Verifying Software Freedom with Reproducible Builds

Vagrant Cascadian

LibrePlanet 2017-03-25

Goals

The Reproducible Builds project aims to bring us closer to a world where binary software can be independently verified as the result of building the provided source code.

Scientific Methods

Reproducibility is the ability of an entire experiment or study to be duplicated, either by the same researcher or by someone else working independently.

`https://en.wikipedia.org/wiki/Reproducibility`

Source Code

- Source code is readable and writeable by trained ~~monkeys~~ humans

```
for banana in bananas:
  eat(banana)
```

Binary code

- Computers run binary code

01100110 01101111 01110010 00100000 00100000 00100000
00100000 00100000 00100000 00100000 00100000 00100000
00100000 01100010 01100001 01101110 01100001 01101110
01100001 00100000 00100000 00100000 00100000 00100000
00100000 00100000 00100000 00100000 00100000 01101001
01101110 00100000 00100000 00100000 00100000 00100000
00100000 00100000 00100000 00100000 00100000 01100010
01100001 01101110 01100001 01101110 01100001 01110011
00111010 00001010 01100101 01100001 01110100 00101000
01100010 01100001 01101110 01100001 01101110 01100001
00101001

From Source to Binary

- How do you know the binary code the computer is running was produced from the source code? Can you Prove it?

  Computer!
  I would like a bunch of bananas for breakfast, please.

Ooooh, Math(s)!

```
$ python -c 'x=1 ; y=1 ; print(x+y)'
2

$ python -c 'x=1 ; y=1 ; print(x+y)' | sha256sum
53c234e5e8472b6ac...8977b010655bfdd3c3  -

$ echo 2 | sha256sum
53c234e5e8472b6ac...8977b010655bfdd3c3  -
```

But software building is more like. . .

x=source code
y=build instructions
z=toolchain (compiler, linker, libraries, etc.)
r=other stuff (time of build, running OS, username building software, environment variables, etc.)
$x + y + z + r = ?$

Independent verification

source code + build instructions + toolchain
=
bit-by-bit identical copies
anyone can verify the result
`https://reproducible-builds.org/docs/definition/`

Trusting Trust

Reflections on Trusting Trust by Ken Thompson 1984

- https://www.ece.cmu.edu/~ganger/712.fall02/papers/p761-thompson.pdf

Diverse Double-Compilation

Diverse Double-Compilation by David A. Wheeler 2005/2009

- `https://www.dwheeler.com/trusting-trust/`

Reproducibility matters

What kind of security implications are we facing?

- CVE-2002-0083: Remote root exploit in OpenSSH, caused by an off-by-one error
- 2015: XcodeGhost: malware variant of Apple's SDK Infected over 4,000 apps in Apple's App store

Freedoms: Use

Freedom

to

use

Freedoms: Study

Freedom

to

study

Freedoms: Improve

Freedom

to

improve

Freedoms: Share

Freedom

to

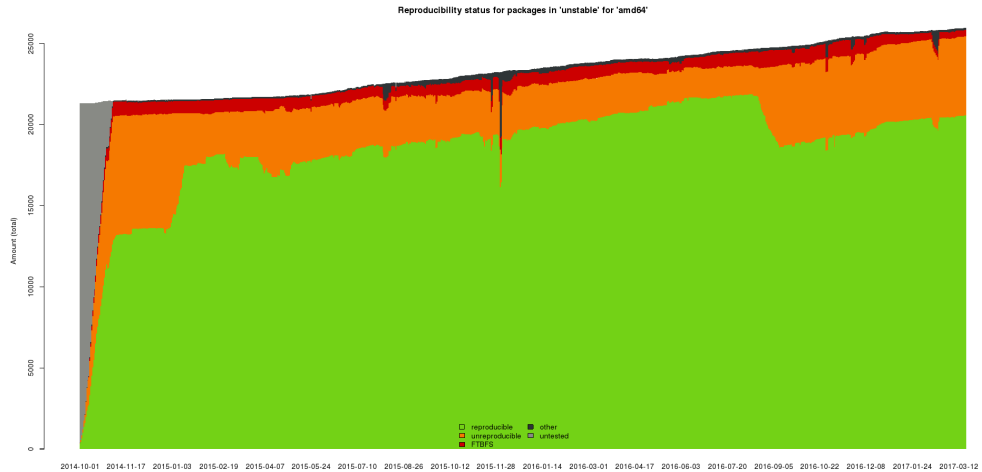
share

History in Debian

- Mentioned on lists as early as 2007
- Automated rebuilding of Debian's 25,000+ source packages began in late 2014
- Currently rebuilding roughly 1,600-2,200 packages a day on each of amd64, i386, arm64 and armhf

A plague of unreproducibility

- About 4,900 (19%) of software in Debian unstable
- About 1,300 (5%) of software in Debian testing
- Patches in Debian toolchains and packages, but patches are swimming upstream



timestamps

- Embedded timestamps:

U-Boot SPL 2016.01+dfsg1-3 (Feb 21 2016 - 21:39:10)

timestamps: Please No

- There's no timestamps like NO timestamps.

timestamps: SOURCE_DATE_EPOCH

- If you really must, use the SOURCE_DATE_EPOCH specification, which specifies the timestamp to use in a standardized environment variable.

`https://reproducible-builds.org/specs/source-date-epoch/`

Other Common problems

- timezone
- file sort order
- locales
- build path

Other projects

It goes well beyond Debian:
`https://reproducible-builds.org/who/`

- NixOS
- GNU Guix
- Fedora
- OpenSUSE
- FreeBSD
- Arch Linux
- Tails
- Bitcoin
- Coreboot
- Tor Browser
- And more...

Future dreams

- Give users a way to only install reproducible software
- Make reproducible builds standard practice for Free Software distributions

Thanks

- Core Infrastructure Initiative
- Profitbricks
- Codethink

All the great folks doing reproducible builds work!

Copyright