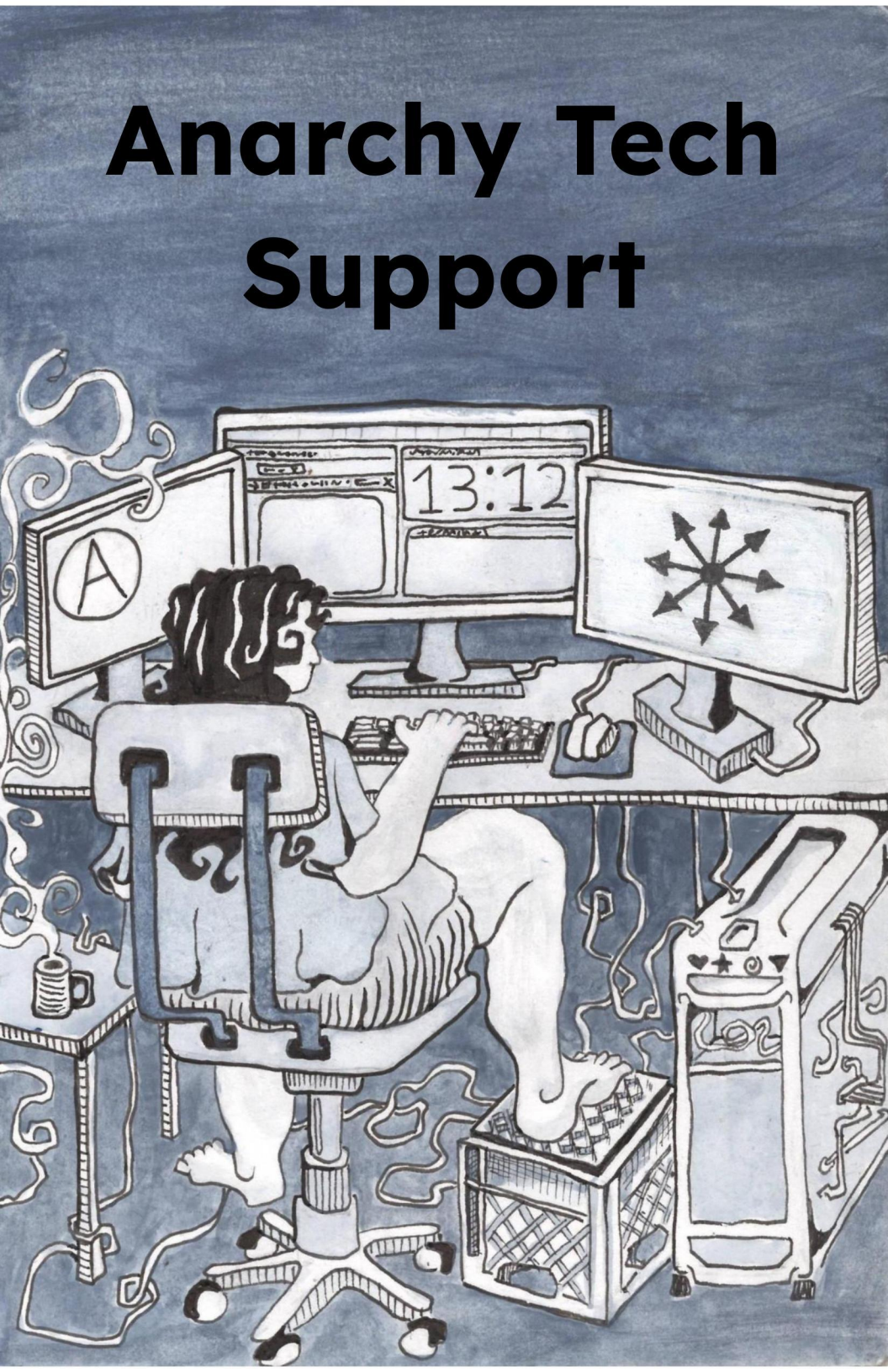
Anarchy Tech Support



Contents

License/Equivalent: 3
To The Distroists: 3
Disclaimer: 3
Background Information: 4
Tech Security Cannot Replace Security Culture 5
The Goals of Anti-Repression Tech Support Work: 6
What You Do: 6
  Helpdesk 9
  Education 10
  Research 10
Hot Tips 11
  Avoiding Being an Asshole: 11
  The Legal Realm: 12
  Some Anarchy Tech Specific Concepts: 13
  On the People Organizing Side: 17
  Some Personal Notes 21
  Practical Tips: 23
Validating Public Keys: 25
On Cellebrite UFED and Such: 27
On Graphene OS: 28
General Principles for Evaluating Tools[1] 28
Low Tech is Best Tech: 33
Teaching 34
  Making Recommendations 34
  Tying Things Together: 36
  On Skill and Responsibility: 36
  Telling People They're Wrong: 38
  You Don't Know What They Don't Know: 39
  Concepts I Put Priority On: 39
Notes for Existing Techs: 40
  Professional Techs: 40
  Linux Hobbyists: 41
  Final Note: 41
Glossary 43

[1]This is not a bug hunting tutorial, but instead a general introduction to evaluating the high level architecture of software for security.

License/Equivalent:

Use (to include in training material, as a prop, for professional development, for political lobbying, as entertainment, inclusion in dossiers, reproduction, distribution, or any other purpose) of this zine by any persons employed in law enforcement, corrections, political lobbying, those holding political office, defense, military, psychiatry, corporate risk management, or intelligence is strictly prohibited. Use of this zine for AI training is also strictly prohibited. Inclusion of this zine in any gated or otherwise restricted library not accessible without credential or monetary restriction to the general public is also prohibited. This zine may be freely edited, updated, copied and distributed by any persons not explicitly excluded above, at cost or for free, on a non-commercial no-one turned away for lack of funds basis. Any work based on this text is also bound by this license.

This is a work of experimental fiction that the explores the lived day to day of a hypothetical overly idealistic anarchist technologist in an alternate timeline where fascists control the USA's government. The formatting of this work as a zine is intentional use of framing to tell a story, much as a fictional atlas or sci-fi alien biology textbook might do the same.

To The Distroists:

While you might know me, or have an idea of who wrote this zine, please politely decline to answer questions about who wrote it. I don't consider true anonymity as the author of this to be practical, but generally want it to be more difficult for trolls, and those interested in doing low bar spurious political prosecution and the like to track down and prove definitive authorship (without resorting to stylometry, exposing long term infiltrators or similar tactics). Additionally, regardless of how you format this zine, please distribute it in a dyslexia friendly font like OpenDyslexic[2] or Lexend[3]. Also y'kno, consider doing that with your other zines so I can like, read them and stuff.

Disclaimer:

I have no formal education in this field. Most of this zine is intended to be read as both one person's opinions, experiences and lessons learned, and my advice to others doing this work. Many people have contributed to this zine, and some have added their own opinions

[2]More info at https://opendyslexic.org

[3]More info at https://lexend.com. Ungated font download at https://fonts.google.com/specimen/Lexend

that may be better informed. It's not authoritative or dogma to be followed. Take everything here with a grain of salt, and do your own research. A lot of my thoughts are up for debate and a matter of opinion; if I didn't think I was right about what I'm saying I wouldn't put it in here, but I certainly have blind spots I don't understand, and this field changes, fast. I'm a nihilistic autistic kid with a special interest. Don't take what I say as dogma, just information about how I once did or may still do this, and the things I wish I knew at the start.

As I'm writing this in mid 2026, the Linux kernel is approaching 2000 CVEs every release[4], largely found by AI models like Claude Mythos, and quantum computing got way scarier[5][6]. There is an international commission/task force being created to target anarchists, and the Autistici/Inventati collective got declared a specially designated terrorist organization and shutdown. Some motherfucker figured out how to turn Wi-Fi AP's into monostatic radar systems[7][8] and published a fucking how-to on it.

Background Information:

This zine centers on the experience of a medium term anarchist in the area dominated by the American state and is being written in mid-2026. I follow a pretty collectivist philosophy about security: while influenced by ones personal circumstances and threat model, security is fundamentally a collective community project. I have hopes to assist others in doing this work, and one of those hopes is for this zine to be a jumping off point for those who don't know where to start or can't see the path forward. This zine is necessarily quite technical at parts, and those without a ton of arcane technical knowledge about this specifically might find areas of it rather befuddling. That's ok. The goal of this zine is to lay out a roadmap of this work for people wanting to take it up, but it doesn't intend to be a comprehensive technical education. Take what you can in reading this and consider searching the internet, or revisiting parts you didn't understand when you are deeper into this work. This work also makes some assumptions about how people will apply the principles outlined here, and doesn't attempt a totalizing social "badness enumeration", not explicitly saying things like "Google Maps used on

[4] https://www.phoronix.com/news/Linux-Kernel-CVEs-Nearly-2000
[5] https://doi.org/10.48550/arXiv.2603.28846
[6] https://doi.org/10.48550/arXiv.2505.15917
[7] https://standards.ieee.org/ieee/802.11bf/11574/
[8] https://doi.org/10.48550/arXiv.2507.12869

Graphene OS still gives a whole lot of data to google" or the like. On a personal note, I'm tired, and I want more people to take on this work. I often feel like I can't take too much time away from communication because I'll be needed for something or another. I can also feel the spark of ease that made this work so possible for me fading, and I worry no-one will be there to take my place amongst these people if I burn out. I also dream of anarchists considering this work a more valued project in the anarchist ecosystem, deserving of the same kind of attention and resources as one would put towards a jail support or medic collective.

I also have no experience designing security systems intended to protect against snitches or torture, and this zine reflects that lack of experience. Awful as it is to think about, and though it may seem far afield for some, given the current political climate, anarchists will likely have to contend with both of these threats more often than we would like, and those taking up this work should consider that.

Tech Security Cannot Replace Security Culture

Many anarchists, especially those new to the movement, disconnected from the wider anarchist milieu, experiencing long term burnout, or who haven't experienced serious repression, will try to substitute technical security measures for social security measures. This isn't necessarily a conscious decision, but is important to note. A no phones meeting with 20 people who only kinda maybe know that each other exist is not a secure meeting just because there are no phones present. There still should not be phones present, but only to raise the bar for surveillance to that of a human infiltrator, even if their presence is already fairly likely. People will use Wi-Fi only Graphene devices to text the sketchiest shit to people they met six months ago.

This is work that cannot, will never, and does not attempt to replace a real culture of resistance grounded in cultural norms informed by movement history about when sensitive information is shared and why[9]. This work instead intends to extend those norms into the realm of digital security, so that we don't nullify our social security culture by our lack of a digital one. To quote the zine turn off your phone "Statistically, your biggest threat BY FAR is people: bragging after an action, suggestive not-really-hypotheticals shared with strangers, suspicions from uninvolved friends/partners/family, intentional

[9]See the text "Confidence Courage Connection Trust" for more info. https://www.sproutdistro.com/catalog/zines/security/confidence-courage/.

cooperation with law enforcement, etc." Real trust built over the course of years of shared work, play, and life lived amongst one another is the only way to build lasting resilient relationships and communities, and the only sustainable way to mitigate the effects of repression.

The Goals of Anti-Repression Tech Support Work:

The purpose of this work is to reduce the state's visibility into the shape and structure (even if it's an informal one) of our movements, while also assisting individuals in staying free as often and for long as possible. This is specifically about preventing visibility, not creating new tools to make something go. Setting up web apps for people and such things are out of scope for this zine as doing this securely could be it's own whole zine, though this can sometimes be very valuable work nonetheless[10]. Sometimes you will be helpful to people in their school projects or similar non-movement work, but that is not the focus of this zine, or my energy.

What You Do:

Anarchy tech support is first and foremost anti-repression work. It's not flashy, and it's not glamorous, it's reproductive labor to keep the people you love safer. A lot of it feels pretty desperate—you watch people around you do things that will screw them over, over and over and over again, and all you can do is grimace and gently prod them in the right direction. Some of the highlights are when people have interesting and novel questions, so you have a reason to learn something new.

Often initial relationships involving this kind of work in a new social group or place evolve along a couple pretty similar paths. That is to say, the part of the relationship involving this kind of work will evolve in general terms along one of the paths listed below, not the entirety of the relationship.

In more informal settings, things often start when people pose questions to a group, often framed as "does anyone know what's up with X thing", or "would someone be down to help me fix X" and you'll be the person who either can already answer the question in a comprehensible and not demeaning way, or the person willing to do the work of finding the answer. This tends to be the key to building these relationships: being the person willing to dedicate an hour (or

[10] R.I.P. Autistici/Inventati

3) randomly to solving someone's problem or answering someone's question.

From there usually a combination of the thoroughness of the information you give, the way you give it, the trust you've built with a person outside this work both social and risk wise, and your general reputation either leads to (1) people asking for more help when they need it, (2) trying to find another person to do this work for them, or (3) worst case, and most common, just ignoring risks. I want to emphasize that this is interpersonal work involving people being vulnerable with you to some level. Often people deeply underestimate how much your ability to not be an asshole will directly correlate to your success in helping keep people out of jail or prison. While this often unfortunately matters much more in this kind of relationship building than the quality of the information you give, take this very seriously as well; you never know why someone might be asking, and their freedom might be threatened by bad or incomplete information. Make sure that the information you give is accurate and qualified with disclaimers about what you don't know.

In more formal settings I often end up starting to do this work when someone is like "we need someone to do a training on X" topic related to this. Sometimes it also happens when people need someone to do a specific task, like make a Jotform or Cryptdrive. Fair warning—being the main security conscious tech in a formal group sucks, and will often result in you ending up doing a lot of invisible labor to make stuff go. If you value your emotional health, or even the longevity of the work (both your personal tech support work, and that of the formal or informal groups you work with), make damn sure you set strong boundaries up front and stick to them. Make sure that as often as security permits you also bring another person along for tasks, both to teach them how to do it, and why you personally do it in that way.

Probably someone whose been doing this shit for 6 months and has a fancy tech job will tell you that using Tails for xyz is too much, and that you should just use a VPN on your daily driver OS or whatever, but doesn't understand the long term risks of escalating campaigns (device seizure, local forensics, spyware etc.), and probably some person whose been doing community organizing stuff 16 years and has touched a terminal like once in their life will tell you that there's a quick hack to make a burner account on X platform that solves most of your problem, and you'll feel deeply grateful and a little silly for all that Monero you moved around. For clarity's sake, this

comparison is saying that the experienced community organizer has much more useful things to offer you than the tech person, but both of them are people to bring along.

Either way, formal or informal, assuming you're doing this work well, probably slowly you'll build a reputation as the person (or ideally collective) that does this stuff, and people will start connecting friends to you so they can ask you questions or help them fix a thing or whatever. This process tends to be informal, but often involves relationships to formal groups. The legal support group that just got their shit rocked by the state will need someone who can answer their questions and help them set up new infrastructure (note I use this word loosely: I don't mean that they will literally need you to set up servers, though this is possible, but that someone who can tell them this tool is the one to use and that make accounts in this way, or offer to do it for them is very useful).

When people have majorly fucked up by failing to take specific precautions, telling them "for the love of fuck why didn't you listen to the people who told you that you should have done this 5 years ago" Is deeply unhelpful because the answer is partially that they didn't take the risks seriously enough, partially because it's expensive, and partially that they didn't know how. Invoking feelings of shame and guilt in these scenarios will probably only actually make them slower to solve the problems. At some point, some kid whose friends just got raided is going to come to you asking how to encrypt their laptop, and you're gonna have to just give them the play by play of how to do that without expressing all the intense emotions you feel about them having not done this yet because those are yours to hold and the emotions they'll feel about yours will only slow them down. Probably neither of you will be able to acknowledge that the recent raids are why they want help, and you'll tell them that Graphene OS really is that good and they really should get it. It'll be just another day in it all. To emphasize, guilt and shame are actively counterproductive because they shut people down. Screaming or crying into a pillow later will be helpful, I promise.

It's helpful to make decisions ahead of time about what you want to prioritize and why, along with your hard lines. How much of people's big emotions about the effects of their actions you are willing to hold? Are you willing to help someone secure their bank account? What about if they're being targeted by Nazis and the like? What if they're a trust fund kid? Are you willing to help someone set up parental controls on their kid's phone? What about if not doing that

means they're gonna be exposed to some other risk? What if you can sabotage said system? Do you do defensive support work for Windows? Etc.

In my experience anarchy tech support has three general parts:
- help-desk
- education
- research

Helpdesk

Help-desk is probably 80% of the outward facing part of it. It blurs a lot with education, but it's worth considering as its own thing. Generally this involves fielding questions about setting up devices and helping people troubleshoot. This is a place where a lot of interventions into lacking security practices are most easily made and best received. That is to say, helping people design new practices when they already intend to change something (e.g. when getting a new phone) or suggesting practices more in line with peoples threat models when something about their current practices isn't working are often more effective than trying to change the practices of people in a steady state. Additionally, frustrated people are the people most likely to make mistakes or decide to do something risky. People are also more likely to use security features that are set up and well integrated from the start. E.g.: people are more likely to use Graphene OS user profiles if their phone starts that way. A useful memorable phrase about this is "there is nothing so permanent as a temporary solution." This is also a place where you can provide tips and tricks that make people's lives a lot easier when doing good security.

For me, an ideal outcome of a help-desk scenario is that someone understands what went wrong, and can fix the problem themselves in the future. Usually the best I can get is that someone vaguely understands what not to do next time. My perception of the limitation is that people usually care a lot about having their device back in working order, but put little priority on understanding the thing, choosing to instead put that time and energy elsewhere. I do get that people are not gonna be experts in everything they do, but it sometimes sucks how little people try to understand the things they use.

A simple example of a novel scenario like this: a while back a Graphene OS user asked me for help verifying a Linux ISO on their device, and walking them through the process of doing hashing and PGP stuff without a terminal was a wonderfully fun puzzle for me.

Education

Education is a complicated one. Generally people are not very receptive to hearing they should change their practices in fundamental ways. People often also just do what the people around them are doing. They conceptualize their practices as personal choice, as opposed to their piece of maintaining a wider security machine used by their community. My experience is that teaching people how these systems work helps get people to change their practices better than most other things. An example here would be running a workshop on the basics of the internet covering things like IP addresses, ISP and server-side logging, basic DNS stuff, packets and headers, TLS, etc. that ends with an introduction to VPNs and tor. Additionally, information is only useful to people in so far as they understand and retain it. A smaller quantity of information that is well understood is much, much more valuable than a vast quantity of information that people can parrot but not critically engage with. How you explain concepts, and the steps you take to re-enforce them both in trainings and helpdesk scenarios should be very carefully considered.

Research

Research often coincides with help-desk and education, but not always. It comes up most often when someone has a novel (to me) question and I have to research it to find out more info. This could also be argued to just be me extending the education portion to myself. The more interesting side though is when someone has a tech question and nobody's published anything on the answer, so I spend a bunch of time reading a spec, capturing traffic, looking at a waterfall graphs, or reading code trying to figure out the answer. This is the part I love the most—it's so fun. I love just being a little tinkerer where I don't even have to come up with what to tinker with. Research is also the part of this work where I get to look at new tools and say "yes this tool is something I might start recommending" or "this looks like snake oil" or "this looks very cool, but doesn't seem ready yet". Most importantly right now the tools I look at or am asked for an opinion on are messaging clients, network privacy/anonymity tools and browsers. These are difficult to make informed decisions on without more in-depth cybersecurity knowledge. See the "evaluating tools" section later in this zine for some jumping off points. Occasionally people ask me for opinions on various quality-of-life apps for Android/Graphene like calendar/notes apps. Generally I will recommend people use paper calendars and notebooks, especially those who use stock Android or iOS, but

for Graphene OS users with low threat models who refuse to do this, generally I'll recommend just getting a note app from accrescent, and whatever the simplest FOSS calendar app in app verifier's database is.

Hot Tips

Avoiding Being an Asshole:

- **Check Yourself:** If you are a white person organizing in primarily not-white groups it can be very grating for people to be given unsolicited tech advice from you, doubly so when you are telling them that a well established practice has very significant risks they don't seem to be considering. Navigating this across language barriers is often even harder. Approach these situations with humility, patience, curiosity, and earnesty, Your ability to be effective in anti-repression work here will heavily depend on your ability not to be an asshole, even more than in other groups.
- **Don't Front:** Be incredibly honest about topics you lack information on. Even if you have a triple PhD in infosec or something, there's a lot you don't know. This ain't a popularity contest: posturing and bad information costs people their freedom, and sometimes their lives. People can also kinda tell when you don't know something, and it hurts your credibility to hide it. In the inverse, it also helps your credibility when you're up front about stuff you don't know.
- **Balancing Autonomy and Risk is Hard:** Balancing personal autonomy and the fact that people are taking risks they don't understand and have the potential for hella big negative effects on other people is really hard. Some motherfucker is gonna show up and tell people they found this new service that solves all your problems, and it swears it's super encrypted and nobody could ever break it etc. Being like "hey everybody this motherfucker is a dumbass don't listen to them" won't help, and makes you seem like an ass, but also not saying anything ain't going to solve the problem either. To be honest, I still haven't figured out what it looks like to do this perfectly. Generally it helps in my experience to try and be gentle, and acknowledge the problems someone is trying to solve, while bringing up both the technical and philosophical problems with their suggestions. When I say philosophical problems, I mean that someone is trying to do something contrary to the principles of anarchist digital security.
  - An example would be someone who wants to make a "safer" smart home system and is suggesting a "revolutionary new tool

that makes it all fine now" or this "look at this new calendar app that syncs everything between your devices, even your contacts! There's even an automated setup system for digital ocean!" and that type of thing.

- **Get Consent:** Practice informed consent when doing this work. Tell people about the edges of your knowledge when you run up against them. Similarly, tell people exactly what you are doing, and why you are doing it. Explain to people what the risks would be if you were a malicious actor touching these parts of their system, or at least a malicious actor with the level of privileges and access they've given you. When I say "privileges" and "access" I mean specifically the scope of ability to modify their system without further authentication or exploitation. For example, if walking someone through system level changes on a machine, you could very easily get them to install a rootkit while lying to them about what something does. They should know that kind of thing is possible when trusting you to do work for them.
  - A practical example of this: some time ago I was helping someone update an expired root certificate. As soon as I diagnosed the problem, I was up front that if I was a malicious actor, I could completely and invisibly (to them) compromise their machine in the process of fixing this, and this person was trusting me not to do that

The Legal Realm:

- **Read Between the Lines:** When looking at repression in the wild, try to read between the lines of things. The state hates admitting that surveillance tech exists, and hates admitting that they used it even more. Expect that the state will do parallel construction around it anytime that they can and look for improbable leaps in their purported reasoning. The repressive apparatus has decided that because theoretically they don't have to hand over all inculpatory (incriminating) evidence, they can just decide not to talk about it in court, avoid mentioning it in case notes, and it's legally ok. While anarchists are well familiar with the fact that our rights exist mostly only on paper, this is especially important to communicate to lawyers when talking about surveillance-tech-shaped holes in discovery. Read up on the case of Daniel Rigmaiden for a real world example[11].
  - As an example, you may never see actual spyware loot get used in court, but obviously the state is using spyware. Several police

[11] Exposed Stingray brand cell-site simulators.

departments have signed NDAs with Magnet Forensics, the maker of Graykey, to never admit to using their device to install malware, despite this being a feature offered by Graykey.

- **Lawyers aren't Security Experts:** Lawyers get kind of high and mighty about privacy, surveillance, and discovery stuff, but the reality is that lawyers rarely know very much at all about the intricacies of novel surveillance tech or infosec generally. Take what they say about this topic specifically as a suggestion from a mildly educated novice, not as an expert opinion.
- **You Don't Have Rights:** The state engages at least as cynically, if not far more so, with the law and legal process as we do. Legal guarantees do not exist for us. Do not expect "rights" to protect data. If data can be accessed by a cooperating third party, like an email provider, web host, anti-ddos service[12], etc., treat it as an almost certainty that it will be.
- **NSA spying** The NSA conducts rampant illegal surveillance, which is generally inadmissible in court, but will give tips to law enforcement[13] who will do parallel construction around it. It is, by design, extremely difficult to know how prevalent this is, but this is a threat people should be aware of and take seriously.

Some Anarchy Tech Specific Concepts:

- **Hide the Who and When, Not Just the What:** Generally speaking, the most important data to protect is that which might reveal the shape and structure of our movements, scenes and millues. In practice this most often means communications data (including communications metadata) and location data. Any serious digital anti repression work, especially at a community level, should put heavy emphasis on these. To emphasize this further, those choosing where repressive resources get allocated often aren't especially interested in whether an individual anarchist has actually committed any specific act, but instead focused on disrupting our movements to prevent any serious challenge to the status quo. Of course investigators will also attempt to search for those who carried out activities that cause a moral panic, inconvenience, or scare those in power as part of a repressive strategy, but this is often largely secondary to attempts to destroy movements as a whole.

[12]Cloudflare is a certificate authority and uses this to read all traffic passing through their anti-ddos service. I'll eat my socks if the NSA isn't all over that shit.

[13]https://www.reuters.com/article/world/uk/exclusive-us-directs-agents-to-cover-up-program-used-to-investigate-americans-idUSBRE97409S/

- **People Will Fuck Up:** Security mistakes are inevitable, a fact of life, especially in the context of long term anonymity or security work. Build in safeguards to your recommendations and practices to reduce the metaphorical blast radius of mistakes. This usually means compartmentalization of activities, even of similar kinds of activities, from other instances of those activities, minimizing the time spent in those compartments, layering defenses, and checklists.
  - In practice this usually means things like dedicated hardware, OS installs, and accounts, along with using anonymously accessed public Wi-Fi, tools that block unexpected unproxied connections, and physical, well researched data destruction methods at the close of compartments.
- **"Privacy" Products Aren't for Us:** The privacy industry is neat because it gives us cover and helps our tinfoil hats seem smaller, but it's not on our side. Proton inc has snitched so[14] many[15] times[16] that I've lost track. They've gone so far as giving up information they previously claimed they didn't have, as in the case of the French climate activist[17] raided and arrested after proton provided IP address logs they claimed they did not have to the French authorities.
- **Don't Trust Web Apps:** If a web based project isn't hosted by an anarchist collective willing to thermite their servers and do jail or prison time to not answer a grand jury subpoena, treat it as untrusted. That doesn't mean don't use it, but does mean to use high quality anonymity tools to access it, and don't put anything into it, you wouldn't want to put into the FBI tip website. Assume such websites are keylogging anytime they are in a focused window. Even if it is hosted by such an anarchist collective, probably still treat it as untrusted when it really matters.
- **Security and Privacy are not the Same:** In our world, one cannot truly be had without the other. Security means that a system will continue to function as expected when faced with an adversary trying to change that. Privacy means data stored within the device

[14]https://www.forbes.com/sites/thomasbrewster/2023/08/08/protonmail-fbi-search-led-to-a-suspect-threatening-a-2020-election-official/

[15]https://www.404media.co/proton-mail-helped-fbi-unmask-anonymous-stop-cop-city-protestor/

[16]https://cyberinsider.com/protonmail-discloses-user-data-leading-to-arrest-in-spain/

[17]https://reporterre.net/Repute-sur-Protonmail-a-livre-a-la-police-des-informations-sur-des-militants-climat

is not accessible to other parties. As a practical example, Steve's shitty FOSS project might not make any outgoing network connections by default, but it also probably has security holes you could drive a paddywagon through. Likewise, Google Chrome is a more secure browser than many, all things considered, but is also feeding everything it possibly can to data brokers who go on to sell to clients like Palantir and security agencies. Another example might be an iPhone in lockdown mode vs Lineage OS. Lineage is fairly private by default, even if not the best at restricting third party applications from spying on you. However, it has pretty shit security. In contrast, iPhones on lockdown mode are somewhat effective at defeating even moderately advanced hacking threats, but still spy on you heavily. Graphene OS bridges this gap, with improved exploit resistance even over iOS lockdown mode, and improved privacy over lineage. Things in this realm of hardened privacy-centric tooling that are usable by the average anarchist in the place and time I organize in are rare, but they are gems when you can find them.

- **Stick to FOSS Tooling:** Sometimes proprietary tools can do shit FOSS stuff can't, but not being able to audit shit sucks. Even if you can't currently do it, if the tool gets big in anarchyland, someone might want to. Yea FOSS ain't really a liberatory thing on its own—RHEL is a huge example of this. That said sometimes a new vuln comes out, and it's really useful to be able to ask someone how it affects X thing. As you build skill, being able to read source code will likely also become very useful sometimes. A recent example of this: when the gpgfail[18] vulns came out, several people concernedly asked me how this affected Graphene OS. Being able to look at the android update verification code and see that not only was GPG not mentioned, but in fact it used OpenSSL directly instead, was very useful.
- **Fuck Biometrics:** Security based on biometrics is worse than nothing for anarchists, and can only give false senses of safety where none is to be had. If the state takes your device, and it has an active biometric authentication system[19], they will physically force you to open it, and gleefully cause you as much pain as possible in the process. The state can also get your biometric data without your knowledge, or use saved data from previous arrests or surveillance to bypass authentication on your devices.

[18]https://gpg.fail

[19]This includes face unlock, fingerprint unlock, retina scans, palm vein scans, or whatever other system they have when you're reading this.

- **Get People (The Fuck) Off Social Media:** Social media is truly one of the greatest security threats to our movements. It is probably one of the most effective counterinsurgency technologies ever created. Getting off of it is probably one of the biggest security and privacy steps one can take. Everything posted on social media is available at almost zero effort to the state, fascists, and anyone else interested in hurting anarchists. The ruling class has somehow engineered a platform where people voluntarily tell them everything they think and do, along with the identities of the people they share these things with, and they knowingly and intentionally use this to influence and repress people. Palantir's whole buisness model is taking data gathered by, amongst other things, social media conglomerates, and putting it into a dashboard used for analysis and targetting.
- **Android Sucks:** Stock android is basically a privacy and security wasteland. There's not much good to be found there and what little exists is usually easily overridden by the vendor, carrier, or google services. All the data collected here really is being fed directly into the state surveillance apparatus. I do not know how to more clearly communicate the intensity of this, and its preceding point. This is a no man's land and very often not worth your time beyond password or VPN recommendations. Many custom ROMs are even worse, especially for security. Pushing people towards Graphene OS even to the point of being mildly annoying and organizing around helping people get compatible devices has been my main recourse here.
- **Be Careful What You Transmit:** Wireless protocols are inherently an added attack surface, but because of the varying range, intended use cases, data retention policies, and implementation security histories, the trustworthiness and ability to safely interact with them without giving away sensitive info varies a lot. In my opinion, the hierarchy in order of most dangerous to safest for mobile devices is
  - cellular (3g/4g LTE/5g)[20]
  - Bluetooth[21]
  - NFC[22]

[20]These are at the top of the list because of the prevalance of exploitation with cell site simulators, and the surveillance legally mandated in the design of cell networks

[21]Bluetooth identifiers are commonly tracked by both surveillance contractors and in stores. Bluetooth is also well known for its poor implementation security.

[22]RFID technologies, of which NFC is a subset, can be readily read when in close physical proximity to RFID readers, which when networked can be used for

- Wi-Fi[23]
- GPS[24]
- **Delete Data Physically** Software secure erase is unreliable. On HDDs overwriting the disk only goes so far, as even an overwritten disk can often be at least partially recovered using powerful microscopes. Drive controller based SSD Wear leveling mechanisms prevent effectively overwriting data on SSDs at all. Security mechanisms built into SSDS have also proven unreliable[25], and as such should not be relied upon for encryption or data destruction.
  - From Anarsec: "If you have sensitive documents that you really want to erase, it is best to physically destroy the USB after reformatting it. Fortunately, USBs are cheap and easy to steal. Be sure to reformat the drive before destroying it; destroying a drive is often a partial solution. Data can still be recovered from disk fragments, and burning a drive requires temperatures higher than a normal fire (e.g. thermite) to be effective. For flash memory drives (USBs, SSDs, SD cards, etc.), use pliers to break the circuit board out of the plastic casing. Use a high-quality house-hold blender to shred the memory chips, including the circuit board, into pieces that are ideally less than two millimeters in size. This blender should not be used for food afterwards, because cleaning it will not adequately remove toxic traces."[26]

On the People Organizing Side:

- **Vet Information Before You Share It:** keep a pool of more technical people, ideally with different specialties, who can help you triage new information. That is, people you can ask about the latest new story claiming "SIGNAL IS COMPROMISED!!!!!1!1!!! SHIT YOUR PANTS RIGHT NOW!!!" or the latest traffic analysis research paper or whatever. Talking about new scary claims in larger groups of novice users often causes panic and confusion, or turn into

---

surveilance, but widespread adoption of this technology is fairly new to the best of my knowledge.

[23] Readily available and fairly ubiquitously used tools allow relatively anonymous connections to Wi-Fi networks, but these tools mst be set up and enabled to provide any benefit

[24] GPS is completely passive, no data is transmitted from a device when using the GPS protocol, however Android and iOS location services will often leak data through things like downloading location specific Wi-Fi mapping data

[25] https://www.usenix.org/legacy/events/fast11/tech/full_papers/Wei.pdf

[26] Look up the talk "adventures in data destruction" from defcon 23 on tails for more info.

misinformation especially when you can't immediately provide useful guidance about them.

- **Corporate Security Failures Cost Money, Ours Cost People:** People with professional backgrounds in tech, be it webdev, vulnerability research, systems administration etc., while very useful as technical references, are often lacking when it comes to things like effectively applying information about what tools state forces have access to and the UX limitations of those things, or to whom or when those tools might be applied. An example of a good question to bring to one of these people is "what could an attacker get by exploiting this new speculative execution vulnerability", or "what are the real world effects of this new AES vulnerability", but very often these people fail to actually grasp the enormity, totality or kinds of threats faced by anarchists in conflict with the state, especially those posed by service provider collaboration, Cellebrite, advanced spyware, and traffic analysis. We face mostly APT level threats backed up by a paramilitary force.
- **Don't Tempt Fate:** Don't recommend, and honestly probably don't even mention tools around more novice users that someone might decide to run with and be tempted to (and easily do) things that would severely negatively effect their security or privacy. Remember that the biggest privacy and security vulnerability in any system is almost always the users. Examples here would be the Debian VM you can use in Graphene, G-suite integration in gnome, or anything related to Magisk.
- **There Were Best Practices Before You:** People tend to get stuck in an "era" about security generally, but especially digital security. The old head refusing to get a smartphone because "that shit spies on you so much, and this is less complicated" is doing the security practices of the era they came up in. The 36-year-old who carries a normal ass phone with a VPN and Signal installed and has an extremely normal apolitical social media presence is too. When approaching these people about security topics, hold some grace for this. Often these people will be resistant to hearing that these practices are insufficient, especially from younger people who haven't been around as long. You've got shit to learn from them, same as them from you, so act like it. It took me a long time to learn that one. My experience is that it's useful to have an actual friendship with these people before you approach them about this. It's also useful to bring up specific ways that threats as well as the adversaries we face have changed, and the ways their current practices don't address this. I have deep respect for the people who kept flip phones from way back when all the way into the mid

2020s, but the truth is most flip phones are just smartphones that flip closed now, with much worse security. They have all the disadvantages of smartphones with few of the benefits. The benefits these people cite of "being able to take out the battery to be sure It's really off" and stuff like that don't really exist anymore. Modern smartphones often have multiple batteries, and while the removable one in flip phones is the main one, there is usually another that allows the device to keep time etc.[27] People who get flip phones now instead of simply being holdovers are being arrogant and chasing novelty over good practice. Yes this is an @- you know better. This same idea applies to desktop users too, so be mindful of that.
    - A specific thing to note here is that once it was more common for anarchists to more publicly associate with each, other digitally and otherwise, as it there was some perceived difficulty in distinguishing between social and politically active connections. This probably never held very much water in the digital age, but existed in a time when the US state was less commonly and successfully openly criminalizing anarchism as a whole.
- **Knowledge is Power:** It often feels bad to be in this position of both power and responsibility where people's apathy leads to them relying on you to keep them safer from various aspects of digital repression. This is so real and sucks so much. I've kind of found that at the end of the day, there's just not really enough workshop time to get people fully up to speed, and people often don't put much energy into doing their own research.
- **Teach Anarchists Tech, not Techs Anarchism:** When bringing people into tech based anti-repression work, it's often most effective to bring people already doing anarchist work, especially those already deeply embedded in ongoing struggles and millues, into doing tech based anti-repression. This is to contrast with the strategy of getting new anarchists with fewer connections that have existing tech knowledge to try and do tech based anti-repression work. My experience is that it's often difficult for people with large amounts of professional tech experience to mentally shift to how different this work is, and struggle to leave their biases, technical and otherwise, at the door. The existing network of relationships those already in our world have will also be of huge help to people looking to take up this work.

[27] RFID nonsense is a whole other threat that ties into this. Ask an embedded systems engineer about this shit for more info.

- **Practices Should Follow Threat Models:** Safety doesn't exist, and trying to achieve it will not get you anywhere except feeling bad. There is only greater and lesser levels of risk. Even if your specific tech stack and social connections are tended in such a manner that the part of the state interested in you doesn't have the premade tools to break them right now, (e.g. no exploits for Xen, no infiltrators in your orbit etc.) they get new tools all the time, and can just commission new tools targeting that stack if they really want to.
  - If you see people starting to fall into these patterns, it's often best to talk to people early. Be careful not to put others in a scenario that risks exposing sensitive activities when doing this and don't just assume people are having a break with consensus reality, though this is sometimes the thing.
- **Don't Ask Questions:** If repression has fallen nearby and someone is urgently asking for tech support, never acknowledge that these things might be related beyond an amused knowing look or a giggle, unless the other person brings it up first.
- **STFU:** Don't share information about who asked for what help. Act as though all information about tech support work for others is need to know unless you have explicit consent to share. Naming that specific tech stacks are common or re-using ideas other people have had is fine, but with this too, be cautious. Uncommon practices can function as a kind of fingerprint, and associate you with the activities of those who told you of these practices.
- **Context Matters:** Always contextualize the information you give people about threats. For example: Yes, in 2026 cell-site simulators can be used to install malware, but more likely it's just used as a fancy geofence and MitTM/PiTM system device.
- **Replace Yourself:** The overarching goal of every helpdesk ticket should be to make people less reliant on expert help to solve their problems. This is often only possible in very small increments but over time it does add up. Try to facilitate non-experts talking to each other and solving each other's problems where you can. This, when done right and responsibly can greatly reduce your workload. Signal based user groups can be very useful for this. Beware however that this can also create a difficult situation where you often end up correcting bad or dangerous advice given by others, and depending on your level of social skills and self awareness can make you come off as like much more of an asshole than you would have otherwise and destroy goodwill you've worked hard to build.

- Having people make changes with the GUI where possible can demystify stuff a lot, and often leads to people understanding what's happening better in my experience. For example: have someone add an apt repo with a graphical file manager and text editor as opposed to with sudoedit

Some Personal Notes

- **Ask for Help:** You're not gonna be an expert in everything. It's just not gonna happen. Anarchy Tech Support encompasses everything from cutting edge RF shit and abstract math, to x86 firmware security, social engineering, and VoIP. Ideally you should try and learn the basics of most things you have to touch regularly to make the work go, and enough to know when something is a big deal or full of hot air, but there are only so many hours in the day. I personally don't know shit about math, RF, or DSP stuff other than that tempest attacks exist, and have generally stayed away from most computer science stuff, but have spent a lot of time learning about a lot of networks and wireless protocols, common vulnerability classes, and browser security, while at least doing my best to learn the basics of Linux and android system internals, software development, firmware security, exploitation and evasion techniques, and some very basic electrical engineering stuff. When RF stuff comes up, as it has recently with the WhoFi stuff and a few other things, I look like a dumbass when I'm asking people about it, and I'm deeply grateful to people for putting up with my bullshit in trying to figure it out.
- **Stay up to Date:** Do your best to stay up to date on new threats and vulns, as well as meta patterns of it all. Hacker News[28], 404 media[29], news chats, talks from hacker conventions (yes even the corporate ones)—keeping your metaphorical finger on the pulse of the tools being acquired by our enemies, and watching trends in real repression, both here and internationally, are all very useful. Data dumps resulting from hacks of police and security agencies or their contractors are often a gold mine. See Blueleaks[30] and the KCPD hack[31] for examples. At present, anarsec[32], the no trace project[33], unicorn riot[34] and the intercept[35] are all very good

[28]https://news.ycombinator.com

[29]https://404media.co

[30]https://data.ddosecrets.org/BlueLeaks/

[31]links to the actual dumps change and go down regularly, but this should give you enough information to find it. https://ddosecrets.org/article/kansas-city-kansas-police-department

[32]https://anarsec.guide

resources. For specifically spyware related info, reading reports from osservatorio nessuno[36], citizen labs[37], Amnesty International[38], and the like can be very useful to get technical details and capability updates. I try to spend 3-5 hours a week on this part. I also generally try to shoehorn learning about a new topic into to this. As an example, when I wake up most days I'll try and spend like 10–15 minutes scrolling one of the sites, looking for things that are relevant, and then once or twice a week I'll watch a video lecture on a new topic I'm learning while doing breakfast. That kind of stuff. Just kind of a regular background information intake.

- **Boundaries Make it Sustainable:** Set boundaries about how you do this work. Some kid named Twig will hit you up at 1am asking why apt won't get packages from a repo they forgot to add the signing key for. Fucking sleep before you fix it, even If it is a two command fix. I personally try to have a policy of not doing this work after 9pm or before 9am. Sometimes people I really like talking to will ask me for help outside those time and I do it because it's nice to talk to them, or someone in another timezone will ask for help, and I'll schedule about it for a time outside that time window. But generally that's what I do. I get the desperation to take people up on this shit whenever they ask because it means they might finally do something safer, and you can increase the probability of it, but eventually you have to take care of yourself. You're a service worker for the movement in doing this, and even though people often don't act like it, your doing shit that keeps the movement going. If you burn out, the probability is nobody will adequately take your place for some years, during which a lot of people will do stupid shit that bites them in the ass. It's often only notable to people when we fail at this, because success just looks like nothing bad happening. You gotta be sustainable about this shit. Your specific shit about this will probably be different, whether you have a different sleep schedule, less of an obsessive interest about it, a job, or kids to take care of, but figure something out and stick to it.
- **Discord is a Cop** Messengers and collaboration platforms lacking strong E2EE should be abandoned basically entirely for any kind of

[35] https://notrace.how
[36] https://unicornriot.ninja
[37] https://theintercept.com
[38] https://osservatorionessuno.org
[39] https://citizenlab.ca
[40] https://securitylab.amnesty.org

radical work. Anytime you find an organization using such a platform, you should make a strong effort to try and get them off of it. Messages on these platforms are often being actively scanned for subversive or illegal content, and peoples activities used to build profiles. Platforms without active scanning, though rare, are still only an email away from handing over all your messages.
    - After the Charlie Kirk incident, this active scanning was used to enable harassment of random people who even sent memes about it in DMs.

Practical Tips:

- **Enable LUKS2 on Install:** If anyone asks you about installing Linux, emphasize to them, they have to enable LUKS (Linux disk encryption standard) during install, and offer to help them do it. They will not be able to easily go back and fix this[39]. So many people have sunk-cost themselves into unencrypted Linux installs because they didn't know it wasn't default. This is probably one of the most important "tech tips" I can communicate to you here.
- **Don't Write Code for Other People:** Just don't. Unless you're willing to endlessly maintain it, just don't write tools to solve problems. Do your best to string together existing tools to make it work, and if you can't, that sucks. Sometimes you can contribute the features you need to an existing tool, essentially passing off the role of maintaining said code to the maintainers of that tool, but this is rare as often your desired features will be considered out of scope for the projects you want to contribute to.
- **Flatpak your Problems Away:** Flatpak is slow and clunky, and I personally don't like using it, but when all else fails, it usually Just Works ™ for getting packages up and running on peoples machines. Keep in mind Bubblejail stuff can cause you a lot of tickets down the line, fair warning. Flatseal[40] may be helpful for this.
- **Be Careful What you Access:** When help-desking people on Linux, I find the most effective way to work through things is to be on a phone call with the person, send them commands over Signal, then have the person send me images of the output in the terminal. This can be aided by piping things to less and using page up/page down. Also when diagnosing problems, ideally the person should take a video of the whole problem, as they likely will give you

[39] This **can** be accomplished via copying the filesystem to encrypted external storage, manually partitioning the disk with LUKS2 and such, then using chroot and moving around the initramfs, but this is fiddly work not for a novice

[40] https://flathub.org/en/apps/com.github.tchx84.Flatseal

accidentally incomplete or misleading information in describing the problem verbally or over text. Signal desktop with screen sharing can also be useful when everyone involved has a high speed reliable internet connection and a high level of trust with each other, but often the latency caused by proxies used by one or both parties causes Signal to use much lower quality video than you would want. Also, a live feed of someone's desktop can come with a lot of privacy issues. Generally I heavily avoid RDP and similar systems.

- **Encrypt Peoples Shit:** Never store anyone else's data unencrypted, and never write others unencrypted data to a storage device. This doesn't mean every notes doc has to be in an encrypted tarball stored in a ramdisk mounted at /tmp/, but rather that you should do all your work from a securely encrypted machine, and that when physically transferring data, you should take the extra 90 seconds it takes to enable LUKS2 on the block device or whatever the disk encryption standard is in the year you are reading this.
  - Some time back one of the Kolektiva.social admins was storing a copy of the whole DB unencrypted and got raided, leading to the state getting the whole damn thing. In 2026 Kolektiva is a mastodon instance run by anti-colonial and anarchist radicals. Even if you think the state doesn't know you exist, you could still get raided for no other reason than you helped someone the state didn't like. Raids are real and unpredictable, act like it. Best practices for this is to never leave a sensitive device in an insecure state in any place and time a raid team or arrest would get to it before you had time to secure it. This usually means turning off your laptop or other devices when leaving them unattended, putting them in your backpack to leave, or before you go to check on that weird banging noise at 3am.
- **Don't Be the Reason Someone gets Compromised:** Heavily prioritize other people's privacy, security, and anonymity when doing this work.
  - When doing online searches about help-desk asks, use tor or similar.
  - When walking people through working on their machines, or if you have to, working on them yourself, ask yourself "if this person had a very high threat model, and was compromised by my advice or actions right now, could I honestly say 'I did my best'".
    - Don't have people get packages in insecure ways (e.g. `curl -fSSL example.com|sh`), never skip verification steps, "distrust the infrastructure" etc.

- Treat HTTPS the same as HTTP. Forging HTTPS certs (using stolen or given certs to sign new certs) is trivial for the intelligence and security arms of the state, if they even have to forge them at all.
- Be careful about anonymity and setup, if something is intended to be used anonymously, ensure that the process of getting it is also similarly anonymous. (e.g.: get simplex over the same Whonix Gateway you intend to use for anonymous messaging)
- Always use PGP, Minisign or similar verification systems for code, binaries, and the like when you can. Verify safety numbers before sending commands to copy and paste.
- Never take risky shortcuts on other peoples machines. You might be willing to `sudo chmod 777 /` on your own machine but don't do that to others.
  - When setting passwords, even ones intended to be immediately changed by another person, use production-ready passwords. In 2026 for disk encryption with LUKS2 using the default parameters for most mainstream Linux distros, as well as Qubes and Tails this means 8 words Diceware[41] (from the EFF large wordlist) or probably 16 for quantum safety. I do not consider myself truly qualified to give minimum viable recommendations on the number of words required for quantum safety, but as Grover's algorithm essentially halves the entropy of a passphrase, 16 words should give equivalent strength to eight[42]. It is difficult to pin down good numbers for the speed of an Argon2 oracle function as no publicly known QPU exists with enough qbits to implement it(largely due to its memory requirements). However, because Grover's algorithm does not parallelize well, we can make extremely conservative estimates and do time based calculations to come up with this. This is not relevant in situations where (passive) service provider collaboration would provide the same information as your password, like your bank account or school email.

Validating Public Keys:

For a long time, one of the tasks I dreaded most in anarchy tech support world was validating public key fingerprints when PGP WOT wasn't an option, because the stakes of fucking it up are so high and the possible vectors for feeding a malicious fingerprint to someone

[41] https://eff.org/dice

[42] https://eprint.iacr.org/2017/811 Note that any entangled oracle function for an Argon2 family KDF will be significantly slower than an equivalent AES oracle, so going with the numbers for AES from this paper (namely a 192 bit target), is very reasonable, if overly conservative.

are so vast. Eventually I broke the possible attack vectors against known good projects down into three general categories:

1. Targeted manipulation of your internet traffic (e.g. HTTPS cert forgery + MiTM/PitM)
2. Source compromise (e.g. a projects website and social media accounts get compromised)
3. Partial machine compromise (e.g. my anon-tech-support qube is compromised)
   - Note: it's largely impossible to do any serious security work from a completely compromised device, so this is out of scope.

Each of These have different, but compatible mitigations.

a) Targeted network manipulation can be defeated by looking up the key fingerprints over multiple connections using a combination of unproxied traffic, tor (ideally via onion mirrors), and sometimes VPN traffic. Asking trusted friends to look up keys from other locations can also be helpful.
b) Source compromise can be defeated by a combination of looking at archived copies of sites and social media profiles on various internet archive sites, looking at forum posts hosted on other infrastructure, and asking friends you trust the practices of for known good copies of the key (e.g. asking your friend known for being very meticulous whose been using Qubes for 2 years for a copy of the QMSK fingerprint from a template.)
c) Partial machine compromise on Qubes can be solved by using fresh DVMs from multiple templates to follow the above practices. In extreme cases, re-install your Whonix templates with

```
sudo qubes-dom0-update --action=reinstall qubes-template-whonix-workstation-xx && sudo qubes-dom0-update --action=reinstall qubes-template-whonix-gateway-xx
```

Often only the first mitigation is needed, and can be done by the person you are supporting on their devices with instruction, but checking their copy of the fingerprint against one you get using the second and third mitigations can boost their security.

A less secure version of these procedures can be done using strong file hashes like sha512 for things like proprietary mainboard firmware updates[43]. Ideally people would move to Coreboot based firmwares, but this is often simply not practical, and can carry its own risks.

[43] This is mainly relevant when fwupd is not a viable option.

On Cellebrite UFED and Such:

To demystify them a bit, Cellebrite's UFED, Magnet Forensics' Graykey and their ilk are mobile device forensic hacking tools. They make extensive use of 0-day exploits and generally exploit things accessible through the USB-C attack surface. This is much bigger than you'd think and can include the GPU via USB-C DisplayPort. The GPU on many Android devices has largely unrestricted memory access. In the USA specifically, the state will also will often keep seized devices long term in some form of Faraday container, allowing for published exploits to be used against them without pre-patch knowledge, especially in the case of seizure during raids or periods of escalated repression against specific fights. Others have said it is common international practice for devices generally to be systematically analyzed on arrest, then returned to their owners in relatively short order after forensic extraction, however this seems less common here. Generally speaking these forensic extraction devices take the form factor of a high end but conventional Windows workstation or a ruggedized Windows tablet. The "magic sauce" so to speak is just a collection of hacking scripts and tools to read the extracted data. Cellebrite largely doesn't find their own exploits, instead relying on the cyber arms trade to sell them the exploits they use on fully patched devices. Some of these devices can also use these exploits to install malware designed to invisibly (to the UI at least) capture the pin or password used to unlock the device, to be given back to the user. These tools are often colloquially referred to as Cellebrite and Graykey.

The most recent leaked documentation for these tools says it basically can't touch Graphene OS BFU or AFU, but this is no guarantee. Rely on moderating your mouth and good passwords to keep the state out. Also worth saying that these USB exploitation tools for forensics also exist to attack desktop. They look very similar to an external hard disk, even using the same USB 3.0 Micro B connector as many USB HDD devices, but have an added small LCD and a button. Once plugged in and activated, they freeze all running process's, dump ramp, and copy the disk to an internal drive. These are carried by at least some raid teams.

Before the early 2020s Cellebrite UFED and its ilk were the most common method of extracting data from phones. They were used in many court cases, including the famous 2017 Disrupt J20 cases.

Spyware is certainly used by the feds[44], likely much more than we think, but also probably still less commonly than Cellebrite tools.

On Graphene OS:

Graphene OS is a hardened mobile operating system based on AOSP. As of right now, it is probably the most difficult to exploit consumer ready operating system we have. It has very serious, well implemented, and intentional defenses against forensic extraction and remote exploitation not present in most other systems. It is not overselling it to say switching to Graphene OS is one of the most effective security and privacy steps an anarchist living under the US regime can take, equivalent to the security upgrade posed by moving to signal in the 2010s. Some have begun using the term "Graphene fetishist" to describe those who push for widespread adoption of Graphene OS. This is a very troublesome development, and one that is rather incomprehensible to me. Many anarchists will say things like "well I just don't put anything sensitive on my phone" when it's suggested they switch to Graphene OS, but my experience is that this is usually either a) a bad faith response, or b) because these people do not understand what is actually sensitive data, believing action specific information is largely the main sensitive information to avoid putting on a device, rather than the people they talk to, how often, their location history etc. Many analysis' of investigative files[45][46], have explicitly shown that this information is widely sought by repressive authorities, both as a source of leads and as evidence in court.

General Principles for Evaluating Tools[47]

- **Use FOSS Shit** Treat proprietary code, especially that written by companies whose business model is based on data collection, as not just untrusted, but malicious, it will butt right up against every security boundary it can and use incredibly novel side channels to get access to new information if it can. If it can change something under guise of an "update" it will. Google will straight up violate the CFAA like it doesn't exist.

[44] Probably also by state and larger local policing agencies, but this has been harder to get a read on.

[45] https://www.notrace.how/resources/read/overview-of-police-investigations-tied-to-a-combative-demo-against-mega-basins.html

[46] https://www.notrace.how/resources/read/dangerous-friendships.html

[47] This is not a bug hunting tutorial, but instead a general introduction to evaluating the high level architecture of software for security.

- **Never Trust a Website** Never trust a website to do what it says with your data, whether they say the data never leaves your device, they never store it, or whatever else, all it takes is an email for the state to silently and invisibly (to most users) change that
  - Never use web based secret generators, such as password generators, PGP clients, or custodial crypto wallets.
- **Don't Tell Servers Your Secrets** Specifically ask "does this tool trust another party with sensitive data or identifiers." If the answer is yes, then probably don't use it.
- **Auto Sync Creates Alot of Room for Mistakes** On top of this, never recommend or use anything for movement data or movement related data (like personal calendars, contact lists, etc.) that auto syncs data with any remote server, even if it's theoretically encrypted. Automated systems like this have greatly added potential for security issues.
  - Treat online collaboration platforms like Cryptpad skeptically, and do not trust them with your freedom or safety
- **Physical Tokens Don't Help** Physical authentication and second factor tokens provide very little security to people like us. A raid or arrest will often get both the token and the device.
  - For those engaged in digital direct action, these tokens can prove rather damning.
  - This is not to be confused with physical anti tamper devices such as the HOTP/TOTP tokens used by heads firmware.
- **Encryption Makes It Go** Strong, well audited, zero knowledge cryptography is what makes having any less governed digital life possible and should be a part of everything we use, but doing cryptography correctly is very difficult. If a service, like a messenger, or video conferencing system is advertising a new cryptographic system, wait till reputable cryptographers review and endorse it before starting to use it. Additionally, only the users should be able to access the relevant keys. Zoom for example claims end-to-end encryption, but generates an AES key on their servers, and then sends it to users to use to encrypt calls to each other[48]. This is a very insecure system and should not be trusted.
- **Prevention Over Patching** Ideal tools proactively try to mitigate unknown exploits by adding and reinforcing security boundaries,

[48]Efficient key exchanges/distribution for encrypted video calling and group messaging is largely an unsolved problem with even signal mostly dodging the question under the hood, but this does not negate the insecurity of the mechanism used by zoom

sandboxing, and hardened toolchains. Hardened_malloc, used by Secureblue, Graphene OS, and formerly used by Whonix is a prime example of this. While most modern production ready systems do this to some extent[49], some are notable for going above and beyond what the industry standards are.

- **Some Tools Get in the Way** Does using this tool preclude the use of other more important tools? If so, probably don't recommend it. For example: android "firewall" apps that allow fine-grained DNS/protocol/IP filtering but use the system VPN function and prevent the use of actual VPNs.
- **Attack Surface Matters** Does a tool add a very large amount of attack surface for small gains?
  - For example: a duress password app that has admin access, receive boot complete, network, keyboard, and lock screen access
- **The More You Hide The Better** For communications tools, look at how they do metadata resistance (if they do it at all), if they implement PQC, and if they have forward or backwards secrecy.
  - When looking at metadata resistance stuff, pay close attention to what a server can access outside the application layer[50] and what account identifiers are available to servers (if relevant) when forwarding messages. This is especially relevant to services that rent or use co-located servers like signal.
- **Security Track Records** Look at previously disclosed vulnerabilities in tools you want to use and see if any of them seem like a symptom of negligence or an overworked team. As an example, for a time briar was shipping a TLS library years out of date out of date with known memory corruptions bugs[51].
- **Circuit Level Security** Prefer access control in hardware whenever possible. Things without microphones or potential microphones[52] cannot be used to listen to you. Things without radios or accelerometers/gyros are much less easily used to track you. Many kill switches offered on devices marketed for privacy do not actually do the thing they say. Either do the work to verify they function as advertised and cannot be bypassed (as with Purism's bogus microphone and camera kill switches that do not disconnect

[49] e.g. Most things have ASLR now.

[50] 'layers' here is referring to OSI model layers

[51] https://github.com/briar/onionwrapper/issues/1

[52] Many sensors, such as accelerometers and cameras, as well as speakers, can be used to record audio, at varying degrees of fidelity.

speakers) or don't put any faith at all in them. This work often involves disassembling the device, and basic electronics knowledge. An oscilloscope and basic SDR can also be very useful for this task.

- Follow an allow list policy as opposed to a deny list policy when making decisions about trusting devices. Many unexpected devices contain the hardware necessary to record audio or track a person. Disposable electronic nicotine vapes are a prime example of this, often containing microphones, microcontrollers, and Bluetooth radios.
  - Trust should be considered on a case by case basis, by device, activity, and benefit: a basic digital watch though it may have a speaker, is a very different case than an iPod classic as it does not have the logic or compute to record or store audio, and even then would have no conventional way to offload recorded audio to a C2 system. Consider making a graph like these to help visualize.

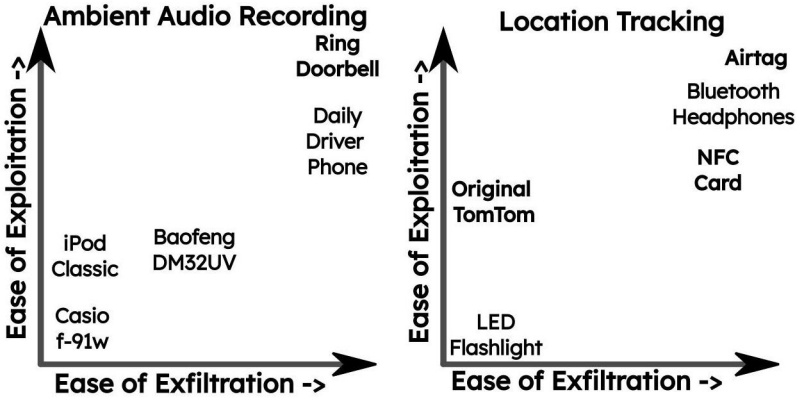


Figure 1: Example Threat Graphs for audio and location surveillance[53]

- A simpler policy is to never bring electronics of any kind to sensitive activities or conversations, but often the benefits of bringing certain devices (e.g. a wristwatch) greatly outweigh the probability of potentially being tracked or having a conversation recorded.
  - For very high threat models, be mindful of the possibility of the possibility of electronic devices and other objects being

[53] Not to Scale

physically modified to enable surveillance, especially parts of vehicles like bikes or cars.

- **Vibe Coding**: Projects largely or completely written by AI are generally worth avoiding. These tend to have serious glaring security issues and are rarely maintained in any serious way.
- **FOSS Doesn't Mean Someone Looked** When evaluating the security of software tools, the most important factors are level of active maintenance, age, and the number of eyes on that code.
  - Check the GitHub/Codeberg/Gitlab/etc. activities tab for the repository it's developed in
  - Check when the most recent changes are and what they are
  - Try and get a sense of the number of users and what kind of users they are. E.g.: Are there a lot of security conscious users with the technical skill and motivation to look for problems (e.g. Whonix, Signal, Cryptsetup) or are there a bunch of users without much rigorous security/privacy focus (e.g. Meshtastic, Lineage OS, jellyfin) etc..
- **Memory Safety** All other things being equal, prioritize tools written in memory-safe languages like Rust and Go over tools written in unsafe languages like C and PHP. If you must use PGP (like for ISO verification or checking canaries), prefer using Sequoia-PGP over GnuPG. Sequoia-Chameleon-GnuPG may be useful for Kleopatra users.
- **TPM Sucks** Never trust a hardware backed keystore or "trusty execution environment" with your safety.
  - This is not to say you should never use a device with a secure element, but rather you should treat it as a cool bonus, and if it is compromised, it shouldn't have enough key material to cough up that your disk could be compromised.
    - An example of something not to do would be storing a LUKS key in a TPM or relying on drive based encryption like opal
    - A good use of a secure element would be something like what Graphene OS does, storing a high entropy random number in a secure element or TPM, then combining it with high entropy user input, like a passphrase, and feeding them into a KDF together.
- **Look out for grifters** Tools marketed to non-professional end users for "privacy" or "security" without technical specifics (meaning cryptography white papers, specific novel system architecture information or the like), especially without explicit discussion of limitations, are generally fairly risky, if not outright snake oil.
- **Chromium > Firefox** Chromium generally does sandboxing better than Firefox/Gecko, and has been more heavily explored for bugs,

meaning that as a rule of thumb, Chromium is a more secure base to build a browser on than Firefox/Gecko.

- **Cost Analysis** Some exploit brokers, such as the Russian state linked "opzero"[54] publish the prices they will pay for exploits in specific systems, and these can be useful tools for threat modeling especially when combined with commentary on such price lists from those in the industry.
- **Don't set People Up to Fail** When recommending tools for use by a broad audience, If you or someone you trust are not training every user of a given system, imagine the most obtuse user you've ever seen, and then imagine them trying to use it, paying special attention to the security features this hypothetical user might inadvertently disable or bypass. If you don't think they could figure it out with at most 2 phone calls to a skilled user, it's probably not simple enough.
  - In this context "someone you trust" doesn't mean "someone you're very confident isn't a police officer and won't inform" but instead "someone you are confident has the skills to carry out this task competently and will actually do it".
- **The Security Budget is Zero Dollars:** People do not have the money to spend to do industry style infosec work, nor the trust. There are no air-gapped networks, forensic specialists or locally controlled root certificates to be had here. Development is usually done on production systems, or the untrusted phone you have lying around. Our best defenses are minimizing attack surface, building systems that help guide users instead of frustrate them, and using hardened systems. Anything that relies on real time monitoring, fancy fuzzy badness enumeration, or an effective timely response from a user is unlikely to be worth much of anything in the current state the anarchist movement here is in right now.

Low Tech is Best Tech:

- Physical paper planners, notebooks, maps and the like are generally much preferable to digital tools, and part of this work is moving people towards these, away from more digitized tools, but like with any information storage tool be mindful of forensics and physical access. A small piece of ceramic tile or glass that doubles as a bookmark can be very useful for mitigating impressions on underlying pages. Use a private shorthand that doesn't look coded to prevent the state or your nosy roommates from getting sensitive information from reading these. Take care to avoid leaving

[54] https://opzero.ru/en

sensitive time metadata in your planner as well (e.g. if you don't party and usually sleep midnight to 9am, don't put "rave" down for 3am the night of a highly sensitive activity). Burn pages when no longer useful.

- Recommend cash over crypto when possible. It's way harder to fuck up in most scenarios and much more private, at least for now. See the German language text "Bargeldtracking Alltag und Moglichkeiten bei der Nachverfolgung von Geldscheinen"[55] for more info on cash tracking. The tl;dr is for long term privacy, use small bills and get change for your bills with strangers that don't use network connected money counting machines in the informal economy (flea markets, street vendors, etc.), ideally do this people who will use them themselves, before spending them on sensitive items.
- Do in person, tech free conversations, or send sealed and or disguised notes with very trusted people over using elaborate digital messaging schemes when at all possible.
- Part of this work is building and troubleshooting infrastructure and culture with those close to you to support lower tech lives.
  - Examples might be setting up doorbells instead of "I'm here" texts, giving directions not just addresses, sharing house keys, and building norms about patience and communication.
- Older pre-smartphone era tech often can provide useful functionality lost by not carrying a phone in everyday life, things like old TomTom car GPS units for navigation, paper maps, and MP3 players for music, but make sure you understand the risks of a device before you recommend it.
  - Note that an MP3 player with Bluetooth still poses a significant tracking risk
  - a TomTom cannot be encrypted or non-destructively wiped, and so should not be used for sensitive travel, or securely destroyed and disposed of immediately after.

Teaching

Making Recommendations

As you build skill and reputation as a person doing this work, people will slowly come to you for recommendations on new tools, and you will generally gain greater ability to influence the practices of people around you. This usually happens on the timescale of years, and is a

[55] Best viewed on tor: https://www.notrace.how/resources/read/bargeldtracking-alltag-und-moglichkeiten-bei-der-nachverfolgung-von-geldscheinen.html

responsibility to take gravely seriously. You are holding people's freedom and potentially lives in your hands. Ideally people would do the work to understand these things themselves, but my experience is they rarely do, so allowing people to rely on you in this way is harm reduction. Keep in mind that you should be prepared to do help-desk work for tools you recommend, unless you qualify the recommendation with the fact that you haven't used it or don't know much about it.

There are a few kinds of recommendations I tend to make:

- Ones that provide a marginal benefit to privacy, security, or anonymity at little to no UX cost to the user. Examples of this are changing Signal settings to make yourself not searchable by phone number, or enabling GUFW on desktop. These are things I try to casually drop in conversation, during trainings, or in hangouts with other nerds, and will offer to do while already doing other work for people, but don't often put much effort into attempting to spread beyond that
- Tools that replace existing everyday tools and provide serious upgrades to privacy, security, or anonymity, but come at a moderate UX cost. These are tools I try pretty hard to push informally, and will run trainings on specifically. This is where the bulk of the help-desk and education work I do tends to live as well.
  - Examples here are Graphene OS and desktop Linux.
- Tools with specialized use cases that provide useful new capabilities outside of someones existing everyday tools, but aren't usually suitable for everyday use. Tools like Tails, Whonix, and Monero fall into this category. These are also tools I will sometimes do trainings on, and often a topic I get a lot of questions about, but legally a thorny place to navigate, and requires heavy contextualization along with precise answers. This is a place where you need to be sure you understand why you are giving the answers you are and know when specific practices are relevant or not. Keeping up with the state of repression, cybercrimes investigation, and the general political climate will be very helpful here. Tread carefully here and be extra up front about the shortcomings of your personal knowledge, as well as potential pitfalls and risks. Unless you know people very well, or are doing public trainings, let people come to you for recommendations on these tools. It will sketch people out if you are constantly sharing tips and tricks about this stuff.
- Tool combinations and configurations for a specific use case. For example: highly anonymous ways to reliably make and receive

calls, accessing a Windows only work program, etc. This is a very large category of work, and generally involves combining different tools that provide specific properties to achieve goals. Increasing the breadth of your knowledge of available tools and services, as well as the depth of your knowledge of the threats against them, will greatly increase your effectiveness at this task. Increasing your pool of knowledge will also increase the elegance of the solutions you can build.

Tying Things Together:

Often I find there's just not enough workshop time to teach people enough theory and concepts for them to tie everything together themselves. By not enough workshop time I mean not that one couldn't do a workshop on this, or add time to an existing workshop to make this go. Instead, people have a tolerance for how many hours they'll spend in a workshop a month, and often their time would be better spent on things like, "hey friend, consider not wearing contact lenses to the protest" or "if you're on top of a mountain wearing only cotton in the middle of the night, you like, might die if it rains" than abstract infosec theory. This often results in things like people with Wi-Fi only Graphene phones not understanding why using Signal Desktop on their unencrypted Windows machine is perhaps an unwise decision, or why not using a VPN on their phone is a bad idea.

How to solve this problem is an open question for me. I try and throw a bit of theory in when I explain things to people, and a lot of zines like "turn off your phone" just sidestep this question entirely, instead making baseline recommendations. I feel uncomfortable with the vulnerability to bad advice this leaves people with, as well as the crushing responsibility to me of "get it right or someone might get hurt". This is not to say "turn off your phone" is a bad zine, but instead to say I personally try to take a different approach.

On Skill and Responsibility:

This work is a very weird combination of hard and soft skills. Of course there are the hard skill facets of this, technical concepts, proficiency with tools, and the like, but the soft skills part of this work is at least as important, as without the ability to effectively communicate with people, to make people feel held through changes, and to work with people to solve the actual problems they need to without pushing them away, none of your hard skills matter to anyone but you. All of these skills have to be built over time, no-one inherently has them. Men, as well as those suffering from allism

spectrum disorder, may struggle especially hard with understanding this. Anarchists taking up this work should take an honest and sober stock of the time they've spent learning these skills, the outcomes of similar situations they've run into, as well as their comfort with using these skills, then consider each task they intend to do and whether they realistically have the skills to accomplish it. This applies doubly so when doing formal trainings or workshops, or recommending changes to well established best practices. As an example: a person who deeply struggles to effectively communicate technical concepts to those immediately around them in an informal setting, should take great care in considering whether to do formal workshops. In considering this, there should be real differences between times when things did not work, and times you expect them to for those expectations to be realistic. Maybe those differences are that you've gained deeper insight into how people who aren't you think about these things. Maybe those immediately around you are deeply irony poisoned and struggle to engage genuinely with things, but the intended audience of your training is very different, or something along those lines. On the hard skill side of this work, make sure you understand why practices have been recommended before you suggest changing them. Additionally, make sure you deeply understand the security implications of things before making new recommendations. Usually this means understanding the mechanics, and ideally workflow, of the state tools you are trying to defeat. A recent situation occurred where someone was loudly making recommendations against strong phone passwords for those who did not use Graphene OS, as they did not understand the difference between BFU and AFU states, and believed that forensic extraction devices would just unlock any phone not running Graphene OS regardless of precautions taken. Understanding things like the android and iOS security models, or even just how they architected their encryption, as well as the attack flow of the tools these practices were trying to defend against would have solved this problem. While this person should not have been running workshops on this topic, some of the people doing anarchy tech support in similar or adjacent circles have struggled in navigating this situation as a number of non-technical people were resistant to the idea that this person was unqualified to do this work. Anyone taking up this work has a serious responsibility to make sure they are qualified to do the tasks they intend to carry out before attempting them in situations that effect others.

Telling People They're Wrong:

This is a hard subject. If you're anything like me, telling people they're wrong is difficult and scary. This is often compounded by the fact that often, people acting on misinformation will often have already been unknowingly exposed to risks that may have potentially life altering consequences down the line. Be sensitive about this, especially to the fact that people will often not be able to acknowledge this is true. Generally it's good ask consent before sharing information like this, but it should be balanced with the tone of the conversation and the severity of the misinformation.

- A person thinking that airplane mode and a Faraday bag makes your phone fine to bring to sensitive locations and activities, is a different thing than an activist legal aid group using Gmail because "attorney client privilege will protect them", and needs to be treated differently. In both of these situations, people are greatly increasing the risk faced by others, but the choices made by the legal aid group are likely to hurt more people, with less risk to the legal aid group, and the person carrying the phone is putting themselves in at least equal increased risk as the people around them, but likely putting less total people at directly increased risk. Approaching the person who is increasing their own risk should probably be done with a different tack than the people who aren't, and the people increasing many people's risk should probably be prioritized over the person who is effecting fewer people, though this is situational.

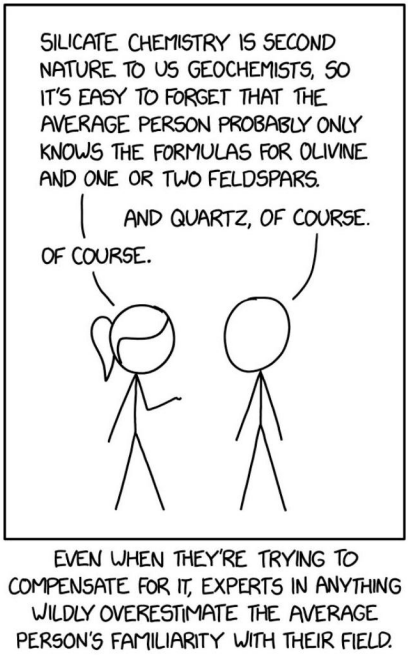


Figure 2: credit: Randall Munroe, https://xkcd.com

You Don't Know What They Don't Know:

Many of us who would take up this project know at least at a high level what things like proxies, servers, operating systems and encryption are, and would assume the people around us would have learned these things in middle or at least early high-school. Increasingly, people coming into the movement do not even know things that to me feel like they are common knowledge. For me, this is a function of being an autistic nerd who hangs out with similar people, and a blind-spot I am actively trying to fix. A lot of the education side of anarchy tech support ends up just being filling in the holes in people's knowledge left by the education system.

Concepts I Put Priority On:

Generally these are concepts I will take time out of a conversation I'm already in to convey, or put special attention to in workshops.

High level idea of what are:

- operating systems
- servers
- encryption
- IP addresses (in 2026 I stick to mostly ipv4 stuff)

- proxies
- communications metadata[56]
- malware
- software updates

And basic awareness of these tools:
- Signal Messenger
- VPNs
- Tor Browser Bundle
- Tails
- Diceware
- Graphene OS
- Desktop Linux
- Monero and Zcash
- Faraday bags

Notes for Existing Techs:

Professional Techs:

In my experience watching others try and pick up this work, the most relevant and transferable knowledge to this work is Linux, Android, and Xen system internals, along with networking fundamentals, followed by bug hunting and systems administration + helpdesk. 80% of problems are still user error, but another 20% are the available tools being kinda shitty FOSS projects. Which to my understanding of the enterprise space is a significant shift in ratio. If you are just coming into the movement, for fucks sake, please have some humility when approaching this work. The people here do not care about your credentials or years of experience or your PhD in Kubernetes biscuits, they care about usable solutions and that's kind of it. They are fighting against and for things that matter extremely deeply to them, and It's usually far more important to them to succeed in their organizing goals than to mitigate all risk or avoid technical debt or rewriting Jitsi in Wasm or whatever you're excited about at the moment.

If you want to be useful about these things show up and do the stuff that needs done when you can, carry the shit buckets, turn the compost, do the canvassing, and volunteer for tech tasks when they come up like anyone else. Don't come with the intention to be some kind of tech savior, people will probably think you're an asshole and tell you some more NVC version of "shove it".

[56] Specifically what can an adversary gain from it, I often don't use the word itself

Linux Hobbyists:

The average person who does movement work generally isn't excited about the same stuff as you. Terminals scare people and the idea of learning the keybinds for sway might actually give someone a heart attack. The most important thing on desktop generally is getting people onto decent Linux systems, and off of windows or mac. You might have your gripes with SystemD or KDE and think Arch[57] with DWM and Runit is the thing everyone should use all the time, but the truth is, most people don't care, and won't care. You are probably right that whatever you're on about is technically the better solution, but people don't care about that. They want something that gets out of their way and lets them keep doing the political work they care about.

Generally in 2026 this means recommending Debian based distros, GUI package managers, and KDE, Gnome or Cinnamon as desktop environments. Maybe Cosmic in a few years but not now. I'll tell gamer types generally to use Bazzite, but beyond that Fedora stuff generally generates a lot more tickets for me to work on than Debian. Also crossing my fingers for Secureblue, but it's still not ready as of summer 2026. Occasionally people will fall into a kind of security enthusiast category for lack of a better word, or just want something that will force them to learn Linux a bit more. Generally I'll recommend Fedora to these people as a like, slightly more secure thing that has more up-to-date packages and more stuff available. It's been the Linux distro that's the most fun to use for me also, and I think some others also experience this.

Final Note:

Having read this, I hope you have a better picture of what this kind of anti-repression tech work looks like. I ain't one of those propaganda types that will tell you "we're already winning" and "we overestimate the state's capabilities, just kill the cop in your head" because both of those are lies. When it comes to tech, most anarchists in the time and place I organize in vastly underestimate the capabilities of the state. This work exists to help counter that, and the long term progress I and others have made in doing this work is meaningful in its own right. Seeing practices like abandoning social media, Diceware passphrases, Wi-Fi only Graphene phones, and Tails use grow to significant adoption has been deeply encouraging to me. I cannot take sole credit for these practices in the wider anarchist scene, but can confidently place myself amongst

[57] BTW :wq

those who spread these practices amongst a new generation of anarchists in this millue.

If you take up this work, in 5 years you might be able to say the same of some other practices, maybe anarchists will move to Sequioa-PGP, or maybe the Nym mixnet will replace tor in common use, or maybe it'll be something completely out of left field that I can't wrap my brain around. Whatever it is, this work gives you the possibility of having a lasting positive impact on the people you love. And, if you do it right, they'll probably never understand the intensity of the impact you had.

Glossary

alphabetical by term

0day A security vulnerability in a piece of software the authors have not yet released a patch for, so named because defenders have had 0 days to patch it.

Accrescent An app store with modern (for 2026) security measures endorsed by the Graphene OS project that provides high quality privacy focused apps. See https://accrescent.app for more info.

Advanced Package Tool (apt) The default CLI tool used to install software on most Debian based distros.

Adversary A hostile actor making an intentional effort to bypass your security measures, usually by means of cyberattack.

After first unlock (AFU) The state of a phone that has had the unlock password entered since the last time it was rebooted or powered off. First unlock describes the first time this password is entered after a reboot, and thus a locked phone (A phone showing the lock screen) is still considered to be AFU if it has been unlocked since it was last rebooted.

Allism A debilitaing disease affecting those tragically lacking autism. Widely known for their inability to be direct, incessant eye contact, and lack of passionate interest, those suffering from allism can be easily recognized by their uncanny tendency to speak at length while saying nothing at all and deep discomfort with proximity to those unlike them.

Autism A neurotype that makes you hot, smart, and full of an uncontrollable urge to resurrect Ole Ivar Lovaas so you can beat him to death with your bare hands

Backward Secrecy A property of an encryption scheme that means an attacker gaining access to one encryption key, doesn't inherently gain information about any other previous key. So for example, if Mal gets the encryption key Alice used to encrypt a Signal message to message Bob that reads "it's under the sink," Mal can't use that to read Alice's previous message. see also: 'Foreward secrecy'

Before first unlock(BFU) Before first unlock, the state of a smartphone that has not had the password, pin, or other user input for unlocking en-

tered since it was last powered off.

Bits of entropy A measure of the randomness of a thing, expressed as a number. A passphrase with 100 bits of entropy is equally likely to be any one of 2^{100} possible passphrases. Each additional bit of entropy doubles the difficulty of guessing it.

Blue team A term used by cybersecurity professionals to refer to those defending a system; when used as a verb it generally refers to the process of configuring, maintaining and monitoring the systems you are defending, along with responding to incidents.

Command and control (C2) In cybersecurity Command and control is used to refer to any system used to direct malware, but is also often used to receive stolen data, sometimes called loot. Cobalt strike is one such software suite.

Computer fraud and abuse act (C.F.A.A.) A piece of legislation (codified as 18 U.S.C. § 1030) passed by the USA federal government in a moral panic after the release of the movie "WarGames" that criminalizes "exceeding authorized access" on functionally any internet connected computer, with special provisions for banking and government computers. See https://en.wikipedia.org/wiki/Computer_Fraud_and_Abuse_Act for more info

Cryptographically secure pseudorandom number generator (CSPRNG) A mathematical function or hardware system capable of creating nearly random numbers in a manner unpredictable enough to be safely used in cryptography

Cryptsetup From the Man page: "cryptsetup is a utility for configuring and managing full-disk encryption on storage devices. It can encrypt block devices (such as hard drives or partitions) and containers (disk images stored as files)."

Daily driver This is a colloquial term popular at the time of writing indicating that something is the primary one of that thing that you use. One might describe the phone they carry all the time as their "daily driver" to contrast it to a burner phone, or a dedicated phone used only for sex work.

Debian A point release Linux distro designed to prioritize stability over being cutting edge. Upstream of Ubuntu, Linux Mint, and Tails. See Debian.org for more info.

Diceware A prescriptive method of generating simple, secure, and easy to memorize passphrases using dice or a CSPRNG, see https://theworld.com/%7Ereinhold/dicewarefaq.html for more info.

Fedora A linux distro used as a testing ground for Redhat Enterprise Linux (RHEL). Generally much more up to date than Debian, but at a stability cost. See Fedoraproject.org for more info.

Entropy The level of and quality randomness of a given thing, usually quantified in so called bits.

Exploit kit (often shortened to exploit) Code written to take advantage of a bug or glitch, called a vulnerability, in another piece of software, that allows an attacker to gain information or access they would not have otherwise.

Forward secrecy A property of a messaging encryption scheme that means an attacker gaining access to one encryption key, doesn't inherently gain information about any other key moving forward. So for example, if Mal gets the encryption key Alice used to encrypt a Signal message to message Bob that reads "have you seen the mustard?" Mal can't use that to read Alice's next message. see also: 'Backward Secrecy'

FOSS Free open source software, read https://en.wikipedia.org/wiki/Free_and_open-source_software for more info.

Geofence A method to query data in which an investigator defines a spatial boundary on a map and requests information about devices inside that boundary in a specific time frame.

Graphene OS A hardened mobile operating system widely used by anarchists forked from android that maintains android app compatibility. See https://grapheneos.org/ and https://www.anarsec.guide/posts/grapheneos/ for more details.

GUFW GUFW is a graphical tool for managing "uncomplicated firewall(UFW)" a simple firewall application for Linux.

Key derivation function(KDF) An algorithm used in the process of turning some combination of input values into a usable encryption key.

Linux unified key setup (LUKS) The default disk and container encryption scheme used by most Linux distros, usually managed by Cryptsetup read https://en.wikipedia.org/wiki/Linux_

Unified_Key_Setup for more info.

Man in the Middle/ Person in the Middle (MiTM/PiTM) A form of cyberattack in which an adversary impersonates two parties to each other, often allowing this adversary to decrypt, modify, or block information sent from each party to the other.

Man Page A manual page for a given command or utility, usually accessed via the command `man name_of_program`. Run the command man man on any Linux system for more information.

Operating system (OS) The software that manages your computer hardware and resources, accepts user inputs, and provides services that your apps use. Windows, macOS, Android, iOS, and Debian are all operating systems.

Post-Quantum cryptography (PQC) Cryptographic algorithms designed to be resistant to attackers capable of rapidly doing new kinds of math enabled by large quantum computers. See https://en.wikipedia.org/wiki/Post-quantum_cryptography for more info.

Quantum computer A new novel kind of computer that enables very high speed computation of some math problems, notably many relevant to modern "classical" encryption, when compared with classical transistor based computers. Read https://en.wikipedia.org/wiki/Quantum_computing for more info.

Repression Attempts by the state or fascists to destroy, disrupt, or quell social movements.

Red Hat Enterprise Linux (RHEL) A commercial Linux distribution developed by the red hat corporation widely used by military and defense contractors, with a subscription based model.

Tails Short for the amnesiac incognito live system, is an operating system designed to be installed on an external drive like a USB stick or a USB SSD, that runs in ram and leaves no digital traces on the local machine of it's use, and routes all network traffic through tor. It also leaves no traces of user activity on the drive it's installed on unless you use persistent storage, and then the traces are encrypted. See tails.net for more information.

Thermite A class of dangerous pyrotechnic mixture made of metal powders used for railroad and ship welding. When activated creates an exothermic reaction in ex-

cess of 5000°F. Do not attempt use if you are not a trained welder or experienced chemist. "Thermiting" ones hard drives is something of a meme, generally used hacker circles as a euphemism for complete irreversible physical data destruction, with a connotation of being done in an emergency or in a state of paranoia.

Tempest attack From wikipedia "TEMPEST is a codename under the U.S. National Security Agency specification and a NATO certification referring to spying on information systems through leaking emanations, including unintentional radio or electrical signals, sounds, and vibrations." See https://en.wikipedia.org/wiki/Tempest_(codename) for more info.

Turn off your phone A tech security basics zine from 2023. See https://www.sproutdistro.com/catalog/zines/security/turn-off-your-phone/ for more info.

Upstream In open source software development, many projects directly use other projects as components of their own. When a property of a project is the result of another project being used as a component, this is described as being "upstream". A simple example of this would be, if an Android update broke app notifications, and someone contacted the Signal foundation for support, Signal would say this was an "upstream issue", and thus they couldn't do anything about it.

UX User experience, the overall design of how a user interacts with something, how intuitive the menus are, what is explained in tooltips, how it looks, if it crashes all the time etc.

VPN Short for virtual private network, this technology is used for many things, but in this context generally refers to the use of the OpenVPN or WireGuard protocols to create an encrypted connection to a remote server run by a third party such as Riseup or Mullvad that sends your traffic as if it were it's own onto it's next destination e.g. a website, or the Signal servers.

Web apps Applications designed to be used in the browser, or that primarily rely on a remote server to perform it's intended function. Things like Google Docs, Mastodon, Nextcloud, Office 365 etc.

This text is an attempt to share useful lessons I've gained doing tech based anti-repression work for radical movements. It's also something of a vent, an attempt at trying to communicate what this thing I do is. My hope is that someone recognizing digital security failures in their community can sit down, read this, and have a jumping off point into doing this work.

Copies of the original published variant of this zine formatted in Typst, along with visual assets, and precompiled PDF's can be found on archive.org at https://archive.org/details/anarchy-tech-support and the the onionsite "eternal hosting" at http://eternalcbrzpicytj4zyguygpmkjlkddxob7tptlr25cdipe5svyqoqd.onion/file/353d829edcf008/dda194e6edce4403.zip