FREE SOFTWARE FOUNDATION

Bulletin

Issue 46
Spring 2025

Contents

page 1: There is no collective freedom without you
page 3: Choose the GPL instead of a "no attribution" license for your next program
page 5: I didn't get free software until I became a reverse engineer
page 7: Free software can strengthen the US healthcare system
page 9: Defending Savannah from DDoS attacks

There is no collective freedom without you

By Zoë Kooyman
Executive Director

> *When there is a deliberate choice to restrict, the harmful consequences are deliberate destruction. [...] GNU serves as an example to inspire and a banner to rally others to join us in sharing. This can give us a feeling of harmony which is impossible if we use software that is not free.*

This quote is taken from the GNU Manifesto, which was published a few months before the Free Software Foundation's (FSF) founding (forty years ago this October). It is this philosophy that launched the free software movement. It shaped the definition of "free software" and resulted in the FSF's mission of promoting worldwide computer user freedom.

The FSF's work covers a wide range of activities. Board members and staff speak about free software all over the world and are active in campaigns on a wide variety of topics related to our cause. We educate people about free software philosophy and explain the workings of copyleft and the GNU General Public License (GPL). We work directly with organizations to increase proper use and compliance with the GPL, process copyright assignments from developers (thank you!), and steward the GPL. We get involved legally where we can, like with our amicus brief for the recent *Neo4j, Inc., et al. v. Suhy, et al.* case and our deposition for the *Software Freedom Conservancy Inc. vs. Vizio, Inc.* case, cementing our position that users should be free to enforce their right to source code under the GNU GPL licenses through any available legal mechanism, without having to rely on a copyright

Local meetups, like the one shown here in Athens, Greece, earlier this year, are vital for building momentum for collective action.

holder to take action. Importantly, we run our organization entirely with free software and support the GNU Project, one of the largest free software projects in the world, by maintaining its infrastructure, and do the same for several community projects.

Our work in free software advocacy is making a difference. Research shows that the GPL is the fourth most used license on one of the most popular collaborative developer platforms, and in 2023, the GNU Affero General Public License (AGPL) moved to the fifth position globally. Another recent statistic states that 92% of all software has free software in it.

As a small nonprofit, we do what we can with limited resources, but this progress in adoption of free software is not thanks to our efforts alone. It is with one change at a time, one developer choosing to use a free software license, and one user after another refusing to exchange their user rights for someone else's profit.

When we call **on** you to stop using a program, avoid streaming services, and look for another way when confronted with a choice between free and proprietary software, we aren't calling you **out**. We don't blame you for the fact that the software you use to maintain your daily life, job, or relationships often eats away at your user rights. It is the fault of corporations and the responsibility of governments, employers, and decision-makers. But our success relies heavily on the efforts of individuals and groups working together to achieve a common goal. Not just major, life-changing choices, but also seemingly small objections to, and actions against, the status quo — that is how most social change movements work, and free software is no different.

History has shown that change at the governmental level happens when

people stand up for their rights and make themselves loud enough that they cannot be ignored anymore. Corporations take note when they see consumer choices hit their profit margins. Large groups of people have changed the world around them through concerted action. As an individual, you may feel small, but there's strength in numbers.

Free software isn't an unreasonable right to ask for. In my experience, there are very few people who disagree that you should have control over your computing. But the complications that come with practicing software freedom in today's predominantly proprietary digital society tend to be intimidating or confrontational, and we understand that. We still ask you to speak up, recognizing you are correcting something you didn't do wrong and understanding you may only be able to contribute in a limited way.

If you believe in the cause of user freedom, we need your voice because the FSF will only achieve its mission with it. We ask *you* to call your representatives, we urge *you* to say no to services encumbered by Digital Restrictions Management (DRM), we depend on *you* to choose to use a free software program (even once), to raise the issue in *your* school, to suggest an alternative for videoconferencing at *your* workplace, to speak with *your* family and friends about free software and supporting the FSF, or to take any other form of action, large or small. Only with your involvement (and that of your neighbors, family, and friends) can our collective action be noticed.

Social movements bring about societal or cultural change, often address systemic injustices, and rely on people joining together because collective action multiplies resources, voices, and influence, making large-scale impacts achievable. Free software is a social movement, and I would argue it is one that is intertwined with many others in today's digital society. The way that we can drive change is by having thousands, hundreds of thousands, or millions, of people reject nonfree software. We don't say "reject nonfree software" to shame you — we say it because we need you.

Choose the GPL instead of a "no attribution" license for your next program

By Krzysztof Siewicz
Licensing and Compliance Manager

Just because a license is free dœs not mean it serves the goals of the free software movement well. With no attribution (NA) licenses, things can get really bad. NA licenses are simple, non-copyleft free software licenses, compatible with the GNU General Public License (GPL). But they do not require

preserving copyright and license notices. Using these licenses leads to confusion, liability risk, and taking freedom away from users.

When there is no copyright notice, you are not able to identify who has given permission to use the software. Consequently, it becomes significantly harder to determine if the license was granted by an authorized person. Users may also think that, without a license notice, they have received a nonfree program. Additionally, while NA licenses do not contain a requirement to preserve the copyright and license notices, that dœsn't actually mean these notices can be removed. In some jurisdictions, if not most, removing copyright-related information may actually constitute copyright infringement.

While NA licenses are still free software licenses, even if you preserve the notices anyone who receives the program from you might remove them. Instead of advancing the goals of the free software movement, NA licenses have a saddening antisocial effect. If the notice is removed from a program under this kind of license, it in effect becomes nonfree to anyone who receives the program after. Such users are left on their own to find the source code and confirm freedom from the original distributors. To avoid this major risk, we recommend that you seek differently licensed free software programs that do the same job when possible.

Fortunately, NA licenses have not gained momentum, especially in comparison to the much more protective and popular GNU licenses. Nevertheless, in the past fifteen years or so, we have observed more and more attempts to prevent users from being able to to run, study, modify, copy, distribute, and improve the software. For example, there has been an increasing number of "tivoized" devices (hardware which renders free software nonfree in practice), users getting tricked into using Service as a Software Substitute, or SaaSS, and projects refusing to accept copylefted code. Releasing programs under non-copyleft licenses, including NA, has contributed to these concerning trends.

The FSF believes that the default choice for releasing programs as free software should be the GPLv3 or later, or, for programs designed to interact over the network, the GNU Affero General Public License version 3 or later. If you are a developer looking for a license for your own program, please consider the following: copyleft licenses are designed to ensure that the four freedoms are granted and protect the program against turning nonfree. Non-copyleft free licenses grant these freedoms, too, but the license dœs not protect against them being taken away. NA licenses go one step further: when notices are not

there, users get a risk of liability instead of the four freedoms. NA licenses just make it easier for those who want to take user freedoms away. Please do not be tempted by the apparent simplicity of NA licenses, and preferably release your program under a strong copyleft license instead.

Free as in Freedom

The GNU GPL is the best copyleft protection against threats to freedom.

While the GNU GPL is the best way to protect user freedom, ensuring that violators comply with the license involves enforcement, and compliance is oftentimes a long process that requires significant resources. This is due to the fact that copyleft is based on the law, and enforcement of licenses generally requires legal involvement. There are, however, measures that can be taken to make freedom-protecting enforcement easier, including:

- Releasing programs using (A)GPLv3-or-later notices, which future-proofs them;
- Assigning copyrights to an organization devoted to protecting software freedom (such as the FSF which accepts assignments for programs in the GNU Project);
- Supporting copyright holders like the FSF in pursuing violations, and using any legal mechanism available for enforcing copyleft to non-copyright holders, preferably following the Principles of Community-Oriented GPL Enforcement.

We cannot go back in time and release programs currently licensed under NA with a better free software license: what's done is done. But we *can* make the choice to say no to NA licenses starting now. By avoiding NA licenses, we avoid confusion, liability risk, and antisocial effects. The best you can do now to choose freedom today and long into the future is to use programs released under strong copyleft licenses!

We encourage you to read the FSF's detailed evaluation of two NA licenses: the Zero BSD License and No-Attribution Expat License.

I didn't get free software until I became a reverse engineer

By Joshua Tint
Free software advocate

Free software can remain an abstract concept until you're staring down the barrel of a 10MB executable in a hex editor. It was to me when I began my first year of college. Like many budding software engineers, I saw free software as a subculture for hobbyists and

tinkerers. It was interesting, even admirable, but not particularly relevant to me. I didn't run a free operating system, and didn't see much reason to. I was a computer science student who loved coding, but assumed that proprietary and free software simply cœxisted, each with its place in the world. I didn't begin to see the stakes more clearly until I spent a summer working with a small engineering firm.

Despite being just a summer intern, I was the only person at this company that was constantly getting new clients and projects. That meant I got handed an unusual job—reverse-engineering a proprietary codebase for a medical console. There are certain things I'm not at legal liberty to disclose, but the essence of it was this: our client wanted to manufacture cheaper peripherals (or accessories, such as a mouse) for the console, but the device's software was designed to prevent third-party compatibility. The company behind it—for our purposes, let's just call it Nonfree Software Incorporated (NSI)—had gone to great lengths to lock users into their overpriced scheme.

The peripherals in question were nothing special—essentially 500-dollar hunks of plastic probably manufactured for thirty cents. There was only one interesting aspect of the design: a small EEPROM (a type of memory storage used for small amounts of data), which tracked how many times the peripheral was used. This didn't have a purpose beyond forcing hospitals to purchase more of these peripherals after just a handful of uses, as each had an artificial limit. There was no justifiable technical or medical reason for this; on their own, these peripherals could function indefinitely. The restrictions were artificial, a way to extract more money from medical professionals and, ultimately, their patients.

Artificial restrictions on devices like this one create a lot of unnecessary waste.

For about a month, I worked mostly solo on the project, tasked with peeling apart the console's security measures and figuring out how it communicated with the peripherals. I had always enjoyed coding, but here the C++ I was used to was a tangled mess of x86 assembly, resembling a sprawling and complicated hydra. Even with Ghidra, a libre reverse engineering tool, it took weeks just to unravel the most simple functions. There were times I felt like I was fighting against the very nature of the machine—everywhere I turned, there were more roadblocks, more unnecessary layers, and more hoops

to jump through. But little by little, I chipped away at it.

I painstakingly reconstructed functions, compiling and testing to see if my versions matched the original behavior. Progress was slow, but eventually, the project started to make headway. As time wore on, it became increasingly hard to ignore that I was spending a perfectly good summer on an artificial problem. Had NSI just published their codebase, the whole endeavor would have been completely unnecessary. Yet, the system had been intentionally designed to be as difficult to understand as possible, restricting users instead of helping them.

That summer changed the way I saw software. I had thought of free software as a niche interest, but I began to realize that it is actually about not being subject to the control of a nonfree program or its developer. Seeing how much time and effort had been wasted fighting against artificial restrictions, I couldn't help but wonder: how many projects like ours never made it this far? How many doctors and nurses are stuck with predatory equipment because they don't have the resources to fight back? How many patients have been saddled with unnecessary medical bills without ever realizing that the technology behind their care was designed to protect profit above all? In every sector, in every industry, there are many obtuse barriers like this. It is unjust that healthcare providers and their patients are left prey to these greedy companies and their artificial restrictions.

Sure, free software might just be a subculture for tinkerers, but a handful of motivated tinkerers can do an awful lot of good. If they don't have to put up with proprietary code they can do a whole lot more.

Free software can strenghten the US healthcare system

By Eko K. A. Owen
Outreach & Communications Coordinator

Few people who have interacted with the US healthcare system can report a stress-free and cost-effective experience, no matter as a patient or provider. The reasons for the anemic healthcare system are broad, including the high cost of care and insufficient number of medical practitioners. Other problems are less quantitative, like increasing distrust in providers and treatments and limited preventive care. Free software, such as GNU Health (a free software hospital management information system), has the power to alleviate some of the problems present in the US healthcare system.

The perception that profitability outweighs patient interest, as well as leaking of healthcare data to

companies like Google, among other factors, have led to a 31.4% drop in trust in healthcare providers. Some of this distrust can be blamed on a lack of transparency, which free software can greatly help with. When medical practitioners use tech that runs on free software, their patients can be much more confident that the software works in their best interest. You can examine (or ask someone else) if the health record management system or telehealth software is built in consideration of your health instead of profit margins.

You shouldn't have to worry about your freedom at the doctor's office.

With free software, you can also confirm if your medical history is sufficiently protected. When software is written for the benefit of the patient, there is much less risk of patient medical history becoming a commodity, and more trust from patients. It is the ethical duty of healthcare providers to secure this sensitive information from those who would abuse it. As a plus side for providers, funds (and time) spent on medical data management can be reduced if every patient's information is kept on a single, secure national database. Using free software for managing medical records is the best choice because it builds trust with patients and instills a sense of reliability, which leads to better health outcomes.

When people trust their healthcare providers, they're much more likely to engage in preventive strategies, many of which nowadays include technology in some form or another. Preventive healthcare technology, such as wellness dashboards and early disease detection devices, must respect each person's user freedom and privacy, especially since many programs have access to an individual's biology and entire health history. We should not have to choose between living in freedom and living without disease, and must demand healthcare tech that supports both needs.

We also must advocate for spending more of our financial resources on actual healthcare instead of over-the-top fees for countless proprietary programs. The cost of care was about $4.9 trillion USD in 2023, or $14,570 per person, between private insurance, Medicare, Medicaid, and personal out-of-pocket costs in the US alone. While medical software dœsn't make up the entire price tag, switching to free software could reduce how much is spent overall. If healthcare providers switched to running free software, they wouldn't

be beholden to arbitrary fees or forced updates. There would be no need to rely on proprietary software that might not be compatible, expire quickly, or deliberately obstruct repairs for any technician that isn't proprietor approved. Free software is free as in freedom and therefore not always gratis, yet it has the ability to reduce costs in addition to guaranteeing boundless freedom to anyone who uses it, including healthcare workers.

Adopting free software throughout the entire healthcare industry could also mean reducing the staffing shortage. In 2023, according to the American Association of Colleges of Nursing, 65,766 qualified applications were turned away from nursing programs due to insufficient clinical placement sites, faculty, preceptors, classroom space, and budget constraints. Instead of trying to fund expansion of solely in-person nursing programs, we could strengthen and increase remote education opportunities with free software. If remote healthcare education runs on free software, it would be more capable of being tailored to student needs. As opposed to proprietary software, free software acknowledges the freedom of students to do what they are supposed to be doing — learning and applying that knowledge to trusting patients nationwide.

The US healthcare system can't be fixed with a single solution, but free software can mitigate some of the problems and make it stronger. Whether you're a patient, medical professional, or insurer, you can talk with others about the difference that free software can make in healthcare. An ethical and sustainable healthcare system powered by free software is much more fitting for a free society than one run on proprietary software could ever be.

The Free Software Foundation has moved!
Please send all mail to:
Free Software Foundation
31 Milk Street # 960789
Boston, MA 02196

Defending Savannah from DDoS attacks

By Michael McMahon
GNU/Linux Systems Administrator

Corwin Brust
Jing Luo
Bob Proulx
Savannah hackers

Savannah is under heavy attack, likely from one or more organizations using a massive botnet to build a dataset for training large language models (LLMs). Since January 2025, a distributed denial-of-service (DDoS) attack has been underway. With metrics for our IP blocklist reaching five million in

February 2025. In this article, we will introduce Savannah and some tools and techniques that the Savannah hackers and FSF system administrators use to mitigate DDoS attacks against GNU resources and the FSF network. This series of attacks is not limited to Savannah: staff and volunteers have read about similar attacks against other software forges including Sourceware, Pagure, GitLab instances, SourceHut, and Codeberg, as well as Gitea and Forgejo instances. We hope this article can help others fight these attacks as well.

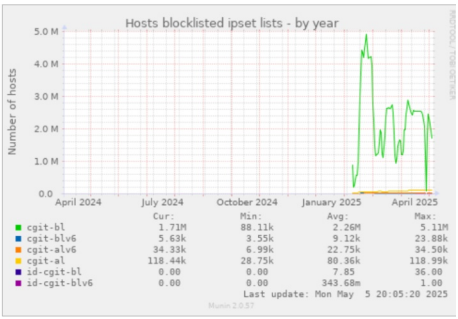
GNU Savannah is the software development forge operated by the GNU Project and hosted by the FSF. GNU Savannah was initially a fork of SourceForge installed by Loïc Dachary, distinguished by an express commitment to only host free software. While `savannah.gnu.org` is reserved for official GNU packages, `savannah.nongnu.org` hosts free software packages that are not officially GNU packages. Savannah is hosted by the FSF with a core infrastructure in Massachusetts, maintained and operated by the Savannah hackers team with the help of the FSF system administrators. Savannah continuously works to maintain an A-grade from the GNU Ethical Repository Criteria Evaluations.

Savannah's hosting is split between many different virtual machines which isolate different functionality, such as front-end web user interface (UI), internal databases, and our supported version control systems (VCS): `bzr`, `cvs`, `hg`, `git`, and `svn` (to view the design of Savannah's infrastructure: `https://u.fsf.org/46z`). The hosts that serve source code for human reading over HTTP and HTTPS currently receive the majority of abuse. These hosts generate web pages with syntax-highlighted source code pertaining to a specific commit to a GNU package in a `git` or other VCS repository.

Defending systems like Savannah from DDoS begins with analysis. Teams must differentiate problematic requests to the system from acceptable ones. For Savannah, analyzing log files revealed a correlation between many of the IPs hitting our servers: it is not one user agent, but many different user agents overlapping simultaneously. This information helped, but it did not solve our problems and introduced a new one! The list of IPs was too large for many of the traditional firewalls. Enter: `ipset`.

`Ipset` is a newer tool for Savannah hackers to help manage large collections of IP addresses. Jing, a Savannah hacker and GNU webmaster based in the Asia-Pacific

region, had been experimenting with it on his own infrastructure successfully. Frustrated with the limits of iptables (a firewall management instrument) and excited by Jing's research, longtime Savannah hacker Bob Proulx immediately put `ipset` to work. It worked fantastically, handling the initial list of two million IP addresses without meaningful degradation of host performance, soon scaling to over five million unique IPv4 addresses.



Ipset is a powerful tool for mitigating DDoS attacks.

`Ipset` mitigated the attack of the moment but, once again, introduced new problems. Many of the addresses were with Internet service providers (ISPs) using Carrier-Grade NAT (CG-NAT). CG-NAT enables individuals to share IP addresses and is used by many ISPs due to IPv4 exhaustion, commonly including people in China, Brazil, Peru, and users of mobile carrier networks. Bob added corresponding allowlists, tracking confirmed "real user" behaviors and exempting them from future bans. This isn't a perfect solution, but it is amazingly effective.

Unfortunately, all of this processing adds up. Hosting software and documentation is a vital part of our work promoting software freedom. This months-long abuse and our continuous work defending against it and adapting to the ever-changing situation presents an enormous drain on resources. Nevertheless, protecting our servers against degradation of service remains one of our highest priority tasks.

To all of the companies crawling the Internet: **there is a better way**! Do not scan code repositories over the web: clone them using version control tools such as `git`, `cvs`, `svn`, `Mercurial`, or `bzr`. Follow the rules set forth in the `robot.txt` files. Identify yourself with a user agent that includes a link describing your activity and a contact address. If your bot was blocked, do not attempt to circumvent the ban. If your program is unblocked after a ban, add more rate-limiting to it. Please contact us with questions by emailing `sysadmin@fsf.org` or visit us on IRC on libera.chat in the `#savannah` or `#fsfsys` channels.

We will fight these attacks for as long as they continue.

The FSF is empowered by individuals like you. Become an associate member at: `join.fsf.org`.

Donate to the FSF with Bitcoin:
1MiL7aKG3YAy8rKqW
HJaoE8w7ZWfFSLmjU

Published biannually by the Free Software Foundation, 31 Milk Street, # 960789, Boston, MA 02196, (617) 542-5942 info@fsf.org

To opt out of physical mailings of the FSF *Bulletin*: https://u.fsf.org/3zh.

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

IMAGE CREDITS

Page 2: *Free software meetup in Athens, Greece* © 2025 by Ellak. This image is licensed under a CC BY-SA 4.0 International license.

Page 5: *GPLv3 Logo* by Free Software Foundation, Inc. This image is dedicated to the public domain under CC0.

Page 6: *Tube à Rayon X dans un hôpital au Bénin 05.* © 2021 by Adoscam. This image is licensed under a CC BY-SA 4.0 International license.

Page 8: *A patient exam room at an urgent care clinic and doctor's office in North Carolina, United States #6.* © 2025 by Harrison Keely. This image is licensed under a CC BY-SA 4.0 International license.

Page 11: *Hosts blocklisted ipset.* © 2025 by Corwin Brust. This image is licensed under a CC BY-SA 4.0 International license.

How to contribute

Associate membership: Become an associate member of the FSF. Associate members will receive a bootable 16GB USB card, email forwarding, and an account on the FSF's Jabber/XMPP and Jitsi servers. Plus: access to our members forum at forum.members.fsf.org! To sign up or get more information, visit member.fsf.org or write to membership@fsf.org.

Support the work we do: Donate at donate.fsf.org, or contact donate@fsf.org for more information on supporting the FSF.

Jobs: List your job openings on our jobs page: fsf.org/jobs.

Free Software Directory: Find free software for any usecase: directory.fsf.org.

Volunteer: To learn more, visit fsf.org/volunteer.

LibrePlanet: Find local groups in your area or start your own at libreplanet.org! You can also use our materials to teach free software at a school near you: u.fsf.org/42i.

Free Software Supporter: Receive our monthly email newsletter: fsf.org/fss.

Full citations are available at fsf.org/bulletin/2025/spring/.