Reproducible Builds

Reproducible builds are a set of software development practices that create an independently-verifiable path from source to binary code. (more)

Home | News | Documentation | Tools | Who is involved? | Talks | Events | Continuous tests | Contribute

Why does it matter?

Whilst anyone may inspect the source code of free and open source software for malicious flaws, most software is distributed pre-compiled with no method to confirm whether they correspond.

This incentivises attacks on developers who release software, not only via traditional exploitation, but also in the forms of political influence, blackmail or even threats of violence.

This is particularly a concern for developers collaborating on privacy or security software: attacking these typically result in compromising particularly politically-sensitive targets such as dissidents, journalists and whistleblowers, as well as anyone wishing to communicate securely under a repressive regime.

Whilst individual developers are a natural target, it additionally encourages attacks on build infrastructure as a successful attack would provide access to a large number of downstream computer systems. By modifying the generated binaries here instead of modifying the upstream source code, illicit changes are essentially invisible to its original authors and users alike.

The motivation behind the **Reproducible Builds** project is therefore to allow verification that no vulnerabilities or backdoors have been introduced during this compilation process. By promising identical results are always generated from a given source, this allows multiple third parties to come to a consensus on a “correct” result, highlighting any deviations as suspect and worthy of scrutiny.

This ability to notice if a developer or build system has been compromised then prevents such threats or attacks occurring in the first place, as any compromise can be quickly detected. As a result, front-liners cannot be threatened/coerced into exploiting or exposing their colleagues.

Several free software projects already, or will soon, provide reproducible builds.

How?

First, the **build system** needs to be made entirely deterministic: transforming a given source must always create the same result. For example, the current date and time must not be recorded and output always has to be written in the same order.

Second, the set of tools used to perform the build and more generally the **build environment** should either be recorded or pre-defined.

Third, users should be given a way to recreate a close enough build environment, perform the build process, and **validate** that the output matches the original build.

Learn more about how to make your software build reproducibly…

Recent monthly reports

Feb 5, 2022: Reproducible Builds in January 2022
Jan 5, 2022: Reproducible Builds in December 2021
Dec 5, 2021: Reproducible Builds in November 2021

(See all reports…)

Recent news

Apr 6, 2021: Supporter spotlight: Ford Foundation
Oct 26, 2020: Second Reproducible Builds IRC meeting
Oct 21, 2020: Supporter spotlight: Civil Infrastructure Platform

(See all…)

Sponsors

We are proud to be sponsored by:

Google Open Source Security Team

Home News Documentation Tools Who is involved? Talks Events
Continuous tests Contribute

software freedom conservancy

Reproducible Builds