Objective

To achieve trustless Ethereum clients or countering the[trusting trust attack](.

This topic of deterministic builds has been addressed in the TOR and Bitcoin community, however, I could not find many resources regarding Ethereum clients, except a few marginal discussions on GitHub and Gitter… I hope the discussion has its place here, otherwise I will move it elsewhere.

Description:

In a trustless blockchain you should not need to trust anything but the source code. However, a tampered compiler/toolchain/package-manager could maliciously modify the builds, similar to the effect malware or a trojan would have on the client.

Exemplified steps for an attack

1. An adversary tampers compilers/toolchains which then get distributed. The compilers would be modified to behave abnormally when they detect the targeted blockchain codebase.
2. Validators/miners compile the latest client from perfectly clean source code with the tampered compiler/toolchain.
3. The modified client then could proceed undetected until enough clients are infected to have an impact. This could be checked by adding a bit somewhere on the RLPx/Libp2p level. Maybe there are more stealthy ways, or the attacker could just wait to increase chances of a successful attack.
4. After activation the consensus or past state could be attacked in numerous ways.

Potential counter measures:

- Goal: achieve deterministic build: Create bit-identical binaries from the program source code
- Gitian (vulnerable toolchain)
- Guix / mes (less dependencies, less trust required)
- Diverse Double-Compiling
- Many clients reduces risk (Check ✓)

Keywords

: deterministic builds, reproducible-builds, GNU Guix / Mes, Toolchain, Gitian, trusting trust attack (Thompson attack)