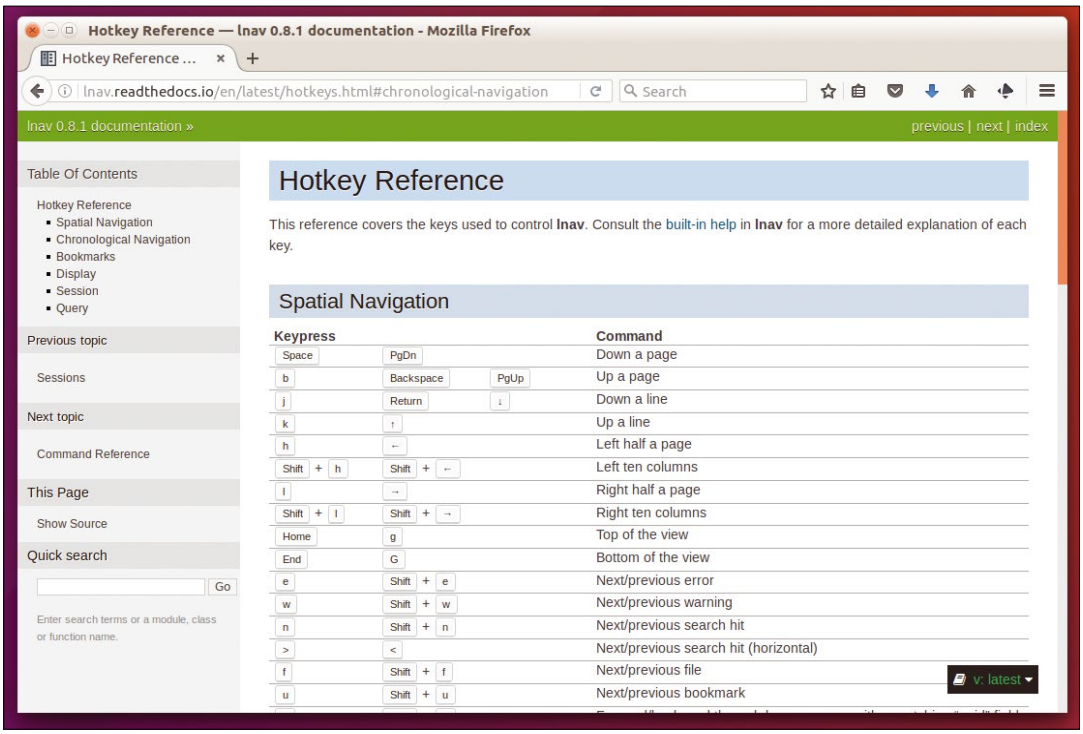


Figure 3: Lnav has a lot of features, but they're well documented [3] if you are having trouble.

```
tail /var/log/syslog | grep cron
```

Fortunately, you don't loose this ability when using lnav, and you can send output to other programs. There are two commands for this: `pipe-to` and `pipe-line-to`. The first sends all the lines that you've bookmarked, and the second sends just the current line. Before looking at how to use these, you first need to know how to bookmark lines. The basic commands for this are `m` (which marks the current line), Shift+M (which marks all the lines between the last marked line and the top of the screen), and Shift+J (which marks the next line). When you mark a line, it becomes highlighted in the user interface, so you can see at a glance which lines are selected. Once you've selected the lines you are interested in, you can enter command mode by pressing the colon. The `pipe-to` command can be followed by any Linux commands, and it will send the data to them. For example, if you want to know how many lines you've highlighted, you can pipe them to the `wc` (word count) command with the `l` flag:

```
pipe-to wc -l
```

The `pipe-line-to` command works in much the same way, but it only sends the current line, regardless of what is or isn't marked.

Once you've found the lines you're looking for, you may well want to extract them for future use. This could be as simple as keeping a copy of them to use in the future, or you might be writing a report about a particular event and need to document what's happened. Whatever the purpose, you can get data out of lnav in a couple of ways. One way is to extract the bookmarked lines to another text file using the `append-to <filename>` or `write-to <filename>` commands. Alternatively, you can output the results of an SQL query as either CSV or JSON with `write-to-csv <filename>` or `write-to-json <filename>`.

This article has been a whirlwind tour of the main features of lnav and should give you a valuable new tool for finding out what's going on deep in the heart of your Linux box. Hopefully you'll find that next time something goes wrong, it should be a little easier to figure out. ■■■

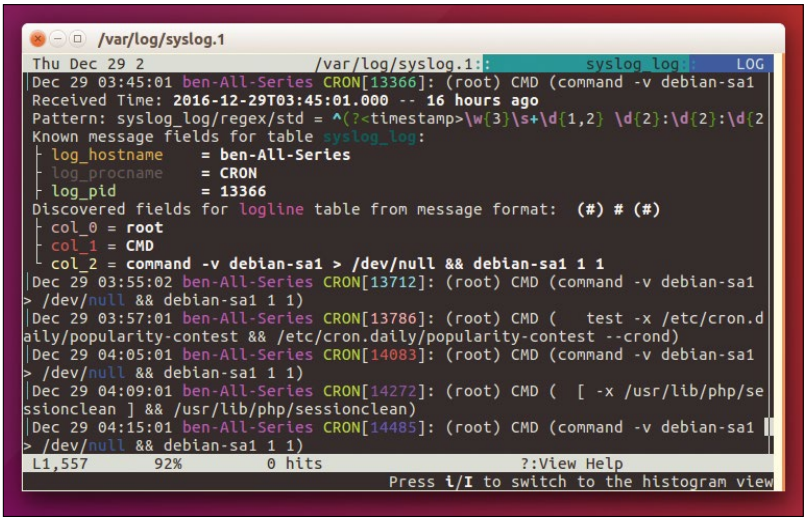


Figure 4: After pressing *P*, you can see the details lnav has managed to glean from the current log line.

Info

[1] "Regular Expressions" by Marco Fioretti, Linux Voice, issue 12, pg. 92, *https://www.linuxvoice.com/issues/012/regex12.pdf*

[2] Logfile navigator: *https://lnav.org/downloads/*

[3] Lnav documentation: *http://lnav.readthedocs.io/en/latest/*

Your very own Linux
Have It Your Way

Get a super-customized Linux installation by configuring and compiling the kernel with just the features you need.

BY MIKE SAUNDERS

Back in the day, rebuilding the kernel was something of a rite of passage for most Linux users. Typically, a Linux distribution would use a plain kernel that wasn't optimized for any specific CPU type and was bundled in various bits and bobs that users might need. Many other features and drivers – especially experimental ones – were left out, though. I remember having to recompile my kernel just to get audio working on my old Cyrix M3 box running Red Hat 5.1 back in the late 1990s.

Today, many desktop Linux distros include multiple kernel packages built for different CPU types, and almost every feature and driver is available as a module. Very few users actually need to build a custom kernel by hand, so why do it? Well, it's still a very useful technique to learn. Even the most bleeding-edge distros don't always enable every single feature in the kernel, and what if a new kernel is released with an important fix or update you need? You could wait a few weeks or months for your distro to package it up, but if you know how to compile it yourself, you can stay ahead of the game.

Some patches that you might want to try aren't part of the main Linux kernel source code tree, so you have to compile your own kernel to use them. Aside from all of the practical benefits, it's just fascinating to see what's going on inside the guts of a Linux installation and is a good little project to take on if you have a few spare hours on a weekend. Over the next few pages, I'll show you how to get, configure, compile, and install a fresh new kernel directly from Linus Torvalds' computer (well, thereabouts) and show you how to apply patches as well.

Before I start, though, it's worth reiterating exactly what the kernel does. Essentially, the kernel is the "core" process on a Linux installation: the first thing that is loaded and started (by the bootloader). The kernel is responsible for managing memory – making sure that programs don't overwrite one another's data – and sharing CPU time between different processes. Additionally, the kernel talks to your hardware via drivers and provides implementation of various protocols (e.g., networking) and filesystem drivers. Stability in the kernel is of utmost importance: Whereas a crash in Firefox, for example, might be slightly annoying and cause you to lose some work, a kernel crash can completely lock up your system.

Getting Set Up

Many Linux distros include source code packages for the kernels they ship, including extra patches and build scripts. In this tutorial, however, you're going to use the vanilla kernel code available from the main kernel website [1]. This site contains the compressed code archives signed off by Linus Torvalds and others, so it's the "pure" form of the kernel without any distro-specific patches or customizations.

On the kernel website you'll see a big yellow button with "latest stable kernel" – click it to grab the source code in `.tar.xz` format. (At the time of writing, this was version 4.9, but 4.10 or newer could be available by the time you read this.) This `.tar.xz` archive is around 90MB, so download it and save it into your home directory. Next, open a terminal, extract the archive, and switch into the resulting directory (using the version number of your download):

```
tar xfv linux-4.9.tar.xz
cd linux-4.9
```

After extraction, the code will take up around 780MB of disk space. That might sound huge, especially when you consider that a typical kernel is only a fraction of that! (Look in your `/boot` directory at files beginning with `vmlinuz` (Figure 1) – they are only around 7 or 8MB.) The vast majority of kernel source code is for drivers, most of which you'll never need, and are compiled into modules that are only loaded if your hardware requires them.

Now you are in your kernel source code directory, so enter `ls` to have a look around. You'll see subdirectories for various parts of the kernel: drivers, firmware, net(working), and so forth, along with a `kernel` subdirectory that contains the low-level workings of the kernel (see `cpu.c` for some CPU management routines and `kmod.c` for the module loader). It's all extremely technical stuff but worth having a look, if just to be wowed by the complexity.

Back in your `linux-4.9` directory, you'll need to make sure you have the right tools for compiling a kernel. Specifically, you need GNU Make, GCC (the C/C++ compiler), and the development packages for OpenSSL and ncurses (Figure 2). On a Debian-based distro, you can grab these with:

```
sudo apt-get install make gcc libssl-dev ➥
  libncurses5-dev
```

If you're running a distro based on Red Hat or Fedora (e.g., CentOS), try the

```
sudo dnf install make gcc openssl-devel ➥
  ncurses-devel
```

command.

Configuring the Kernel

Now the fun part begins. Before you compile the kernel, you have the opportunity to customize it to your liking. You can spend minutes, hours, or even days doing this – it all depends on what you want to achieve. If you're installing a new kernel for practical reasons, such as to get security updates and bug fixes, you might want to use the configuration for your currently running kernel and apply it to the new one. Have a look in your `/boot` directory, and you'll see one (or more) files beginning with `config-` followed by a version number (Figure 1).

Take a look inside one of these files; you'll see that they contain several thousand lines of options, followed by `y` or `m`. The `y` means that the option or feature should be compiled into the kernel image (i.e., into the `vmlinuz` file mentioned beforehand). If a feature is followed by `m`, it means it should be compiled as a module that is loaded only when it is needed (the sensible choice for the vast majority of kernel features).

If you want to use your existing kernel configuration with the new kernel, then, copy the appropriate `config-` file in `/boot` to `.config` inside the new kernel's source code directory. For instance, in this case, I'm in `linux-4.9/` in the home directory, and by running this command, I can see which kernel I'm currently running:

```
uname -r
```

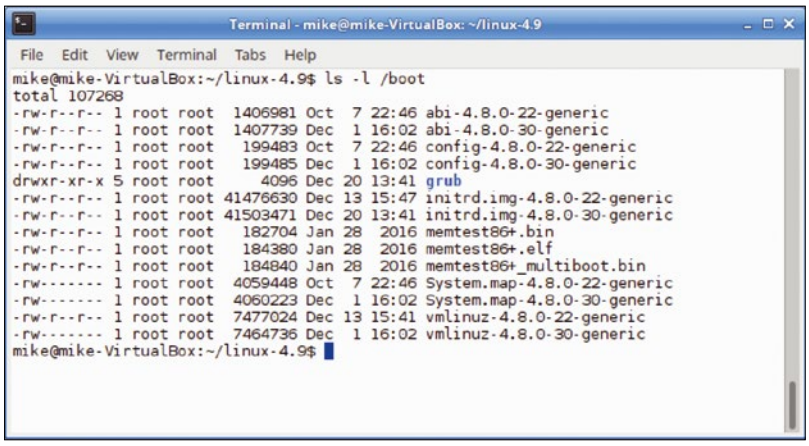


Figure 1: Have a peek inside your `/boot` directory, and you'll see `vmlinuz` files for kernels and `config` files for their build settings.

This tells me I'm running *4.8.0-30-generic*, so I copy the config file for that kernel into my source directory:

```
cp /boot/config-4.8.0-30-generic .config
```

If you don't want to make any changes to the configuration at this stage, you can now simply jump ahead to the "Building and Installing" section in this tutorial. If you do want to perform some fine-tuning, however, you have various options.

First, if you don't want to use the existing config file from your `/boot` directory but would rather generate a fresh (default) one, enter:

```
make defconfig
```

Now look inside `.config`, and you'll see a set of options that have been created for your CPU architecture. These may be OK, but chances are you still want to tweak them a bit. There's an interactive way to generate a `.config` file, with:

```
make config
```

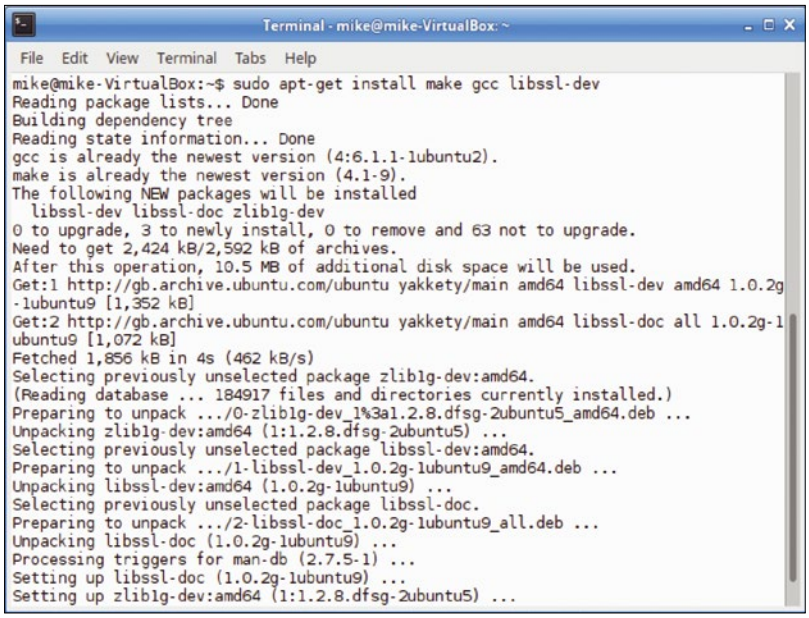


Figure 2: You'll need a few dependencies installed to build your kernel, namely GCC, GNU Make, and OpenSSL development headers.

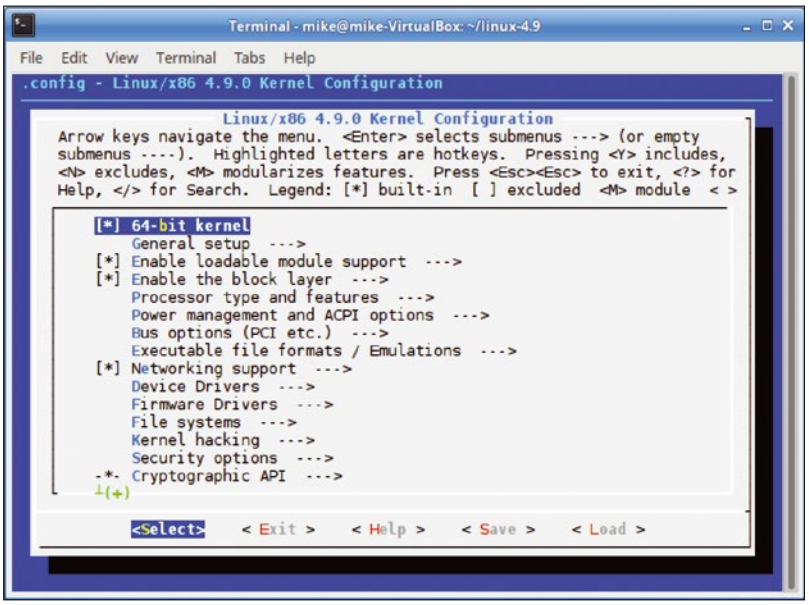


Figure 3: The `menuconfig` tool helps you customize your kernel setup in a simple menu-driven user interface.

The problem here is, you could spend several days answering questions! You have to answer yes/no/module to most of them, and with thousands of questions it becomes unwieldy. So the tool that almost everyone uses to configure kernels, including distro developers, is `menuconfig`:

```
make menuconfig
```

Now that's more like it. This provides a user-friendly menu-driven system (Figure 3) for exploring and enabling different features, drivers, and options in the kernel configuration. Use the Up and Down arrow keys to browse through menus and options, and hit the spacebar to select or disable them. Options listed with square brackets can only be built into the kernel; in other words, they cannot be loaded as modules. (This usually applies to very low-level features that need to be accessible right from the start of booting, before there's even a filesystem available from which to load modules.)

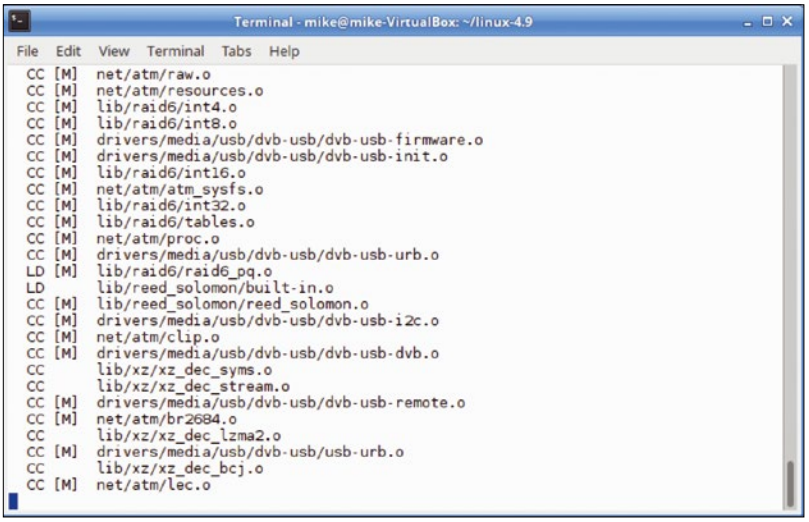


Figure 4: Compiling the kernel and modules can take anywhere from five minutes to five hours, depending on the specs of your machine.

If an option uses angle brackets (< >), however, it can be enabled as a module; just keep hitting the spacebar until *M* appears. You'll notice that most drivers are available as modules, and it doesn't make much sense to compile them directly into your kernel image unless you're building an embedded system with extremely restricted requirements. Most features and options have accompanying help text, so to access it, use the Right and Left arrow keys to select *Help* at the bottom of the window and then hit Enter. To go back to a higher level menu, choose `Exit`.

The big question now is: What options should you enable? If this is your first time building a kernel, I highly recommend copying an existing config file from your `/boot` directory into `.config` before running `make menuconfig`, as described earlier. This means your new kernel will be set up in much the same way as your existing kernel, so you shouldn't have problems booting later on.

From here, it's all up to you. Try exploring the menu options *General setup*, *Processor type and features*, *Kernel hacking* (if you're a developer), and *Security options*. Even if a lot of the features are unfamiliar to you, it really highlights just what a flexible and feature-rich kernel Linux is. No surprise then that it's being used on everything from wristwatches to mainframes.

When you're finished, choose *Exit* at the bottom and choose to save your configuration.

Building and Installing

With your dependencies installed and the kernel configured to your liking, now comes the big moment: compiling it. This step used to be a more involved procedure with several commands, but these days, it all comes down to a single:

```
make
```

A better approach, if you have a multicore CPU (like pretty much every PC built in the last five years), is to run a parallel `make`, specifying the number of cores you have. For instance, on a dual-core machine, use:

```
make -j2
```

This option speeds things up considerably. How long the whole build process takes, though, depends entirely on the speed of your machine: If you're rocking the latest CPU and building on all four cores, you could be done in a matter of minutes. Older boxes and low-spec machines like the Raspberry Pi can take a few hours. You can watch the build as it progresses (Figure 4): *CC* in the output refers to the C compiler doing its work, whereas *LD* is the linker, which essentially links together the executable binary code generated from

each compile so that you end up with a single kernel image. Additionally, *[M]* indicates that something is being compiled as a module.

Once the build process has finished and you land back at the command prompt, you need to install the newly built kernel and its modules into your filesystem. (If you're on RHEL, Fedora, or CentOS, make sure that you have the *grubby* package installed at this point). Enter:

```
sudo make modules_install install
```

When this command has finished, have a look in `/boot` and you should see your new kernel (e.g., `vmlinuz-4.9.0`). Reboot your distro, and bring up the GRUB menu – if it doesn't appear by default, keep tapping Esc just after the BIOS screen appears. You should see a list of kernels, including your shiny new one, which you can now try out (Figure 5). And if anything's wrong with it, simply reboot and choose the older one. Enjoy.

If you want to add more custom and experimental features to your kernel, see the "Patching Your Kernel" box, and if you want the latest and greatest, see the "Living on the Bleeding Edge" box. ■■■

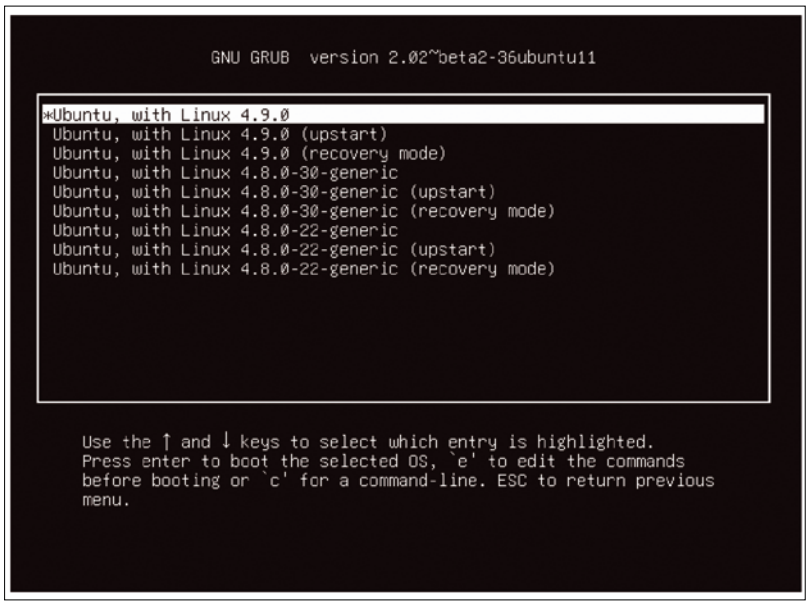


Figure 5: In Ubuntu-based distros, choose ***Advanced options*** in the GRUB boot menu to access and boot all available kernels.

Info

[1] Linux kernel archives: *www.kernel.org*

[2] RT patch: *https://www.kernel.org/pub/linux/kernel/projects/rt/*

Patching Your Kernel

In some cases, you might want to modify your kernel with third-party code (i.e., code that's not included in the stock kernel itself) before building it. Many extra features and customizations are available as "patches" – text files that modify the source code. You "apply" a patch to your kernel code to get the new features or changes, but bear in mind that you're then using possibly experimental code that the core kernel developers haven't signed off on. If you patch your kernel and it crashes, you won't get far reporting the problem on the Linux kernel mailing list; they'll tell you to test with a vanilla kernel from kernel.org first.

Here, I'll look at how to apply a patch. One of the most popular patches is the real-time kernel patch, which, as the name suggests, rewires the Linux kernel to be more suitable for real-time work (i.e., tasks that must be completed at very specific times, rather than just handled when the scheduler has some free time). The patch is available online [2], so go into the appropriate directory for the kernel you're building, grab the `.patch.gz` file (e.g., `patch-4.9-rt1.patch.gz` for the example in this article), and save it in your home directory.

In a terminal window, switch into your kernel source code directory (e.g., `linux-4.9/`) and run:

```
zcat ../patch-4.9-rt1.patch.gz | patch -p1 --dry-run
```

This command does two things: It decompresses the downloaded patch file and sends the results to the `patch` command. The `--dry-run` option tells `patch` simply to test the code to see whether it can be updated correctly, without actually changing the contents of your kernel source code. If everything looks OK, rerun the command, this time omitting the `--dry-run` part. Now your kernel has been modified, and you can continue to build it with `make` as in the main text.

Try having a look inside the patch,

```
zless patch-4.9-rt1.patch.gz
```

and you'll see how it works: Lines that should be removed from the kernel code are prefixed with dashes, and lines to be added have pluses. Some changes are very minor, whereas others involve big new chunks of code.

Living on the Bleeding Edge

At the start of this tutorial, I mentioned getting the code fresh from Linus' computer, and if you really want the absolutely latest code from his repository, you can get it via the Git source control system (which, funnily enough, was created by Linus Torvalds as well!). Install Git via your distro's package manager and then enter this in a terminal prompt:

```
git clone git://git.kernel.org/pub/scm/linux/kernel/git/
  torvalds/linux.git
cd linux
```

Git will "pull" or download the very latest code into the `linux` directory, which you then switch into before continuing to follow the process in the main text of this article. Note that this is bleeding edge code, though; it will probably be usable, but there's no guarantee that everything will work perfectly. You could experience crashes or even data loss, depending on what state some features are in – so don't sue us (or Linus!) if something goes wrong. On production machines, it's always best to use official, final releases from the main kernel website.

FEATURED EVENTS

Users, developers, and vendors meet at Linux events around the world. We at *Linux Magazine* are proud to sponsor the Featured Events shown here.

For other events near you, check our extensive events calendar online at *http://linux-magazine.com/events.*

If you know of another Linux event you would like us to add to our calendar, please send a message with all the details to *events@linux-magazine.com*.

2017 HPC for Wall Street

Date: April 3, 2017

Location: New York City, New York

Website: *http://www.flaggmgmt.com/linux/*

Register now for HPC for Wall Street for an All-Star Conference Program! This event is designed for forward-thinking industry veterans to get the latest on cloud and data centers. Visit the website to learn more and register today.

DrupalCon Baltimore

Date: April 24–28, 2017

Location: Baltimore, Maryland

Website: *https://events.drupal.org/baltimore2017*

The Drupal community is one of the largest open source communities in the world. We're developers, designers, strategists, coordinators, editors, translators, and more. Once a year, we comes together in a US city for one of our biggest events: DrupalCon. This year we bring DrupalCon to Baltimore. Visit our website to learn more!

OSDC Berlin

Date: May 16–18, 2017

Location: Berlin, Germany

Website: *https://www.netways.de/events/osdc/overview/*

OSDC offers the opportunity to meet open source professionals and insiders and gather and share expertise over three days of presentations, hands-on workshops, and social networking. OSDC aims to "simplify complex IT infrastructures with open source" for experienced administrators and architects.

EVENTS

Chemnitzer Linux-Tage 2017	March 11–12	Chemnitz, Germany	https://chemnitzer.linux-tage.de/2017/en
CeBIT 2017	March 20–24	Hanover, Germany	http://www.cebit.de/home
World Hosting Days Global	March 25–31	Rust, Germany	http://worldhostingdays.com/global/
SPTechCon 2017	April 2–5	Austin, Texas	http://www.sptechcon.com/
2017 HPC for Wall Street – Cloud and Data Centers Show and Conference	April 3	New York, New York	http://www.flaggmgmt.com/linux/
JAX DevOps	April 3–6	London, United Kingdom	https://devops.jaxlondon.com/
DrupalCon Baltimore	April 24–28	Baltimore, Maryland	https://events.drupal.org/baltimore2017
Grazer Linux-Tage 2017	April 28–29	Graz, Austria	https://www.linuxtage.at/
Check_MK Conference #3	May 2–4	Munich, Germany	http://mathias-kettner.de/
Linux Presentation Day 2017.1	May 6	Europe-wide in numerous cities	https://linuxday.ch/index.php/en/
Open Source Data Center Conference I OSDC 2017	May 16–18	Berlin, Germany	https://www.netways.de/events/osdc/overview/
ISC High Performance (ISC 2017)	June 18–22	Frankfurt, Germany	http://www.isc-hpc.com/
AnDevCon	July 17–19	Washington, DC	http://www.andevcon.com/
InterDrone	September 6–8	Las Vegas, Nevada	http://www.interdrone.com/

CALL FOR PAPERS

We are always looking for good articles on Linux and the tools of the Linux environment. Although we will consider any topic, the following themes are of special interest:

- System administration
- Useful tips and tools
- Security, both news and techniques
- Product reviews, especially from real-world experience
- Community news and projects

If you have an idea, send a proposal with an outline, an estimate of the length, a description of your background, and contact information to *edit@linux-magazine.com*.

The technical level of the article should be consistent with what you normally read in *Linux Magazine*. Remember that *Linux Magazine* is read in many countries, and your article may be translated into one of our sister publications. Therefore, it is best to avoid using slang and idioms that might not be understood by all readers.

Be careful when referring to dates or events in the future. Many weeks could pass between your manuscript submission and the final copy reaching the reader's hands. When submitting proposals or manuscripts, please use a subject line in your email message that helps us identify your message as an article proposal. Screenshots and other supporting materials are always welcome.

Additional information is available at:
http://www.linux-magazine.com/contact/write_for_us.

AUTHORS

CONTACT INFO

Editor in Chief
Joe Casad, jcasad@linux-magazine.com

Managing Editor
Rita L Sooby, rsooby@linux-magazine.com

Localization & Translation
Ian Travis

News Editor
Swapnil Bhartiya

Copy Editors
Amber Ankerholz, Amy Pettle

Layout
Dena Friesen, Lori White

Cover Design
Lori White

Cover Image
© Oleksandr Omelchenko, 123RF.com

Advertising – North America
Ann Jesse, ajesse@linuxnewmedia.com
phone +1 785 841 8834

Advertising – Europe
Brian Osborn, bosborn@linuxnewmedia.com
phone +49 89 99 34 11 48

Publisher
Brian Osborn, bosborn@linuxnewmedia.com

Marketing Communications
Gwen Clark, gclark@linuxnewmedia.com
Linux New Media USA, LLC
616 Kentucky St.
Lawrence, KS 66044 USA

Customer Service / Subscription
For USA and Canada:
Email: cs@linuxpromagazine.com
Phone: 1-866-247-2802
(Toll Free from the US and Canada)
Fax: 1-785-856-3084

For all other countries:
Email: subs@linux-magazine.com
Phone: +49 89 99 34 11 67
Fax: +49 89 99 34 11 98

www.linuxpromagazine.com – North America

www.linux-magazine.com – Worldwide

Printed in Germany

Distributed by COMAG Specialist, Tavistock Road, West Drayton, Middlesex, UB7 7QE, United Kingdom

LINUX PRO MAGAZINE (ISSN 1752-9050) is published monthly by Linux New Media USA, LLC, 616 Kentucky St., Lawrence, KS, 66044, USA. Periodicals Postage paid at Lawrence, KS and additional mailing offices. Ride-Along Enclosed. POSTMASTER: Please send address changes to Linux Pro Magazine, 616 Kentucky St., Lawrence, KS 66044, USA.

Published monthly in Europe as Linux Magazine (ISSN 1471-5678) by: Sparkhaus Media GmbH, Zieblandstr. 1, 80799 Munich, Germany.

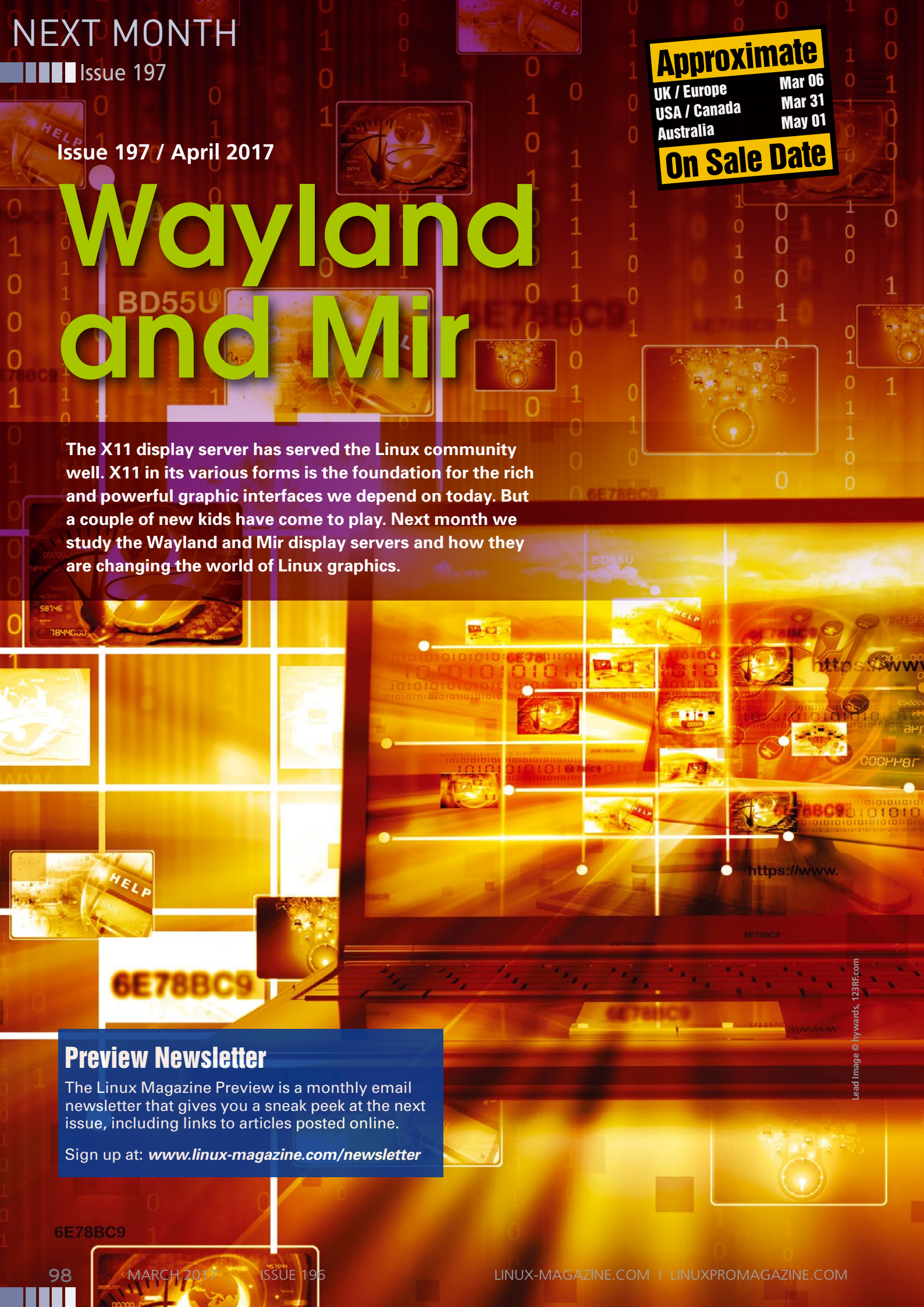
Issue 197 / April 2017

Wayland and Mir

Approximate On Sale Date

UK / Europe	Mar 06
USA / Canada	Mar 31
Australia	May 01

The X11 display server has served the Linux community well. X11 in its various forms is the foundation for the rich and powerful graphic interfaces we depend on today. But a couple of new kids have come to play. Next month we study the Wayland and Mir display servers and how they are changing the world of Linux graphics.



Preview Newsletter

The Linux Magazine Preview is a monthly email newsletter that gives you a sneak peek at the next issue, including links to articles posted online.

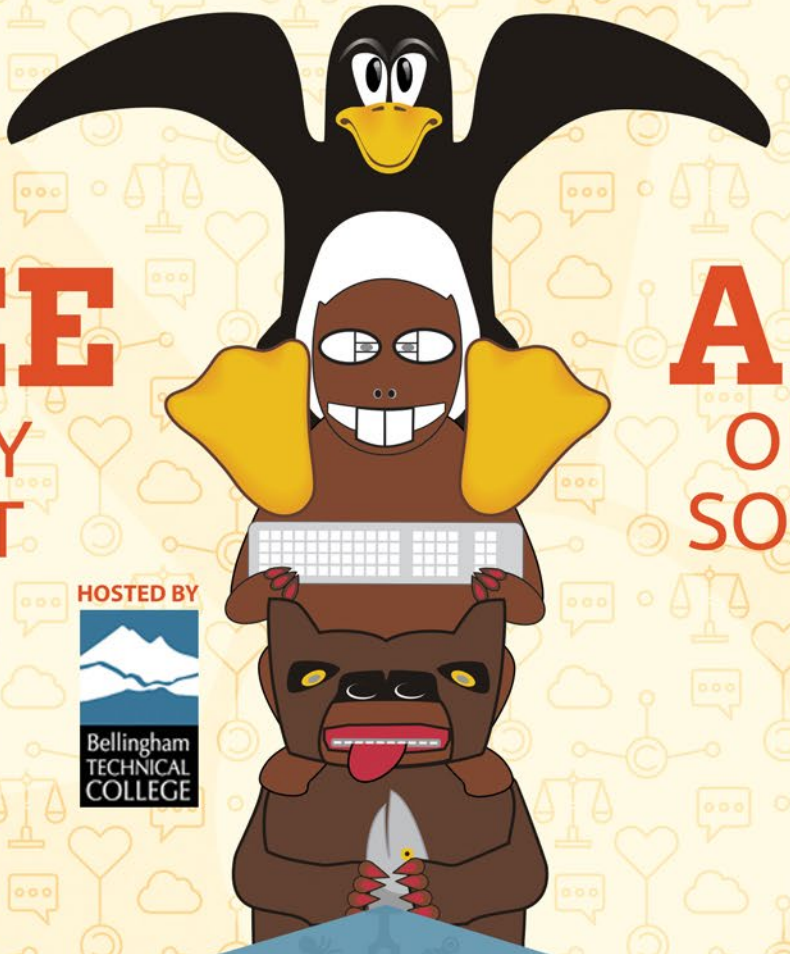
Sign up at: ***www.linux-magazine.com/newsletter***

LINUXFEST NORTHWEST

2017

MAY 6TH & 7TH

BELLINGHAM, WA



FREE
FAMILY EVENT

ALL
OPEN SOURCE

HOSTED BY
Bellingham TECHNICAL COLLEGE

1500+
Attendees

40+
Exhibitors

80+
Sessions

LINUXFESTNORTHWEST.ORG

MicroBlade

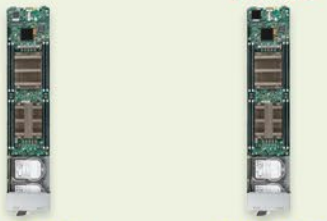
High Density • High Performance • High Efficiency • Cost-Effective
Enterprise, Data Center, Web Applications, HPC and Cloud Computing Solutions

Intel® Xeon® Processor E5-2600 v4/v3 Product Families Supported

3U MicroBlade
*Up to **28 UP/ 14 DP** Nodes*

6U MicroBlade
*Up to **112 UP / 28 DP** Nodes*

10Gbps



MBI-6128R-T2 MBI-6128R-T2X

Dual Intel® Xeon® Processor E5-2600 v4/v3

New! ***HD Graphics*** New!
NVMe
10Gbps

MBI-6219G-T7LX MBI-6119G-T7LX

Intel® Xeon® Processor E3-1578L v5

New! ***10Gbps*** New!

MBI-6218G-T81X
MBI-6218G-T41X
MBI-6118G-T81X
MBI-6118G-T41X

Intel® Xeon® Processor D

New! ***SAS3*** New! New! New!

MBI-6119G-C2 MBI-6119G-C4 MBI-6119G-T4 MBI-6219G-T

Intel® Xeon® Processor E3-1200 v5

MBI-6118D-T2/T2H MBI-6118D-T4/T4H

Intel® Xeon® Processor E3-1200 v4/v3

MBI-6418A-T7H MBI-6418A-T5H

Intel® Atom™ Processor

Intel Inside®.
Powerful Productivity
Outside.

TITANIUM We Power the Cloud X11 SERVER PLATFORMS nvm EXPRESS

Learn more at www.supermicro.com