An Anarchist's Guide to Free Software

DC Radical Tech Collective

About the RTC

The Radical Technology Collective is an anonymous organization dedicated to the furtherance of the free software movement and of digital security practices in the anarchist community. The RTC believes that as anarchism enters the 21st century, it will have no option but to embrace these concepts, or perish or mutate into something fundamentally opposed to what it has stood for for centuries.

The RTC sees the Free Software movement as a logical extension of anarchist ideals applied to software. Free Software supports community involvement in the use, distribution, and production of software over corporate involvement, and ensures the user has liberty in the digital realm. Without liberty, and without community, how can we be anarchists?

Additionally, the RTC supports the safe and secure use of the Internet and other computer-related infrastructure to further the goals of the anarchist struggle. We recognize that many, if not most, among us are hunted by state or corporate interests, and we attempt to help anarchists of every color and creed remain secure, safe, and free.

It is the belief of the Radical Technology Collective that technology is neither inherently good or bad, but fully neutral. Therefore, it is not only our responsibility to advance technology as far as humanly possible, but also to ensure it is only used for ethical means. We hold that the negative uses of any technology have the potential to be limited and outweighed by the positive uses. We hold that the access to information is a fundamental human right, and therefore, access to technology related to information, such as computers and the Internet, is a fundamental human right.

Contacting the RTC

You can find the RTC's contact information on our homepage, currently on LibrePlanet:

<http://groups.fsf.org/wiki/RTC>

Our homepage contains links to our materials, our OpenPGP public key, and our current contact information.

Anonymity

All RTC contributors remain anonymous with respect to their work with the RTC. The RTC believes that the source of an idea is irrelevant to its worth, and will never take the source of an idea into regard when attempting to judge the worth of an idea.

An Introduction to Free Software

The central idea of the free software movement is, as is obvious, freedom. The movement seeks nothing more, and nothing less, than freedom for the user in the software realm. The desire for freedom is probably not new to anarchists, seeing as it is the crux of that movement as well. One important distinction to make, however, is between the two meanings in English of the word 'free'. 'Free' can mean either "having liberty", or "without monetary cost". The 'free' in 'free software' refers only to "having liberty". While many, if not most, free programs are distributed at zero cost, some are not, and remain free. The reason for this distinction is not to justify limiting the dissemination of software by means of cost (though that would be impossible with free software), but to point out that being available at zero cost does not make software free.

The free software movement has sort of a creation story, and that's the best way to introduce it. It begins a long, long time ago, in the early 70's, back when all software was free.

In the early days of computing, all software was free. All source code was publicly available, because programs were written in machine code -- as a result, the human-readable "source" code and computer-readable "machine" code were the same thing. Eventually, programs began being written in assembler language, but this wasn't really a divide between source and machine code, since one instruction in assembly language had a 1:1 relationship with machine code on the processor chip. As a result, if you had a program, you had access to the source code, and could study, modify, and improve the program.

Furthermore, in the early days of computing, the community of computer users was very small. There were only a few companies involved in making computers and writing software, and most universities or corporations that used computers wrote their own software. There was a sense of collaboration rather than competition in these days -- if a programmer at a university wrote code for a certain computer, he would freely share it with the programmers at the company that made that computer, or with other universities, or even with other corporations. At this point, in the 1970's and early 1980's, there weren't really companies that dealt only with software, and thus no reason to zealously guard software -- the software was just a thing you could do with the hardware, analogous to instructions for building, say, a bookshelf, coming with a toolkit.

One of the epicenters of this community was at the MIT Artificial Intelligence lab. The AI lab was the birthplace of hackers -- the word, 'hacker', originally meaning prankster (at MIT, senior pranks, or other pranks, were called 'hacks'), came to refer to the people who gathered around the computers in the AI lab, writing software not as a purely utilitarian exercise, but as a lifestyle. Hackers distrusted authority, manifested by the IBM engineers that kept them off the monstrous mainframes outside the AI lab -- while the massive IBM computers were maintained by a cabal of "priests" that limited access to anyone else, the hackers had their own, much smaller, but more accessible, computer that they favored. The hackers disdained hoarding tools or equipment -- they were known to break into offices of people who did so and "liberate" what they needed. The hackers had a culture of sharing -- all the code they wrote was kept (in paper spools, because that's how code was stored then) in a desk in the AI lab, giving anyone the ability to learn from those that had come before, and adapt on their designs to form new things. And the hackers kept a strict meritocracy -- someone came to be respected by the hackers, no matter what their age, degree status, or title, by writing good, elegant or clever, 'hackish', code. The hackers would be hostile to nosy administrators that tried to clear them off the computer for "legitimate" users, and would welcome any who proved their skill (the most notable example of this is Peter D, a local 10-year-old who could out-hack grad students).

Eventually, however, the hacker community, after stretching from MIT to Berkley, fell apart. In the 80's, two companies formed, seeking to profit from the Lisp Machines the AI Lab wrote software for: Symbolics, and LMI. While LMI was more open and hacker-ish in its dealings. Symbolics began to choke off the atmosphere of collaboration that had made the hacker community rise to greatness. The AI lab began to crumble, with all the hackers being drafted off to one of the two companies. One hacker, however, remained stalwart at the AI Lab. His name was Richard Stallman.

Stallman was enraged at Symbolics' efforts to end the free exchange of ideas that the AI lab had represented, and, in retaliation for their actions, re-implemented every new feature Symbolics' programming team produced, and released the code as free software (though it wasn't called that, yet). He was able to keep cloning the output of Symbolics, Inc. for a matter of years.

However, at this point, the genie was out of the bottle. With the advent of higher-level languages with a definite split of source and machine code, it was possible for software distributors to lock out users from modifying their software, and the war between LMI and Symbolics made it clear that non-free software was going to become the norm. With this in mind, Stallman began work on a free operating system, GNU, and years later, founded the Free Software Foundation. Stallman can be credited more than any single other person with starting the free software movement -- he was the first person to realize that free software, formerly the standard way of using software, was going to need a movement to defend it.

The story of the free software movement's beginning is a tragic one, because it is also the story of the free software lifestyle's end. However, since the 80's, the movement has progressed farther than even Stallman could have thought. Today, it is possible to use an operating system with only free software, and even to use an entire computer, from the lowest levels of the hardware up to the OS, with only free software. The Free Software Movement has taken back the ground it held decades ago -- a person can use a computer, and have freedom.

However, "having liberty" is a rather ambiguous way to define free software. As such, the de facto standard for what makes software free is the Free Software Definition[1], originally written by Stallman, and maintained by the Free Software Foundation. The Definition itself is a moderately sized document, but fundamentally, it boils down to four Freedoms software must have if it is to be free. Since the Definition was written by programmers, and programmers count starting from zero (since that's how computers count), the freedoms are numbered zero to four.

Freedom 0: The user is free to use the software, for any purpose.
Freedom 1: The user is free to study and modify the software.
Freedom 2: The user is free to redistribute (share) the software.
Freedom 3: The user is free to redistribute (share) modifications or modified versions of the software.

If a user has all of these freedoms with regards to a given piece of software, that software is free, and if a user sticks to only free software, that user is free.

Software that is available at zero-cost, like "shareware" or "pirated" software, is clearly not free, since even though the user didn't have to pay for the software, they still don't have those four freedoms, and as a result, they're still digitally chained to the software's author when they use it.

Technology, as a rule, is a genie that does not go back into the bottle. No matter how much we might wish it, there was no return to cottage industry from the dawn of the industrial revolution, no return to hunting and gathering after the dawn of the agricultural revolution, and there will be no return to the analog after the digital revolution. The frameworks upon which civilization itself depends are going to change, as everything else will, to incorporate itself into this new digital world.

In short, everything is going to be, at some level, a computer. All computers must run software. If that software is free, the user is as well. If it isn't, the user isn't -- the freedom that a user should be able to exercise turns into a power the software-controlling entity is able to bring to bear against the user.

Like all technologies, the computer is neutral. It can be used to coordinate protests and to spread our message, or it can be used as a tool for the state to further its oppression. It is up to us to ensure that as the world goes into the Digital era, the people of that world do not leave their freedom in the past.

Why anarchists should use free software

When we talk about smashing the state, we often refer to the tools and tactics we use to express dissent. Usually the subjects that arise are black blocs, Really Really Free Markets, zines, pvc pipes, radical spaces and dumpsters. Each of these tools has a unique place in supporting our efforts from feeding us, to protecting one another to spreading information. But one rarely mentioned tool that has the capacity to protect us and liberate us from the institutions of the old world is software.

Like anything else, Information Age technology has been embraced to varying degrees by the anarchist community. While some are overly wary of technology, not using it to communicate at the expense of efficiency, while some use technology too liberally, endangering the community with surveillance. But security and self-preservation are important goals for anarchists, they are not the primary goal -- creating anarchy. It is well known that technology can threaten our security, but what's not as well known is that it can threaten our ideals.

At the heart of the anarchist dream is freedom. We strive to create a world free from coercion, from an oppressive state, from gender and race and every other hierarchy, a world in which we can be free. As such, as anarchism, like every other movement, it swept into the technological age, it is imperative that we evolve and adapt to the changing world, but we must do this in ways that do not betray our goals. The only way to do this is with Free Software.

The "free" in Free Software means free as in freedom. An individual program, or a group of programs acting as an operating system, can be called Free Software, but what Free Software is primarily is a social movement, dedicated to preserving the freedom of computer users. While non-free software (like Microsoft or Apple products) forces you to use your computer on their terms, free software (like GNU/Linux) allows for active participation in a vibrant community.

In a world built on greed, hoarding information is advantageous to the classes that oppress us. To be enslaved by their software, relying on their programs with security holes and inefficiency is exactly what they want. They want you to be forced to buy more products, to be attacked by Adware and Spyware (their own, of course, not that of "criminals" who piggyback on their shitty platforms), and they love that we can't change or even understand their software when we use it. They love that they are the central dictators of their software world -- the only entity allowed to distribute, prohibiting us from sharing, the only entity allowed to modify, prohibiting us from tinkering and improving upon the tools we use. But in a world built on reciprocity and time-honored Do It Yourself attitude, sharing and improving software is one of the most basic freedoms. One of the primary functions of capitalism is to force humans into a passive consumer role, just as Bill Gates wants, and as

Steve Jobs wants, and any other software monarch wants. With free software, the user is a participant in the development and distribution of safe, non-corporate, and most importantly, free, programs.

So if we reject corporate art, corporate media, corporate lifestyles, greed, suburbs, factory farms, capitalism and authoritarianism, why would we want the software that supports it? Why do we rely on authoritarian capitalist software when we could be using software that has a value system truly compatible with our own? We have a better system now. It's time to build a new world out of the monitor and keyboard of the old.

Why you should say Gnu/Linux instead of Linux

When I tell people that I use GNU or GNU/Linux, a lot of the time they don't know what I'm talking about. But when I say "I run Linux", they know just fine. The "open source" revolution has been represented for the most part by Linux, and that's just the term people know. When someone refers Linux or GNU/Linux, they're referring to the same thing: the free operating system, using Linux as a kernel and the GNU utilities as a base. But it's important to say GNU/Linux instead of Linux, especially for radicals.

Linux was released in the early 90's. In 1992, it was released under the terms of the GNU GPL, making it copylefted free software. It gained popularity during the beginning of the dot-com boom in America, and Linux soon gained a lot of publicity for what it was: free software, but not just free, stable and secure, too. Companies running the new Linux operating system on their servers proliferated freely, and people wondered about the "death of Microsoft".

It's important to know that the term Linux wasn't popularized by free software advocates (the open source movement didn't exist at the time), but by the mass media, needing a name for the system that was swarming the dot-coms. The GNU Project had been around for quite a while, since 1983, but they didn't care about that. Linux was the cool new thing. The fact it was the last piece in a system that had been in the making for years was a non-issue. However, there were other reasons why the media decided to use the term Linux.

Let's break things down a bit now. Media outlets aren't really in it to spread information around to the populace. They're there to make money. How do media outlets make money? Advertising. In the tech wold, who pays for advertising?

For the most part? Software companies.

Software companies don't like free software; not because it would put them out of work, but because software companies need users to feel that they aren't in control (see the article WHY FREE SOFTWARE? for more on that), that they aren't free. On that note, let's compare the two people "behind" the Linux project and the GNU project.

For Linux, we have Linus Torvalds. Linus was a student at the University of Helsinki when he wrote the first versions of Linux. Linus is extremely apolitical, and was one of the first supporters of the Open Source movement (see WHY *NOT* OPEN SOURCE for details on the O.S. movement in general). He says that 'open source is the best way of doing things', but is more pragmatically affiliated with open source than ethically: Linus has overseen the incorporation of non-free code into Linux, and has forced kernel hackers to use non-free programs to access the source. He's apolitical as far as freedom goes.

For GNU, we have Richard Stallman. Stallman was a grad student at MIT during the golden age of MIT hackers, but was one of the last hackers to inhabit the MIT AI Lab. During the mid 80's, a series of events happened in succession which proved one thing to Stallman: software must be free. During the days of the AI Lab hackerdom, all code was shared and free. But now, Stallman was encountering more and more non-disclosure agreements and license agreements that prevented him from helping his community or his own interests with software. He started the GNU Project in 1983. In stark contrast to Torvalds, Stallman is extremely political. He speaks out against non-free software, its purveyors, and even the apolitical open-source movement.

Now, let's imagine: If you were the head of a media monolith, or better yet, a tech media monolith, and you depending on millions of dollars in ad revenue from proprietary software companies, what would you call this upstart free OS? Would you call it GNU/Linux, the more techincally proper but politically dangerous name, or Linux, the apolitical and proprietary-software-friendly name?

As activists, it's important for us to put freedom before the ad dollars and doublethink of the software 'industry'. Saying GNU/Linux puts freedom first, and supports OUR goals, not theirs. So the next time you're talking to a friend about your new OS, don't say that you installed Linux. Tell them you installed GNU/Linux.

See also: http://www.gnu.org/gnu/why-gnu-linux.html

Why Radicals Should Not Use Non-Free Software

OR

Non-Free Software Considered Harmful

The biggest reason why anarchists should not use non-free software is the simple fact that it denies them their freedom. By using this software, they are implicitly advocating it and expanding the power that software's developer has over users. The reasons why anarchists should only use freedom-respecting software are obvious to any anarchist that values freedom, but what many do not know is how destructive, subversive, and detrimental to the movement non-free software can be.

Most computer users, anarchists included, don't give much thought, if any, to the software installed on their computer past the immediately pragmatic. This ignorance of both freedom and security ensures that corporations like Apple, Adobe, and Microsoft are de facto standards. However, like all non-free software, the software produced by those corporations is not controlled by the people who end up using the software, but by the initial authors of the software, a closed, private cabal of developers buried inside a corporate structure. These developers have absolute power over a users computer. The code they write, and users run, can and will do anything, from introducing security unintentional security flaws to opening backdoors, from over-aggressive logging that can give away sensitive information to intentionally spying on users.

While anarchists should not use non-free software first because it denies them freedom, the second most immediate reason is the massive risk its use imparts on their struggles, whatever it may be. Non-free software is a "black box", an unknown entity -- there is no knowing what it can do, and what it can do is nearly limitless. However, on notable occasions, the activities of corporations peddling non-free software have come to light as being openly harmful. This article attempts to document the worst offenses in terms of corporations abusing user's misplaced trust through non-free software.

While it's important to be aware of what has happened in the past, it's much better to be wary of what might happen in the future. As the saying goes, an ounce of prevention is worth a pound of cure. Don't take this as a definitive list of "bad" non-free software purveyors -- never trust anything that doesn't allow you your freedom.

Microsoft is quite frequently demonized in the free software community, and not without reason. Microsoft grew to prominence by scolding users for daring to share Atari BASIC, angering scores of hobbyist users who had spread a pre-release version of the long-awaited and long-delayed program. Over the course of its growth, Microsoft came to be known as one of the more cutthroat in a sphere of cutthroats, and in 2000, was actually convicted in the United States of monopolism (though this never came to anything, giving Microsoft an effective antitrust immunity).

Microsoft's software has historically had a shitty security track record. This stems from the fact that Microsoft's first operating systems (and later ones, until around Windows 98) were based on DOS, a single-user system.

We interrupt this article to bring you a fun fact about DOS. Typically, if you ask someone what DOS stands for, they will tell you "Disk Operating System". This is from PC-DOS, the operating system distributed by IBM in the 80's, which was actually just a licensed version of Microsoft's 86-DOS. Microsoft didn't write 86-DOS -- they bought it for a pittance of what it eventually came to be worth from a much smaller company. Its original name should give you an idea of the quality of Microsoft software -- QDOS, for Quick and Dirty Operating System.

All programs in DOS ran in the most privileged mode on the central processor -- this meant, among other things, that a flaw in a single program could bring the entire system down. Since DOS was meant to be used by one user sitting at the console, there wasn't much of an effort to make it secure, and when Microsoft re-wrote Windows from scratch to make Windows NT, it was burdened by having to maintain some compatibility with older code. As such, running a Microsoft program or operating system, assuming no malice on the part of Microsoft (a naive assumption), exposes you to inordinate amounts of security holes. This is why there is such a thriving market for Anti-Virus, Anti-Spyware, and other "security" software for Windows.

Microsoft is a massive software company, with a near monopoly on the home operating system market. As such, it is naive to assume that such a company, with its software running -- completely unaccountable to any outside of Redmond -- on so many computers, would be unapproachd the United States government. Like any capitalist enterprise, Microsoft has no moral obligation to its users -- only financial obligations to its shareholders. Thus, if an arrangement between Microsoft and say, the NSA, or the FBI, were profitable, Microsoft would have no logical objection. However, we need not rely on speculation to show Microsoft's involvement with the state. Their own actions speak far more comprehensively than speculation ever could.

In 1999, Andrew Fernandes was analyzing the cryptographic engine in Windows NT 4 Server Pack 5. This service pack had been shipped to users without having it's symbols stripped, meaning that things like variable and function names from source code were still present in the binary. This mean that previously incomprehensible chunks of hex were labeled and categorized. Once such chunk of hex turned out to be a cryptographic key -- it's name was marked as _NSAKEY. Microsoft immediately denied any collusion with the NSA, asserting that the key was merely a second key used to sign modules to be loaded into the cryptography engine.

United States law prohibits the export of "strong cryptography" -- this is mostly nominal, but software companies like Microsoft still need to abide by the law. Part of their compliance is the fact that cryptographic modules can only be loaded into the Windows NT crypto engine if they are cryptographically "signed" by one of the keys in the system -- either Microsoft's key, or _NSAKEY, or a mysterious third key that was found later. The holders of these keys are the only people able to insert cryptographic software into Windows. While the NSA could use their key to load their own super-secret crypto routines onto their copies of Windows, they could also use it to load backdoored or crippled crypto modules into compromised systems -- for example, dissident groups the government decides would be worth watching. Combined with the fact that the FBI is known to hack into the computers of those it investigates and install their own rootkit, the possibility of a malicious crypto module signed by the NSA's key is not remote.

Microsoft does not, however, rely on government agencies to produce surveillance-ware for its operating system -- it does that for them. In order to aid law enforcement agencies, Microsoft put together COFEE -- Computer Online Forensic Evidence Extractor. COFEE combines 150 tools for extracting passwords, web browser logs, and other information that might help the state monitor its targets. According to Microsoft, all of the tools in COFEE are publicly available and exploit no backdoors in Windows -- but COFEE is only available to law enforcement agencies, so nobody is able to verify Microsoft's claim. In reality, it's likely that if it doesn't exploit any specially-designed backdoors in Windows, COFEE is likely to exploit any one of the myriad of holes in a Windows system.

The best protection against Microsoft's dealings with the state is to not use Microsoft -- though the lessons learned from these two incidents is easily transferable to any non-free software company. Non-free software is opaque to the user and to the world at large. It cannot be accounted for by anyone save the cabal which produced it. Especially since most producers of non-free software are corporations and heavily invested in the maintenance of the status quo, anarchists should never trust them to provide neutral platforms upon which we can work freely and securely.

Free software, although developed and copyrighted by individual people or entities, can be verified as benign by the community, and if any such backdoor was detected, it could be quickly and easily removed.

While the state is the biggest threat to any anarchist, we must be equally concerned about corporate wrongdoing committed not to be law-abiding citizens or for the state's interest at all, but merely for profit. Adobe is the best example of a non-free software company harming the security and privacy of its users for the sake of profit.

The first example of Adobe's aggressive anti-privacy measures is "Flash Cookies", or "Local Shared Objects". Flash cookies are a 'feature' of the Adobe Flash Player. They act in a similar manner to browser cookies, with oneexception -- they are only modifiable by Adobe's non-free Flash Player. This means that a user's web browser is unable to delete them, or even tell the user that they exist. As such, Flash cookies are immune to any browser-based "private browsing" modes.

The implications of this for user privacy are obvious -- where a normal cookie will be deleted by browser-based privacy restrictions, flash cookies will not. Flash cookies are difficult to delete, and can only be done through certain flash-based editor programs. While sites are restricted to reading their own flash cookies (a flash cookie stored by, say, google.com cannot be read by yahoo.com), the only guarantee users have of this is Adobe's word. There is no way of verifying their claim without access to the source code of the Adobe Flash Player, and needless to say, nobody has access to that source code save Adobe itself.

Whenever non-free software runs on your computer, you are giving it the proverbial keys to the castle. There is nearly no limit to what it can do without the user's knowledge. Users of Adobe's CS3 programs learned that when the more vigilant among them realized it was making network connections to a site called "192.168.112.2o7.net" (that's two-letter-oh-seven-dot-net). Of course, a non-free program dialing home isn't exactly new. Non-free programs lock users into a single distribution source, so they have to call back to the Central Control for information on updates and security fixes (when those come and you can get them without paying). But there's something special about the address "192.168.112.2o7.net".

The Internet is a global IP (Internet protocol) network. Every machine on it has an IP address in the form of a special, "dotted-decimal" number -- four numbers ranging from zero to 255, so 0.0.0.0 through 255.255.255.255 are all valid IP addresses. However, some IP address ranges are for internal, non-Internet traffic only -- so that you can have your own network without taking up space on the global internet. Those ranges? 10.xxx.xxx.xxx, 172.16-31.255.255, and -- you guess it -- 192.168.xxx.xxx. So when Adobe had their software connect to "192.168.122.2o7.net", they were deliberately trying to fool people analyzing their own network usage with a firewall or other tool into thinking that the traffic going to the Internet (to the site 2o7.net) was instead going to their local network.

2o7.net is owned by Omniture, a "behavioral analytics firm" -- in other words, a company that buys and sells user's information. From this, we can infer that Adobe is logging what its users are doing, in their software and possibly beyond it (unless you take very stringent measures, any program on a computer can "see" any other and tell, to some extent, what it is doing), and shipping this data off to a collecting house where it can be sold to advertisers -- or to anyone else with the money. While that data might never be put to more harm than sending on-topic spam, there is no limit on the uses to which that data could be put, and there is no way for the user to control those uses in any way.

At this point, it should be obvious how harmful non-free software is to any security-conscious or really, anyone with an expectation of privacy. Non-free software acts as a spy for the state and capitalists, and by allowing it to have free reign over a computer, its users are likely doing their communities a great deal of harm.

Software is, fundamentally, a tool. But we must not fall into the trap of thinking it a common tool, a dumb tool -- a hammer that will hit only what we aim it at, a blade that will cut whatever we place underneath it. Software is a smart tool -- a tool that can serve you loyally, or betray you without a shred of guilt. Non-free software is a set of chains, and an insidious one, for its makers have become adept at making the enslaved unaware of their chains, and even accepting of them. But, at a moment's notice, at the flip of a bit, at the wave of a finger, those chains can bind as tight, if not tighter, than any other.

There is no reason to live chained when there is the possibility of living free. For our communities safety, for our own safety, and most importantly of all, for both their continued freedom, we must shed our chains.

What's Wrong with Software "Piracy"

NB: "PIRACY" IS THE COPYRIGHT FASCIST'S PROPAGANDA TERM TO SLANDER SHARING. WE USED IT IN THE TITLE TO MAKE THIS ARTICLE'S SUBJECT RECOGNIZABLE, BUT WITHIN, WE WILL REFER TO THE SPECIFIC ACTION TAKING PLACE.

There is a fallacy pertaining to the free software fight that is worth rebutting here. That fallacy is the argument "I didn't pay for it, so it's free", where typically the means used to obtain the software is a peer-to-peer system or other illicit means.

The most blatant way this fallacy can be disproved is with simple linguistics. While the software may have been "free as in beer", it was not "free as in freedom" -- using a non-free program without initial monetary cost does not give you freedom. There are far more pressing reasons not to use free software beyond the matter of cost, and indeed, cost is not even discussed in the free software movement, as it is completely tangential.

The propagandists' use of the term "piracy" to mean "forbidden sharing" was not without reason. While the term holds strong negative connotations to the "common citizen", to the youth it is not a negative term, but a positive one. To the young, the pirate is not a figure to be feared, but an icon of personal freedom. If piracy was universally unappealing, we can imagine that Pirates of the Caribbean would have fewer sequels. Anti-authoritarians drawn to classical pirate lore are naturally attracted to the online world of "piracy" as well, and they are encouraged by the pro-piracy movement that has embraced the terminology of its enemy and the images of their namesake. The Pirate Party, initially founded in Sweeden, has expanded to other European countries and has gained seating in the European Parliament. The Pirate Bay has become one of the core facets of BitTorrent -- roughly half of all torrents are tracked by its servers.

These images are not chosen arbitrarily. Filesharing provides possibly the best advertising for non-free software -- it allows users to grow accustomed to non-free software products and insinuates them into culture until they become de facto standards. Adobe Photoshop would not have become a verb had it not been readily available over peer-to-peer networks and other means of distribution, allowing graphic design students and others who might not have been able to obtain it legally to use it and become dependent on it.

Non-free software's long-term marketing strategy, like any other ensnaring substance, has always been "target the children". Schools are given massive discounts on Microsoft Windows and other software, in order to create unbreakable bonds between computer use and non-free software. Companies that produce multimedia-editing software turn a blind eye to filesharing, knowing that if enough professors look the other way at student's illegal downloading of their programs, generations of digital artists will learn not generalizable tools like their AFK counterparts, but will be locked into single programs: Photoshop, Illustrator, and Avid all gain far more profits by creating lifelong users than they lose through not cracking down on student's filesharing.

When one person in a community uses one of these non-free programs, damage is minimal to the community -- their lack of freedom does not transfer. However, problems invariably arise whenever these programs are used in a collaborative setting, since typically, users dependent on non-free programs will reject any free alternative. Even if there is no conflict over the use of unethical software, problems will arise when a free program becomes available to the community -- like any other non-free software, illicitly obtained non-free software does not make switching to any other format and breaking the chains an easy process. Often, a substantial amount of reproduction must occur before a migration is possible -- reproduction that would not need to happen if a free program was used from day zero.

The fact that these programs can be obtained without paying the license fee is completely irrelevant. That does not make them free. Illicit sharing exposes activists to legal problems that weaken the movement. Security issues, caused by accidental bugs or deliberately by user-tracking methods (common in Adobe products, among others), weaken the security provided by other software, making the computer essentially a hostile platform. These programs cannot be adopted by the community, as they can only be altered by a closed cabal of corporate developers. They do not allow the users to exercise their freedom.

Freedom on the Net:
Why Anarchists Shouldn't Use Facebook

There is a misguided conception in radical circles pertaining to free software that has lead most, if not all of us astray. This conception is that of confusing zero-cost access with free access. While it affects use of free operating systems, that turns out to only be half the problem. Most activists, anarchist and free software alike, are unaware and have been until recently completely unaware of a new threat to their freedom: non-free network services.

A network service, or "software as a service", is an installation of software that is accessible to users over a network. Instead of using software running on their computer, users connect to the software over a network. Examples of this paradigm include Facebook, Twitter, GMail, Google Docs, and AIM -- mail services, instant messaging services, and social networking websites are all more general examples of network services.

Free software ideals, specifically the four freedoms, are irrelevant in the context of network services, because the only person "using" the software is the person actually running it on their computer, and none of the users of a network service are doing that -- they're just interacting with it over a network. As such, many of the predominant free software licenses, including the GNU GPL, can be "exploited" by this loophole -- the fact that the public is not a real "user" of the software means that a network service can take free software code, add proprietary modifications that would be illegal to distribute, and then run the network service. Meebo is a good example of this -- it is based on the libpurple library, which is GPL'd. If Meebo were a traditional application, running on the computers of those who actually used it, it would have to be free software to use libpurple, but since it is a network service, it can remain non-free.

The Free Software Foundation, among others in the free software community, understood the danger this loophole posed to the Free World, and in 2007, when they released the third version of the GNU General Public License, they released the GNU Affero General Public License. The main difference between the GPL and the AG PL is that the AGPL mandates free access to source code for users of a network service. This is obviously important for freedom in the network service world.

Unfortunately, however, free access to the source code of a network service is only half the battle. Having the ability to host (at personal cost) an alternative copy of network service software is irrelevant if all the data you've accumulated on a network service is inaccessible. Take Facebook, for example. What use would it be to be able to create alternative Facebooks, if you were unable to take your friends, your pictures, and your messages with you? What use would it be if once you were there, you were in a walled garden, unable to communicate with anyone outside?

The criteria for freedom in a network service is clearly different than that for traditional software. For a network service to be free, a user must have access to two things:

- Freedom to Source: The corresponding source to the network service software under a free license, so that they can have at least all of the freedoms of traditional software

- Freedom to Data: Unfettered access to all of their data in the network service, and the ability to export it in a standardized, portable format so that they are not shackled to one particular instance of a network service.

These two freedoms are nonexistent in the vast majority of network services used by anarchists. Facebook, Twitter, AIM, MSN, all are network services that deny us our freedoms. While it's certainly convenient for anarchists to use these services, and some might be helpful for organizing or protesting, they deny us our freedoms and as such are detrimental.

Non-free network services might also prove themselves to be treacherous, beyond merely being non-free. Most network services used by anarchists are provided by corporations operating within the boundaries, and laws, of the United States -- this means two things. First, the providers of those network services will not be interested in ethics, they will be interested in profit. Second, the providers of those network services will cooperate with the State against anarchists if it is profitable. Should we, as opponents of the State and capitalism, willingly hand over our communications and our social networks to our enemies? Should we risk the shock of losing our infrastructure, if those enemies were to rip it out from under us?

Twitter is a perfect example of the non-free network service. Twitter is obviously non-free software -- no user is able to see how it works, or create their own instances. Twitter does not allow users access to their data in an exportable way. These two factors mean that users are bound to Twitter, and are unable to take their Twitter accounts elsewhere if they wish. Additionally, Twitter is a "walled garden": Twitter users can only communicate with other Twitter users. While anarchists have successfully used Twitter to communicate at actions, most notably the 2008 Republican National Convention, this success has attracted the attention of the US Department of Defense and the Department of Homeland Security (MINILUV, for short). Twitter has shown it is willing to cooperate with the US Government's requests by delaying schedule downtime in order to help Iranian dissidents spread their message (which happens to be parallel to that of the US Government) -- would it not cooperate with an effort to capture "domestic terrorists"?

Twitter is also a perfect example of a solved problem, since a free alternative exists that is free in every sense that Twitter is not. This software is called Laconica.

Laconica is free software licensed under the GNU AGPL. All user data is stored in the open standard FOAF (friend-of-a-friend) format, allowing users to export their data in a single file. Additionally, Laconica is based on a federated protocol, the Open Microblogging Protocol, allowing users on one Laconica installation to communicate with users on another. Laconica implements the Twitter program interface, so any software written for Twitter will work for Laconica. Right now, the largest site running Laconica is identi.ca, but if anarchists wanted to run Laconica on their own servers, and thus have full control over the network service, it would be simple to do. Combined with privacy-enhancing technologies like Tor, anarchists could create fully anonymous and untraceable Laconica instances for use in a single day of direct action -- removing the influence of capitalists and the state on our infrastructure. Microblogging is a relatively new phenomenon, and while the system is in its youth, we as anarchists have an opportunity to influence society as a whole to adopt the free system, rather than the encumbered one, and the best way to do that is to lead by example.

Clearly, non-free network services are not an insurmountable obstacle. Like most other issues in the free software struggle, the real problem is simply human inertia: even anarchists find it difficult to rouse themselves enough to shed the chains of Twitter or Facebook. But if we are to continue struggling for freedom, for consent over coercion, for autonomy over control, we have to shed those chains -- for our own benefit, and for the security of our communities. Right now, network service freedom is in the same state software freedom was in circa possibly 1988: the call for freedom has been heard, but not many have yet heeded it. With time, and with effort, the world of network services can become a part of the Free World -- but we have to put in that effort to make it happen. Anarchists fought for freedom throughout the 20th century -- we must keep that fight alive throughout the 21st, and take it to whatever new battlefields emerge.

Free Software Alternatives

AIM/MSN/Whatever → Pidgin

Photoshop → The GIMP

InDesign → Scribus

Illustrator → Inkscape

MS Word → OpenOffice.org

IE/Opera → Firefox

Outlook → Evolution or Thunderbird

Microsoft Windows or Apple OS X → GNU/Linux