Note: I put this in zx-s[nt]arks because it is related to anonymity and I couldn’t find a better section.

Assumptions for this discussion:

1. There exists a design of a semi-centralized blockchain that does not require trust in the centralized operator (imagine PoA with some mechanism for censorship resistance).
2. There is a reasonable expectation that state actors will actively search for the operator at some point in the future.
3. The project is large/complicated enough that multiple developers are necessary to complete it.

I would like to discuss what good operational security would be both for the development team and the network operator that protects them from an aggressive state actor, without having to rely on some other state protecting them.

The naïve solution is to simply do everything out in the open (on GitHub) but do so under a pseudonym and use a VPN anytime the developers are working on the system. This may work “well enough” for some time, but eventually the GitHub repository will likely be taken offline, and it is impossible to find a VPN that is provably resilient to state pressure. Also, being a network operator is hard because while you can purchase some servers/services online anonymously, the options are somewhat limited and you’ll likely lose your DNS record(s) at some point, and your servers will have to continually move around as they get taken down and you spin them back up.

Onion routing (TOR) is interesting, and it somewhat addresses the DNS takedown issue but network analysis makes it not terribly good at anonymizing traffic. I2P seems to be a nice upcoming technology for anonymizing network traffic that replaces TOR, and it has a very well documented threat model but t is still young and it is unclear if it has a big enough userbase to be secure against state attacks. Interestingly, I2P uses Monotone for their source control to try to address state attacks against developers.

One of the harder parts about defending against state actors is that one needs to address the concerns of attacks via software updates to tools being used. For example, if I2P and Monotone was being used for anonymization then the developers/operators need to make sure that I2P/Monotone updates don’t contain malicious code. This problem quickly escalates to the point where the dev team/operator is spending most of their time just doing code reviews of third party projects rather than actually getting anything done. Are there any resources or tools for trustless code reviews or a network of interested parties that distribute code reviews and throw up red flags when something is amiss?

The reason I bring this up is because as I watch the crypto-currency ecosystem evolve I continue to see companies throwing more and more resources and wasting more and more time discussing how to protect themselves from increasingly aggressive state actors. I fear/suspect that it won’t be too long before we see regulatory agencies around the world change the rules to make working on/operating a crypto-currency illegal, at which point many projects will simply disappear because they don’t have the infrastructure to work anonymously. Recently in the Ethereum EIP process we have witnessed people dropping out of participation in fear of legal repercussions. While I don’t necessarily agree/disagree with that course of action I think it is representative of the increasingly hostile regulatory environment that everyone feels

coming.