



### Contents

How and why the FSF drives free software development . . . . .	1
Why we still need GnuPG. . . . .	3
Nonfree digital drugs are a bad idea . . . . .	4
Escaping the JavaScript trap with LibreJS . . . . .	6
On the road with RMS. . . . .	8
How the tech team has been supporting you and GNU . . . . .	10

### How and why the FSF drives free software development

*By John Sullivan  
Executive Director*

While much of the public attention the FSF receives focuses on our activist campaigns and our advocacy work, associate members who join the FSF as a way to "give back" for all the free software they use should feel confident that their support will indeed help produce more of that software. Here are three ways in which the FSF drives free software development.

First, we directly fund development, from our general budget – especially programs we need for our own operations, or

programs connected to one of our current campaigns. Most recently, we funded GNU LibreJS, which meets both of these criteria (see page 6). We also serve as a fiscal sponsor to several projects, allowing people to fund their development. Because of the trust donors have in the FSF's integrity, affiliation with the FSF means that projects are potentially able to attract significant amounts of funding. This fall, for example, a group called Handshake gave a total of \$500,000 to multiple projects sponsored by the FSF.

Second, the FSF provides direction to guide overall free software development. Working



*The FSF is everywhere: Gertrude the Gnu accompanied Campaigns Manager Molly de Blanc to a free software conference in Cuba.*


with a committee of experts who solicit extensive feedback from the public, we maintain a High Priority Projects list highlighting areas that need more investment of resources and time ([u.fsf.org/hpp](http://u.fsf.org/hpp)). We recruit volunteers to work in these and other areas, including placing interns who are able to spend a few months helping a project while learning skills that will lead them to a lifetime of promoting and creating free software. Furthermore, we set high standards under our Respects Your Freedom product certification program that motivate development ([fsf.org/ryf](http://fsf.org/ryf)). These standards inspired Libreboot, the fully free boot firmware project, and they continue to encourage developers to solve difficult problems.

Third, we provide infrastructure used by thousands of contributors to GNU and other free software projects, including mailing list servers; a shell server; build machines; wikis; software distribution; bug trackers; Web servers; virtual machines; and more. Not only do we provide all of this gratis, but we run it using free software on hardware we maintain, enabling developers to stay true to their principles (see page 10). We also provide legal infrastructure, managing copyright assignments for many GNU packages, and enforcing the GNU General Public License to protect the free software commons.

In our 2017 fiscal year, we spent over \$300,000 supporting free software development in these ways. In our new fiscal year, thanks to Handshake, as well as a \$1 million donation from the

Pineapple Fund, we are poised to dedicate even more resources to it. We make this a priority because our mission is to enable users everywhere to live fully free digital lives. To do that, they must have free software that does all of the things they want and need to do.

When free software does an excellent job meeting users' needs, it helps prove proprietary software companies wrong, demonstrating that we actually can have fully free computing. Using a free program is also often the way people first get interested in learning more about how unethically the software industry typically treats them. Supporting free software development is a double win: it leads to programs we need while also boosting our advocacy and campaigns work.

While the FSF doesn't currently employ anyone full-time to develop software, we inspire and drive development in these impactful ways. We're fortunate that donors are increasing their commitments in this area; given the immensity of the task, it's imperative that we turn those resources into successes that will attract even more such commitments. Our long-standing reputation and skilled staff allow us to provide the technical, communications, and fundraising infrastructure enabling hackers to focus on their area of expertise: the code. You can rest assured that your donations and memberships are an effective way to give back and support future much-needed free software development. 

# Why we still need GnuPG

*By Andrew Engelbrecht*

*Web Developer*

The use of GNU Privacy Guard (GPG) encrypted emails is important for political dissidents, journalists, whistleblowers, and those who need to protect the privacy of their messages. GPG is an essential tool for securely encrypting and signing communications, in order to mitigate surveillance and impersonation. For some people, their very life, and the lives of those they love are at stake, so ensuring that their communications are secure is critical. Even for those of us who do not have this level of need, we should still aim to not simply hand over our private information to whichever surveillance states and email service providers happen to be recording our communications. In addition, ordinary everyday use of GPG helps to provide cover for others who need it the most. The FSF created the Email Self-Defense Guide at [emailselfdefense.fsf.org](http://emailselfdefense.fsf.org) to expand the practice of email encryption.

In May 2018, [efail.de](http://efail.de) announced a vulnerability, dubbed EFAIL, for some email clients that use GPG or S/MIME to decrypt encrypted messages. S/MIME is an encryption scheme that relies upon a certificate authority instead of peer-to-peer key signing. In the case of GPG encrypted emails, the vulnerability was not in GPG

itself, but rather in the way that some email clients decrypt messages that would allow attackers to embed such messages in the context of maliciously crafted HTML code. After decryption, loading HTML elements or clicking on links could then exfiltrate that data to remote servers. Some users could work around this problem by disabling HTML rendering of emails, because without loading HTML elements, the data could not be exfiltrated. However, for S/MIME users, disabling HTML rendering alone was not entirely secure. Some GPG users remained at risk, because Apple's proprietary email client doesn't allow disabling HTML rendering of emails.

The Electronic Frontier Foundation (EFF) published an article about the vulnerability, drawing attention to the problem, and controversially recommended that people stop encrypting and decrypting emails within their email clients until the issue was resolved. Since that time, all known attack vectors have been patched and resolved in Icedove and Thunderbird, as well as Enigmail, which is a GPG encryption and decryption plugin for those email clients. As far as EFAIL is concerned, using GPG with Enigmail should be safe, as long as you and the people you are corresponding with are using the latest version of the plugin and email client. Even if there are more unknown attack vectors, we still believe that people should use


GPG to protect their email and to provide coverage for those who depend on it the most, with the caveat that the right answers here may differ for people who believe they are or may be individually targeted by well-resourced surveillance.

For added safety, you should still disable HTML rendering in your email client. Disabling HTML rendering of emails should reduce the attack surface of your email client if future vulnerabilities are discovered. While disabling the automatic loading of external HTML assets, another common feature, will help to protect your IP address and other information, it is not sufficient to mitigate EFAIL alone, because loading assets manually remains possible, and is therefore insecure.

Another preventative measure that is not strictly necessary, if you are using the latest version of Enigmail and your email client, is to copy and paste links into a text editor and to eyeball them before following those links. If you use another email client, you should check the project's Web site, or contact one of its maintainers, to see how well it's mitigating known attacks against GPG encrypted email.

The EFAIL vulnerability is just another demonstration of the benefits of free software: GPG and Enigmail are free software, and their source code is available to be audited by anyone, so researchers can find security holes and fix them. Researchers who find issues

and write patches may share them freely with everyone, and those patches may also be audited by anyone who knows how. On the other hand, if you use proprietary software, like Apple's bundled email client, you may have to deal with antifeatures that compromise your security when opening encrypted email, and you are left unable to write and share patches to that proprietary software.

This attack requires that we reach out to our friends to let them know how to secure their email communications, because both the past senders and recipients of encrypted emails are potential targets of this attack. If you are new to GPG email encryption, using GPG to encrypt your emails is quite easy, and [emailselfdefense.fsf.org](http://emailselfdefense.fsf.org) contains a step-by-step guide to get you started. 

## Nonfree digital drugs are a bad idea

*By Molly de Blanc*

*Campaigns Manager*

**A**ripiprazole, also known as Abilify, is an antipsychotic drug used in the treatment of schizophrenia and bipolar disorder. It's also a successful conjunctive (or add-on) therapy for major depressive disorder, and has been known to help bipolar individuals during major depressive episodes. For some people, Abilify may be a lifesaving treatment.

Abilify is also the first pill approved by the United States

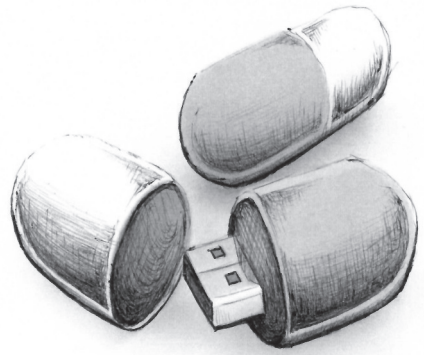
Food and Drug Administration (FDA) to be available in both digital and non-digital forms ([u.fsf.org/fb1](http://u.fsf.org/fb1)). A digital pill is a pharmaceutical drug containing an ingestible sensor, which transmits medical data.

Digital pills were designed to solve a problem: people are bad at taking drugs. Whether it's lifesaving treatments for conditions like HIV and tuberculosis, or antipsychotics necessary for the treatment of schizophrenia or bipolar disorder, or even oral contraceptives, treatments fail when pills are not taken at the necessary times. However, we need to look beyond the potential medical benefits and consider how digital pills relate to user freedom, including the ways they leave users vulnerable to invasions of privacy.

When someone takes Abilify MyCite (the digital form of Abilify), a sensor is activated by stomach acid. It then transmits a signal to a patch worn on the patient's skin. This, in turn, communicates to a smartphone app. (This is a good opportunity to remind ourselves that smartphones are really just pocket computers that can also make phone calls.)

The obvious danger here is that people with highly stigmatized mental health conditions are being called to trust some of their most personal data to a smartphone app.

It seems almost redundant to emphasize the risks posed by proprietary smartphone apps. In the Spring 2018 issue of the *Bulletin*, I wrote about Uber,



covering ways the company has taken advantage of the opacity of its software ([u.fsf.org/fb2](http://u.fsf.org/fb2)). Security leaks and vulnerabilities run rampant across every kind of computing device, and data is stored in ways that range from vulnerable to downright dangerous. Users need to be in control of what is being transmitted from their devices, rather than companies in control of what is being received.

When some of our most precious data is being entrusted to proprietary software, we are unable to trust it and should not feel confident with its security. Privacy, security, and surveillance are major free software issues. Software freedom is a necessary first step to accountability in computing, which enables user consent and user freedom. We cannot have accountability in computing until all of the software we use is free. Without the ability to examine code for vulnerabilities, and without the opportunity to see exactly what it is doing and how it is doing it, we cannot be sure how safe a given technology actually is, and therefore we cannot, in good

conscience, consent to its use.

Knowing this, we can see how digital pills could go terribly wrong for users. Imagine a malicious party infiltrating a digital drug app to misreport the user's adherence to their daily regimen. Imagine a smartphone app being able to list and track every person who is prescribed a certain drug or class of drugs. Imagine how easy it would be to use the smartphone app to build lists of everyone with mental health concerns, and then to track not just what they're taking, but how much, and when. In societies where mental health issues are highly stigmatized, what is the value of this sort of medical information?

People with mental health conditions benefit from their prescriptions.

Medication nonadherence is expensive and dangerous, and I appreciate that those who invented digital pills want to help people take care of their health ([u.fsf.org/fb4](http://u.fsf.org/fb4)). But we cannot simply and blindly accept digital medicine without first taking a serious look at the implications the technology has. This technology needs to be in the hands of the users of these pills, and the only way for that to be possible is when the software behind it is free. 🐾

## Escaping the JavaScript trap with LibreJS

*By Ian Kelling*

*Senior Systems Administrator*

Most Web pages contain nonfree JavaScript programs that, like other nonfree programs, deny you freedom. The primary functionality of many sites won't work without running JavaScript. This includes almost all online shopping.

Why is JavaScript so prevalent? Without it, Web sites are written solely in HTML and similar markup languages that intentionally lack the features of a programming language, and are limited to a specific set of behaviors. JavaScript can accomplish much more, and is often used to enhance HTML. Because JavaScript is so easy to download and works almost the same on every browser, it has become one of the most popular programming languages, and many complex programs have been created with it, including Google Docs.

The dark side of JavaScript, however, includes many of the standard abuses that can come with nonfree software: for instance, one common function of nonfree JavaScript is to record your actions while looking at a Web page ([u.fsf.org/fb5](http://u.fsf.org/fb5)), often used to identify and profile you. An ethical Web site does not know who you are until you tell them. Identifying and profiling users without their

consent is a huge violation of your privacy, but it is unfortunately very common.

The program GNU LibreJS detects nonfree JavaScript in pages you visit and blocks it, preventing it from running and thus saving you from giving up your freedom ([u.fsf.org/fb6](http://u.fsf.org/fb6)). LibreJS is included in the GNU IceCat browser, and is available as an add-on for Firefox and Abrowser.

Blocking JavaScript is already more common than you think – adblockers often block JavaScript programs coming from domains that serve ads, since these programs are usually malware or unnecessary. But just blocking ad-related JavaScript is not good enough. We need the freedom to run, edit, contribute to, and share the software we use. You can find out more about our Free JavaScript campaign at [u.fsf.org/freejs](http://u.fsf.org/freejs).

This issue desperately needs more action. Here are some things you can do:

- Install and use LibreJS.
- Ask Web site owners to stop including nonfree JavaScript. In most cases, a Web site should not require running any JavaScript at all in order to use it, as the functionality of the Web site does not justify requiring you to run a program. Any JavaScript that is included should be made free and validated with LibreJS.
- When someone recommends you go to a site with nonfree

JavaScript, make sure they know about this issue.

- Make free replacement programs for nonfree JavaScript. Sometimes this includes reverse engineering the existing JavaScript so the replacement will work with a specific Web site. IceCat includes some extensions that are free JavaScript replacements for specific Web sites. The FSF, with the help of an intern, Alyssa Rosenzweig, is close to recommending a replacement for nonfree JavaScript for making payments with PayPal called Pagamigo ([u.fsf.org/fb8](http://u.fsf.org/fb8)).
- Improve JavaScript tools so they facilitate creating LibreJS-validated sites. For instance, JavaScript tools which compact source code should be improved so they automatically provide the location of the source code to LibreJS.
- General JavaScript development learning resources also need contributions to include teaching developers how to make their code free for users.

Firefox 57, released in November 2017, switched to a new API for extensions called WebExtensions. For LibreJS to work with this new API, it required an almost complete rewrite. As I write this, LibreJS 7.18.0 was recently released. It has major improvements and is even better than before the rewrite, including a more intuitive interface and much faster source code checking.

Our contractor, Giorgio Maone, the author of NoScript, has been helping us with a lot of improvements to LibreJS. The new versions are compatible with modern Mozilla-based browsers that use the WebExtensions API, including the recently released GNU IceCat 60 ([u.fsf.org/fb11](http://u.fsf.org/fb11)). LibreJS is now much faster and more robust, and we are working on further improvements to make licensing-tagging easier for Web developers.

LibreJS has one core developer and is improving quickly, but it could use help with further development, including features such as internationalization of the user interface, support for running on Android and Chromium, and better documentation, more testing, and debugging. If you want to help with these tasks, please subscribe and write to the mailing list at [u.fsf.org/fb9](http://u.fsf.org/fb9). 🐣

## On the road with RMS

*By Jeanne Rasata*

*Assistant to the President*

The past six months have been characteristically busy for FSF president Richard Stallman (RMS). He gave thirty speeches, spoke at eight conferences, keynoted one conference, and took part in two panels. A little over half of his talks were in Spanish, a couple in French, and the rest were in

English. His travels took him to twenty-six different cities, across ten countries.

In April and May, he participated in two events in Canada. In Ottawa, Ontario, he took part in a public discussion with Joseph Potvin, executive director of the Xalgorithms Foundation, about a free software solution to the debacle of the Phoenix pay system, the Canadian government's payroll processing system, which relies on nonfree software. Phoenix was rolled out in 2016, and has affected payment to over 200,000 government employees, and cost the Canadian government over \$1.2 billion. Potvin set the ambitious goal of providing a free software solution within a year, setting an excellent example for why governments should use free software exclusively.

RMS then headed to the University of Quebec in Montreal (UQAM) for the fifth colloquium of the Adte, a nonprofit that promotes free/libre educational resources in higher education. There, he delivered the keynote speech, "Education and freedom."

From mid-May to early June, he was in South America. In Brazil he spoke at Campus Party Bahía, in Salvador de Bahía, at the Instituto Federal de São Paulo, in Araraquara, on the Pato Branco campus of the Universidade Tecnológica Federal do Paraná, and at the Universidade de São Paulo.

RMS also toured Argentina, starting in Misiones Posadas, speaking to, among others, elected

officials from the province. He then spoke at the Universidad de Tucumán, which awarded him an honorary doctorate, and at the Universidad Tecnológica Nacional in Mendoza, whose graduates make up half of Argentina's engineering-degree holders, and which awarded him an honorary professorship, its highest honor. Next, he headed to Río Cuarto, Córdoba, before heading to the Autonomous City of Buenos Aires, where he gave two speeches: "Free software: Sovereignty and technological independence," at the Instituto Universitario de la Policía Federal, and "Copyright vs community," at the Centro Cultural de la Cooperación Floreal Gorini, for gcoop, a free software cooperative. He ended his visit to Argentina with a speech at the Universidad Nacional de La Plata, and another in Mar del Plata.

After a brief respite in Boston, RMS headed to Barcelona, Spain, for two events. At the first, Maker Faire, he had a public conversation with Francesca Bria, the Barcelona City Council chief of technology, titled "Digital cities, digital freedom, digital privacy." Next, he delivered a speech at the Universitat Politècnica de Catalunya. During his visit to Spain, he also spoke remotely for the seventh No-Spy Konferenz, which was taking place in Stuttgart, Germany.

He then both gave a speech and sat on a panel titled "Smart city, spy city? Avenues for making a city 'smart' while respecting



*In Montreal, RMS spoke and met with supporters at the University of Quebec for the fifth colloquium of the Adte.*

privacy and anonymity," at the Next Generation Cities conference, in Amsterdam, the Netherlands, before returning to Spain to speak at the Universidad de Jaén.

Back in the United States, in New York City, RMS took part in the the eleventh HOPE (Hackers On Planet Earth) conference, driving home the point that "we must legislate to block collection of personal data."

In August, he headed to Chile for seven speeches: in Santiago de Chile, at both the Universidad de Chile and at the Universidad Tecnológica de Chile INACAP; in Rancagua; in Valdivia, both privately, for regional authorities, and publicly, at the Centro de Estudios Científicos; at the Universidad Católica de Valparaíso, in Valparaíso; and at the Universidad Técnica Federico Santa María Sede Viña del Mar, in Quilpué.

RMS ended the summer in Brussels, Belgium, where he took part in the first edition of EduCode, giving his speech "Control your computer, so as not to be controlled!"

So far this fall, he's spoken at the Istituto Italiano di Tecnologia, in Genoa, Italy, and at the Illinois Institute of Technology, in Chicago, Illinois, and sat on the "Bits, bots, and our biome: Technology and tech power" panel at the Bioneers Conference, in San Rafael, California.

See [u.fsf.org/zi](http://u.fsf.org/zi) for a list of RMS's confirmed engagements. Please write to [rms-assist@gnu.org](mailto:rms-assist@gnu.org) to extend him a speaking invitation. 🐼

## How the tech team has been supporting you and GNU

*By Ruben Rodriguez*

*Senior Systems Administrator*

The tech team at the FSF is in charge of maintaining and extending a large set of

infrastructure that includes the systems at the Foundation's office as well as over a hundred virtual machines on a handful of servers at three different data centers. Our machines host the GNU Project, FSF campaigns, and services for the community. This gets us hundreds of task requests every month, and this year we met our goal by resolving over 2,800 tasks and keeping the number of pending tasks under 150.

We made good progress on the GNUHope cluster project, a modern server stack that is replacing our main infrastructure. Since its deployment at the beginning of the year, we have now increased its storage capacity, improved the network design, and hardened the firewall. Thanks to that gained capacity, we were able to decommission the biggest server of the old stack, Pyxis, a venerable machine which had been hosting some critical infrastructure – including [gnu.org](http://gnu.org) and [fsf.org](http://fsf.org) – and that had started to show its age by overheating and crashing. We also had some hardware and network incidents that caused a bit of downtime, but they were resolved without any data loss.

We also launched an improved forum for FSF associate members at [forum.members.fsf.org](http://forum.members.fsf.org), powered by free software called Discourse ([u.fsf.org/fb10](http://u.fsf.org/fb10)). This is a dedicated space where members can meet, communicate, and collaborate with each other. You'll be able to log in using the same Central Authentication

**Get 10% off!**

**Support the Free Software Foundation by purchasing GNU Press merchandise!**

**Visit our [shop.fsf.org](http://shop.fsf.org) with new stuff coming soon!**

**Enter discount code  
FALL2018.**


Service (CAS) account that you used to set up your membership. We encourage you to log in, check it out, and get the conversation going.

Over the spring and summer we welcomed a great group of collaborators, including four talented interns and an independent contractor (see page 6 for info about our contractor's work). During the spring, as part of our mentorship-internship program with GNU, Darshan Kadu worked closely with the GIMP project to update the JPEG 2000 plug-in to use OpenJPEG as its backend library. His code was released in GIMP 2.10.0.

Next, during the spring, our on-site intern Alyssa Rosenzweig documented the state of free software compatibility with single-board ARM computers, designed a fully free remote management device for servers, and developed a tool to make PayPal payments without nonfree JavaScript ([u.fsf.org/fb8](http://u.fsf.org/fb8)). We also welcomed two Outreachy interns, Sonali Singhal and David Hedlund, who both did important work on the Free Software Directory ([directory.fsf.org](http://directory.fsf.org)). Sonali added images to Directory entries, improved our Semantic MediaWiki templates, and upgraded the site to the most recent long-term support version. David made changes to scripts that update the site with information about popular free Mozilla extensions, made adjustments to our license naming format, and researched

performance improvements to the site.

For the fall, we're collaborating with UC Berkeley's Blueprint program to write a phone application that will offer a way to make small, convenient donations to the FSF, and help members keep track of FSF news, blog posts, and petition action alerts. We're also starting with two remote interns, Hrishikesh Barman and Lei Zhao, who will be helping us to improve our system monitoring, and who will contribute to LibreJS and other projects.

All this work is part of our never-ending effort to bring better tools and support to the community so our advocacy work can be better heard, and much-needed free software can be developed. We rely on your support and contributions, both through donations and through intern work. We accept interns four times a year; if you are interested in applying, see the latest schedule at [u.fsf.org/gj](http://u.fsf.org/gj). For information about being an intern through Outreachy, see [outreachy.org](http://outreachy.org). 



Donate to the FSF with Bitcoin  
12DaJbqw5SZTh5DJuEMgr9oPMermG  
h8cF9

Copyright ©2018  
Free Software Foundation, Inc.

The articles in this *Bulletin* are  
individually licensed under the Creative  
Commons Attribution No Derivative  
Works 4.0 license.

<https://creativecommons.org/licenses/by-nd/4.0/>

Published twice yearly by the Free  
Software Foundation, 51 Franklin  
Street, 5th Floor, Boston, MA 02110-  
1335, (617) 542-5942 - [info@fsf.org](mailto:info@fsf.org)

This *Bulletin* was produced using all  
free software, including Scribus and  
GIMP.

Photo credits:

Page 1: Photo by Molly de Blanc,  
Creative Commons Attribution-  
ShareAlike 4.0 International.

Page 5: Illustration by Dana  
Morgenstein, Creative Commons  
Attribution-ShareAlike 4.0  
International.

Page 9: Photo by Adte.ca, Creative  
Commons Attribution-ShareAlike 4.0  
International.

## How to Contribute

### Associate Membership:

*Become an associate member of the FSF. Members will receive a bootable USB card, email forwarding, and an account on the FSF's Jabber/XMPP server. Plus: participate in our members forum at [forum.members.fsf.org](http://forum.members.fsf.org)! To sign up or get more information, visit [member.fsf.org](http://member.fsf.org) or write to [membership@fsf.org](mailto:membership@fsf.org).*

**Online:** Make a donation at [donate.fsf.org](http://donate.fsf.org), or contact [donate@fsf.org](mailto:donate@fsf.org) for more information on supporting the FSF.

**Jobs:** List your job offers on our jobs page: [fsf.org/jobs](http://fsf.org/jobs).

### Free Software Directory:

Browse and download from thousands of different free software projects: [directory.fsf.org](http://directory.fsf.org).

**Volunteer:** To learn more, visit [fsf.org/volunteer](http://fsf.org/volunteer).

**LibrePlanet:** Find local groups in your area or start your own at [libreplanet.org](http://libreplanet.org)! And join us for the yearly LibrePlanet conference on March 23-24, 2019: see [libreplanet.org/2019](http://libreplanet.org/2019).

**Free Software Supporter:** Receive our monthly email newsletter: [fsf.org/fss](http://fsf.org/fss).