Two Ways to Trustworthy

Vagrant Cascadian <vagrant@reproducible-builds.org>

SeaGL 2024-11-08

Who am I

	Vagrant
debian user	2001
debian developer	2010
reproducible builds	2015
guix contributor	2017
guix committer	2019

Trustworthy vs. Trust

Trustworthy projects require little Trust

- ...

Trustworthy vs. Trust

Trustworthy projects require little Trust

- ...
- Auditable

Trustworthy vs. Trust

Trustworthy projects require little Trust

- ...
- Auditable
- Verifiable

Trustworthy vs. Trust

Trustworthy projects require little Trust

- ...
- Auditable
- Verifiable
- Trust should be minimized

Archaic and Relative time

Debian 1.5 score and a year ago
Guix 0.5 score and a couple years ago

Specified time

Debian founded 1993
Guix founded 2012

A Human Generation apart

What happened in those nineteen years between 1993 and 2012?

- ...

A Human Generation apart

What happened in those nineteen years between 1993 and 2012?

- ...
- massive adoption of free and open source technologies

A Human Generation apart

What happened in those nineteen years between 1993 and 2012?

- ...
- massive adoption of free and open source technologies
- plethora of software distributions

A Human Generation apart

What happened in those nineteen years between 1993 and 2012?

- ...
- massive adoption of free and open source technologies
- plethora of software distributions
- version control became a common best practice

Variably Crude, Sisyphus

Debian

- ...

Variably Crude, Sisyphus

Debian

- ...
- predates modern version control

Variably Crude, Sisyphus

Debian

- ...
- predates modern version control
- one repository per package

Variably Crude, Sisyphus

Debian

- ...
- predates modern version control
- one repository per package
- in a variety of packaging formats (source included, debian/ dir only, etc.)

Variably Crude, Sisyphus

Debian

- ...
- predates modern version control
- one repository per package
- in a variety of packaging formats (source included, debian/ dir only, etc.)
- not all on debian infrastructure

Variably Crude, Sisyphus

Debian

- ...
- predates modern version control
- one repository per package
- in a variety of packaging formats (source included, debian/ dir only, etc.)
- not all on debian infrastructure
- not all packages use version control

Variably Crude, Sisyphus

Debian

- ...
- predates modern version control
- one repository per package
- in a variety of packaging formats (source included, debian/ dir only, etc.)
- not all on debian infrastructure
- not all packages use version control
- debian archive itself a crude sort of version control

Variably Crude, Sisyphus

Guix

- ...

Variably Crude, Sisyphus

Guix

- ...
- git was dominant VCS from the beginning

Variably Crude, Sisyphus

Guix

- ...
- git was dominant VCS from the beginning
- all of guix in a single shared git repository

Variably Crude, Sisyphus

Guix

- ...
- git was dominant VCS from the beginning
- all of guix in a single shared git repository
- git only contains references to other archives of software

Variably Crude, Sisyphus

Guix

- ...
- git was dominant VCS from the beginning
- all of guix in a single shared git repository
- git only contains references to other archives of software
- full source code mostly on infrastructure outside of guix

Reproducible Builds

`https://reproducible-builds.org/docs/definition/`

A build is reproducible if given the same source code, build environment and build instructions, any party can recreate bit-by-bit identical copies of all specified artifacts.

Binaries at Large

Debian is fundamentally a binary distribution

- ...

Binaries at Large

Debian is fundamentally a binary distribution

- ...
- relies on ABI for when to rebuild packages

Binaries at Large

Debian is fundamentally a binary distribution

- ...
- relies on ABI for when to rebuild packages
- a given package is built with a specific version

Binaries at Large

Debian is fundamentally a binary distribution

- ...
- relies on ABI for when to rebuild packages
- a given package is built with a specific version
- other packages might use an ABI compatible version

Binaries at Large

Debian is fundamentally a binary distribution

- ...
- relies on ABI for when to rebuild packages
- a given package is built with a specific version
- other packages might use an ABI compatible version
- rebuilding all of debian today would result in many different pacakges

The Source With Optimizations

Guix is a fundamentally a source distribution

- ...

The Source With Optimizations

Guix is a fundamentally a source distribution

- ...
- packages optionally pull from substitute server as needed, falling back to building from source

The Source With Optimizations

Guix is a fundamentally a source distribution

- ...
- packages optionally pull from substitute server as needed, falling back to building from source
- packages get rebuilt whenever their dependencies change

The Source With Optimizations

Guix is a fundamentally a source distribution

- ...
- packages optionally pull from substitute server as needed, falling back to building from source
- packages get rebuilt whenever their dependencies change
- a given guix git commit builds exactly one set of packages

Correctly annoying resources

Guix is correct to rebuild, although a bit annoying and resource intensive
Makes it difficult to accurately track reproducibility stats

Correct enough and less annoying

Debian is mostly correct to rely on ABI, and less annoying and resource intensive
Fewer rebuilds makes it easier to track reproducibility stats

Reproducible Builds: Debian

Reproducible Builds in Debian

- ...

Reproducible Builds: Debian

Reproducible Builds in Debian

- ...
- Record the build environment (.buildinfo)

Reproducible Builds: Debian

Reproducible Builds in Debian

- ...
- Record the build environment (.buildinfo)
- Recreate the build environment

Reproducible Builds: Debian

Reproducible Builds in Debian

- ...
- Record the build environment (.buildinfo)
- Recreate the build environment
- Perform the build!

Reproducible Builds: Debian

Reproducible Builds in Debian

- ...
- Record the build environment (.buildinfo)
- Recreate the build environment
- Perform the build!
- Compare against the original

Example .buildinfo

```
Source: libtext-simpletable-perl
Version: 2.03-1
Checksums-Sha256:
 7a285...a8b 10788 libtext-simpletable-perl_2.03-1_all.deb
Build-Architecture: amd64
Build-Date: Fri, 03 Mar 2017 07:56:17 +1400
Build-Path: /build/libtext-simpletable-perl-2.03/2nd
Installed-Build-Depends:
 autoconf (= 2.69-10),
 automake (= 1:1.15-6),
 zlib1g (= 1:1.2.8.dfsg-5)
Environment:
 DEB_BUILD_OPTIONS="parallel=15"
 LANG="C"
 LC_ALL="C"
```

Reproducible Builds: Debian

.buildinfo files

- ...

Reproducible Builds: Debian

.buildinfo files

- ...
- Find the correct file: buildinfos.debian.net

Reproducible Builds: Debian

.buildinfo files

- ...
- Find the correct file: buildinfos.debian.net
- Download the correct set of packages: snapshot.debian.org

Reproducible Builds: Debian

.buildinfo files

- ...
- Find the correct file: buildinfos.debian.net
- Download the correct set of packages: snapshot.debian.org
- Recreate the build environment

The challenges of Reproducible Builds in Debian

Debian Challenges

- ...

The challenges of Reproducible Builds in Debian

Debian Challenges

- ...
- corner cases with versioning (epoch, source != binary version)

The challenges of Reproducible Builds in Debian

Debian Challenges

- ...
- corner cases with versioning (epoch, source != binary version)
- snapshot.debian.org may be missing some package versions

The challenges of Reproducible Builds in Debian

Debian Challenges

- ...
- corner cases with versioning (epoch, source != binary version)
- snapshot.debian.org may be missing some package versions
- snapshot.debian.org was sometimes to slow to use

Reproducible Builds: Guix

Guix

- Record the commit hash of guix git

Reproducible Builds: Guix

Guix

- Record the commit hash of guix git
- binary substitutes may not be available

Reproducible Builds: Guix

Guix

- Record the commit hash of guix git
- binary substitutes may not be available
- sources may not be available

The challenge of Guix

```
guix challenge --verbose PACKAGE
```

Compares against local builds and substitute servers.

The challenge of Guix

```
guix challenge --verbose --diff=diffoscope PACKAGE
```

Compares differences using diffoscope!

Checking in with guix

```
guix build --check PACKAGE
```

Check again

```
guix build --check --no-grafts --keep-failed PACKAGE

diffoscope /gnu/store/...PACKAGE/ /gnu/store/...PACKAGE-check/
```

So many ways in Debian

Debian has many, many options...

- dpkg-buildpackage

So many ways in Debian

Debian has many, many options...

- dpkg-buildpackage
- debuild

So many ways in Debian

Debian has many, many options...

- dpkg-buildpackage
- debuild
- sbuild

So many ways in Debian

Debian has many, many options...

- dpkg-buildpackage
- debuild
- sbuild
- pbuilder

So many ways in Debian

Debian has many, many options...

- dpkg-buildpackage
- debuild
- sbuild
- pbuilder
- etc.

Rebuilding Debian packages

```
debrebuild --builder=mmdebstrap hello_2.10-2_amd64.buildinfo
```

- rebuild a package with a given .buildinfo file

Rebuilding Debian packages

```
debrebuild --builder=mmdebstrap hello_2.10-2_amd64.buildinfo
```

- rebuild a package with a given .buildinfo file
- Leverages snapshot.debian.org to recreate build environment from .buildinfo file

Rebuilding Debian packages

```
debrebuild --builder=mmdebstrap hello_2.10-2_amd64.buildinfo
```

- rebuild a package with a given .buildinfo file
- Leverages snapshot.debian.org to recreate build environment from .buildinfo file
- Active work in progress

Troubleshooting Debian packages

`debrepro`

- build the package in the current working directory

Troubleshooting Debian packages

`debrepro`

- build the package in the current working directory
- no chroot or isolation

Troubleshooting Debian packages

`debrepro`

- build the package in the current working directory
- no chroot or isolation
- basic reproducibility check with some variations

Troubleshooting Debian packages second edition

```
reprotest --verbose --vary=-fileordering,-domain_host,-user_group,-time auto --
```

- reproducibility build checks with extensive variations

Troubleshooting Debian packages second edition

```
reprotest --verbose --vary=-fileordering,-domain_host,-user_group,-time auto --
```

- reproducibility build checks with extensive variations
- flexibly define which variations to use

Troubleshooting Debian packages second edition

```
reprotest --verbose --vary=-fileordering,-domain_host,-user_group,-time auto --
```

- reproducibility build checks with extensive variations
- flexibly define which variations to use
- options to "bisect" to identify nature of reproducibility issue

Troubleshooting Debian packages second edition

```
reprotest --verbose --vary=-fileordering,-domain_host,-user_group,-time auto --
```

- reproducibility build checks with extensive variations
- flexibly define which variations to use
- options to "bisect" to identify nature of reproducibility issue
- needs maintenence and updating

Troubleshooting Guix packages

Guix troubleshooting

`guix build --keep-failed PACKAGE`

Keeps the source tree to re-try the build manually, perhaps with...

Imperfect troubleshooting

```
guix shell --container --development PACKAGE
```

Which produces a very similar environment to the build, but not the same...
Pretty much stuck iterating build after build entirely from source, partial builds are not really possible.

Bootstrapping: Debian

A simple Debian chroot is 522MB of binaries.
The original sources used to build the original binaries have been lost to time.

Bootstrapping: Guix

Guix is bootstrapped from 357 byte hex binary (and 25MB of static binaries)
`https://guix.gnu.org/en/blog/2023/the-full-source-bootstrap-building-from-source-all-the-way-down/`
The historic versions and their hashes are recorded in guix from very early on.

Thanks

Help make it happen!
https://reproducible-builds.org/contribute/
https://reproducible-builds.org/donate/
https://reproducible-builds.org/who/sponsors

Copyright and attributions