Reproducible builds everywhere
eg. in Debian, OpenWrt and LEDE

Bit by bit identical binaries
from a given source

Alexander ’lynxis’ Couzens
Holger ’h01ger’ Levsen

OpenWrt Summit in Berlin, Germany
2016-10-13
about h01ger

- B8BF 5413 7B09 D35C F026 FE9D 091A B856 069A AA1C
- Debian user since 1995
- Debian contributor since 2001
- OpenWrt user since 2006
- Debian developer since 2007
- DebConf organizer, founded the DebConf video team
  - http://video.debian.net
- Debian-Edu (Debian for education)
- Debian QA (quality assurance)
  - https://piuparts.debian.org
  - https://jenkins.debian.net (1200 jobs continuously testing Debian)
- Debian Reproducible builds team member
  - since April 2015 funded by the Linux Foundation
about lynxis

- 390D CF78 8BF9 AA50 4F8F F1E2 C29E 9DA6 A0DF 8604
- Debian user since 2003
- OpenWrt user since 2006
- LEDE founding member
- coreboot hacker
- tests.reproducible-builds.org contributor
- CCC member
In this talk we’ll ignore the distinction between the two:
- when we say ”OpenWrt” me mean ”LEDE and OpenWrt”,
- when we say ”LEDE” me mean ”OpenWrt and LEDE”,
- when we say ”OpenWrt and LEDE” we mean ”LEDE and OpenWrt”.
about OpenWrt and LEDE

In this talk we’ll ignore the distinction between the two:
- when we say ”OpenWrt” me mean ”LEDE and OpenWrt”,
- when we say ”LEDE” me mean ”OpenWrt and LEDE”,
- when we say ”OpenWrt and LEDE” we mean ”LEDE and OpenWrt”.
- They are two projects though and when there are differences we’ll mention them.

Reproducible builds everywhere

OpenWrt Summit, Berlin
Who are you?
Who are you?

- Seen a talk about reproducible builds?
Who are you?

- Seen a talk about reproducible builds?
- Contributed to the effort?
Who are you?

- Seen a talk about reproducible builds?
- Contributed to the effort?
- Uses Debian or a Debian based system?
Debian reproducible builds team

akira
Alexis Bienvenüe
Andrew Ayer
Asheesh Laroia
Ceridwen
Chris Lamb
Chris West
Christoph Berg
Daniel Kahn Gillmor
Daniel Shahaf
David Suarez
Dhole
Drew Fisher
Emmanuel Bourg
Emanuel Bronshtein
Esa Peuha
Fabian Wolff
Guillem Jover
Hans-Christoph Steiner
Helmut Grohne
Holger Levensen
HW42
Intrigeri
Jelmer Vernooij
josch
Juan Picca
Lunar
Mathieu Bridon
Mattia Rizzolo
Nicolas Boulenguez
Niels Thykier
Niko Tyni
Paul Wise
Peter De Wachter
Philip Rinn
Reiner Herrmann
Santiago Vila
Sascha Steinbiss
Satyam Zode
Scarlett Clark
Stefano Rivera
Stéphane Glondu
Steven Chamberlain
Tom Fitzhenry
Valerie Young
Valentin Lorentz
Wookey
Ximin Luo
Debian reproducible builds team

akira
Alexis Bienvenüe
Andrew Ayer
Asheesh Laroia
Ceridwen
Chris Lamb
Chris West
Christoph Berg
Daniel Kahn Gillmor
Daniel Shahaf
David Suarez
Dhole
Drew Fisher
Emmanuel Bourg
Emanuel Bronshtein
Esa Peuha
Fabian Wolff
Guillem Jover
Hans-Christoph Steiner
Helmut Grohne
Holger Levsen
HW42
Intrigeri
Jelmer Vernooij
josch
Juan Picca
Lunar
Mathieu Bridon
Mattia Rizzolo
Nicolas Boulenguez
Niels Thykier
Niko Tyni
Paul Wise
Peter De Wachter
Philip Rinn
Reiner Herrmann
Santiago Vila
Sascha Steinbiss
Satyam Zode
Scarlett Clark
Stefano Rivera
Stéphane Glondu
Steven Chamberlain
Tom Fitzhenry
Valerie Young
Valentin Lorentz
Wookey
Ximin Luo
jenkins.debian.net.git contributors

akira
Alexander Couzens
Levente 'anthraxx' Polyak
Antonio Terceiro
Axel Beckert
Bryan Newbold
Chris Lamb
Daniel Kahn Gillmor
Gabriele Giacone
Hans-Christoph Steiner
Helmut Grohne
Holger Levensen
HW42
James McCoy
Joachim Breitner

Johannes 'josch' Schauer
Jérémy Bobbio
Mattia Rizzolo
Niels Thykier
Paul Wise
Petter Reinholdtsen
Philip Hands
Reiner Herrmann
Samuel Thibault
Steven Chamberlain
Tails developers
Ulrike Uhlig
Wolfgang Schweer
Wouter Verhelst

OpenWrt Summit, Berlin
jenkins.debian.net.git contributors

akira
Alexander Couzens
Levente 'anthraxx' Polyak
Antonio Terceiro
Axel Beckert
Bryan Newbold
Chris Lamb
Daniel Kahn Gillmor
Gabriele Giacone
Hans-Christoph Steiner
Helmut Grohne
Holger Levsen
HW42
James McCoy
Joachim Breitner

Johannes 'josch' Schauer
Jérémy Bobbio
Mattia Rizzolo
Niels Thykier
Paul Wise
Petter Reinholdtsen
Philip Hands
Reiner Herrmann
Samuel Thibault
Steven Chamberlain
Tails developers
Ulrike Uhlig
Wolfgang Schweer
Wouter Verhelst
Motivation

Common ressources

Status Debian

Status Non-Debian World

Future work

Getting involved

Questions, comments, ideas?
The problem

Available on media.ccc.de, 31c3
A few examples from that 31c3 talk

- CVE-2002-0083: remote root exploit in `sshd`, a single bit difference in the binary
A few examples from that 31c3 talk

- CVE-2002-0083: remote root exploit in `sshd`, a single bit difference in the binary
- 31c3 talk had a live demo with a kernel module modifying source code in memory only
A few examples from that 31c3 talk

- CVE-2002-0083: remote root exploit in `sshd`, a single bit difference in the binary
- 31c3 talk had a live demo with a kernel module modifying source code in memory only
- How can you be sure what’s running on your machine or on a build daemon network connected to the net? Do you ever leave your computers physically alone?
A few examples from that 31c3 talk

- CVE-2002-0083: remote root exploit in `sshd`, a single bit difference in the binary
- 31c3 talk had a live demo with a kernel module modifying source code in memory only
- How can you be sure what’s running on your machine or on a build daemon network connected to the net? Do you ever leave your computers physically alone?
- Huge financial incentives to crack developer machines or a project’s build infrastructure…
A few examples from that 31c3 talk

- CVE-2002-0083: remote root exploit in `sshd`, a single bit difference in the binary
- 31c3 talk had a live demo with a kernel module modifying source code in memory only
- How can you be sure what’s running on your machine or on a build daemon network connected to the net? Do you ever leave your computers physically alone?
- Huge financial incentives to crack developer machines or a project’s build infrastructure...
Another example from real life

At a CIA conference in 2012:

[edit] (S//NF) Strawhorse: Attacking the MacOS and iOS Software Development Kit

(S) Presenter: [Redacted] Sandia National Laboratories

(S//NF) Ken Thompson’s gcc attack (described in his 1984 Turing award acceptance speech) motivates the StrawMan work: what can be done of benefit to the US Intelligence Community (IC) if one can make an arbitrary modification to a system compiler or Software Development Kit (SDK)? A (whacked) SDK can provide a subtle injection vector onto standalone developer networks, or it can modify any binary compiled by that SDK. In the past, we have watermarked binaries for attribution, used binaries as an exfiltration mechanism, and inserted Trojans into compiled binaries.

(S//NF) In this talk, we discuss our explorations of the Xcode (4.1) SDK. Xcode is used to compile MacOS X applications and kernel extensions as well as iOS applications. We describe how we use (our whacked) Xcode to do the following things: -Entice all MacOS applications to create a remote backdoor on execution -Modify a dynamic dependency of secyrd to load our own library - which rewrites secyrd so that no prompt appears when exporting a developer's private key -Embed the developer's private key in all iOS applications -Force all iOS applications to send embedded data to a listening post -Convince all (new) kernel extensions to disable ASLR

(S//NF) We also describe how we modified both the MacOS X updater to install an extra kernel extension (a keylogger) and the Xcode installer to include our SDK whacks.

firstlook.org/theintercept/2015/03/10/isy-cia-campaign-steal-apples-secrets/
The solution

Promise that anyone can always generate identical binary packages from a given source
The solution

We call this:

“Reproducible builds”

h01ger and lynxis

OpenWrt Summit, Berlin 13 / 52
Debian demo (skipped)

- Build a package 5 times, get 5 .debs with different checksums
- Build a package 5 times, get 5 .debs with the same checksum
Debian demo (skipped)

- Build a package 5 times, get 5 .debs with different checksums
- Build a package 5 times, get 5 .debs with the same checksum
Yes, it’s really this simple.
This should become the norm.
This should become the norm.

We want to change the meaning of "free software": it's only free software if it’s reproducible!
More benefits than "just" security…

- smaller deltas, thus faster updates possible
- in Debian: lots of QA benefits
- Google does reproducible builds, to save money
- …
Motivation

Common resources

Status Debian

Status Non-Debian World

Future work

Getting involved

Questions, comments, ideas?
reproducible-builds.org

- https://reproducible-builds.org
- git repositories, IRC channels, mailinglists, webspace

What is it about?

**Reproducible builds** are a set of software development practices which create a *verifiable path* from human-readable *source code* to the *binary* code used by computers.

Why does it matter?

Most aspect of software verification is done on source code, as that is what humans can reasonably understand. But most of the time, computers require software to be first built
Debugging problems:
https://try.diffoscope.org

- Examines differences **in depth**.
- Recursively unpacks archives, uncompresses PDFs, disassembles binaries, unpacks Gettext files, ...
- Easy to extend to new file formats.
- Falls back to binary comparison.
- Outputs HTML or plain text with human readable differences.
- Available from git, PyPI, Debian, Arch Linux, Guix, Homebrew. Works on BSD.
- Maintainers in other distros wanted.
- https://diffoscope.org/
### install.rdf

<table>
<thead>
<tr>
<th>Offset 5, 15 lines modified</th>
<th>Offset 5, 15 lines modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>... &lt;Description about=&quot;urn:mozilla:install-manifest&quot;&gt; ... &lt;Description about=&quot;urn:mozilla:install-manifest&quot;&gt;</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>... <a href="">em:name</a>HTTPS-Everywhere&lt;/em:name&gt; ... <a href="">em:name</a>HTTPS-Everywhere&lt;/em:name&gt;</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>7</td>
</tr>
<tr>
<td>... <a href="">em:creator</a>Mike Perry, Peter Eckersley, &amp; Yan Zhu&lt;/em:creator&gt; ... <a href="">em:creator</a>Mike Perry, Peter Eckersley, &amp; Yan Zhu&lt;/em:creator&gt;</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>... <a href="">em:aboutURL</a>chrome://https-everywhere/content/about.xul&lt;/em:aboutURL&gt; ... <a href="">em:aboutURL</a>chrome://https-everywhere/content/about.xul&lt;/em:aboutURL&gt;</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>9</td>
</tr>
<tr>
<td>... <a href="">em:id</a><a href="mailto:https-everywhere@eff.org">https-everywhere@eff.org</a>&lt;/em:id&gt; ... <a href="">em:id</a><a href="mailto:https-everywhere@eff.org">https-everywhere@eff.org</a>&lt;/em:id&gt;</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Extension --&gt;</td>
<td>Extension --&gt;</td>
</tr>
<tr>
<td>... <a href="">em:extension</a>Encrypt the Web!&lt;/em:extension&gt; ... <a href="">em:extension</a>Encrypt the Web!&lt;/em:extension&gt;</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>11</td>
</tr>
<tr>
<td>Automatically use HTTPS security on many sites.</td>
<td>Automatically use HTTPS security on many sites.</td>
</tr>
<tr>
<td><a href="">em:description</a></td>
<td><a href="">em:description</a></td>
</tr>
<tr>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>... <a href="">em:version</a>5.0.6&lt;/em:version&gt; ... <a href="">em:version</a>5.0.7&lt;/em:version&gt;</td>
<td></td>
</tr>
</tbody>
</table>
diffoscope is "just" for debugging

- Reminder: diffoscope is for **debugging**
- "reproducible" according to our definition means: **bit by bit identical**. So the tools for testing whether something is reproducible are either **diff** or **sha256sum**!
**diffoscope** is "just" for debugging

- Reminder: **diffoscope** is for **debugging**
- "reproducible" according to our definition means: **bit by bit identical**. So the tools for testing whether something is reproducible are either diff or sha256sum!
- [https://try.diffoscope.org](https://try.diffoscope.org)
tests.reproducible-builds.org

- Continuously testing Debian testing, unstable and experimental
- Also testing: coreboot, OpenWrt, LEDE, NetBSD, FreeBSD, Arch Linux, Fedora and soon F-Droid too
- 8-12 amd64 nodes, 150 cores and soon 500 GB RAM - thanks to Profitbricks.com!
- 22 armhf nodes, 98 cores and 53 GB RAM
- 329 jenkins jobs running on jenkins.debian.net
- 43 scripts in Python and Bash, 283 lines of code in average
- 37 contributors for jenkins.debian.net.git
## Variations (when testing Debian)

<table>
<thead>
<tr>
<th>variation</th>
<th>first build</th>
<th>second build</th>
</tr>
</thead>
<tbody>
<tr>
<td>hostname</td>
<td>jenkins</td>
<td>i-capture-the-hostname</td>
</tr>
<tr>
<td>domainname</td>
<td>debian.net</td>
<td>i-capture-the-domainname</td>
</tr>
<tr>
<td>env TZ</td>
<td>GMT+12</td>
<td>GMT-14</td>
</tr>
<tr>
<td>env LANG</td>
<td>C</td>
<td>fr_CH.UTF-8</td>
</tr>
<tr>
<td>env LC_ALL</td>
<td>not set</td>
<td>fr_CH.UTF-8</td>
</tr>
<tr>
<td>env USER</td>
<td>pbuilder1</td>
<td>pbuilder2</td>
</tr>
<tr>
<td>uid</td>
<td>1111</td>
<td>2222</td>
</tr>
<tr>
<td>gid</td>
<td>1111</td>
<td>2222</td>
</tr>
<tr>
<td>UTS namespace</td>
<td>shared with the host</td>
<td>modified using /usr/bin/unshare --uts</td>
</tr>
<tr>
<td>kernel version</td>
<td>Linux 3.16 or 4.X</td>
<td>on amd64 always varied, on armhf sometimes</td>
</tr>
<tr>
<td>umask</td>
<td>0022</td>
<td>0002</td>
</tr>
<tr>
<td>CPU type</td>
<td>varied on i386</td>
<td></td>
</tr>
<tr>
<td></td>
<td>on armhf varied a bit, not on amd64</td>
<td></td>
</tr>
<tr>
<td>filesystem</td>
<td>same for both builds on amd64: (tmpfs), on armhf ext3/4</td>
<td>(and we have disorderfs, but the code is disabled)</td>
</tr>
<tr>
<td>year, month, date</td>
<td>on amd64: 398 days variation, on armhf not yet</td>
<td></td>
</tr>
<tr>
<td>hour, minute</td>
<td>hour is usually the same… usually, the minute differs…</td>
<td>is likely the same…</td>
</tr>
</tbody>
</table>

---

*Reproducible builds everywhere*

*OpenWrt Summit, Berlin*
Common problems

- time stamps
- time zones
- locales
- build paths
- everything else (separated into known issues and the blurry rest)
Documentation about common problems

- https://reproducible-builds.org/docs
- Lunar’s talk from CCCamp 2015 also on https://media.ccc.de

Avoid (true) randomness

- Randomness is not deterministic

Example

```
$ gcc -flto -c utils.c
$ nm -a utils.o | grep inline
0000000000000000 n.gnu.lto_.inline.381a277a0b6d2a35
```
**SOURCE_DATE_EPOCH**

- Build date (timestamps) usually not useful for the user
- **SOURCE_DATE_EPOCH** is defined as the last modification of the source, since the epoch (1970-01-01)
- can be used instead of current date
- can also be used for random seeds etc.
- in Debian, set from the latest debian/changelog entry
- can be set to the latest git commit too or the latest file modification date
source_date_epoch

- **SOURCE_DATE_EPOCH** spec available:
- https://reproducible-builds.org/specs/
- many upstreams support it already
- has been adopted by other distributions (OpenWrt, LEDE, NetBSD, FreeBSD, Arch Linux, coreboot, Guix, …) and many many upstreams (GCC, dpkg, rpm, mkirosfs, ghostscript, libxml, sphinx, texlive-bin, …)
Progress in Debian testing ("stretch")

21,527 (91.2%) out of 23,597 source packages are reproducible in our test framework on amd64
18,898 (75.8%) out of 24,931 source packages are reproducible in our test framework on amd64 (difference due to build path variations)
Progress in the Debian bug tracker

As a rule, we file bugs with patches. There are very few exceptions.
Details on tests.reproducible-builds.org

- https://reproducible.debian.net/$src
- 43 package sets
- 250 categorised distinct issues
- 6,944 notes
- 1,894 unreproducible packages in stretch (testing), but only 177 without a note (5,777 in unstable but also only 277 without a note)
- maintained in notes.git by 47 contributors
- currently Debian only, but cross distro notes are planned
Summary / What’s left to do

- This is a proof-of-concept, Debian is neither 91.2% reproducible nor 75.8%. (and 10% > 2,300 sources packages!)
Summary / What’s left to do

- This is a proof-of-concept, Debian is neither 91.2% reproducible nor 75.8%. (and 10% > 2,300 sources packages!)
- All our required changes are finally in Debian now, except dpkg and .buildinfo file support on the archive side.

We hope that Debian 9, “stretch”, will be partially reproducible in a meaningful way, in 2017.

What’s beyond (rebuilding, .buildinfo file handling, user tools) still needs design and code.

Will Debian 10, “buster”, be 100% reproducible?
Summary / What’s left to do

- This is a proof-of-concept, Debian is neither 91.2% reproducible nor 75.8%. (and 10% > 2,300 sources packages!)
- All our required changes are finally in Debian now, except dpkg and .buildinfo file support on the archive side.
- We hope that Debian 9, ”stretch”, will be partially reproducible in a meaningful way, in 2017.
- What’s beyond (rebuilding, .buildinfo file handling, user tools) still needs design and code.
Summary / What’s left to do

- This is a proof-of-concept, Debian is neither 91.2% reproducible nor 75.8%. (and 10% > 2,300 sources packages!)
- All our required changes are finally in Debian now, except dpkg and .buildinfo file support on the archive side.
- We hope that Debian 9, ”stretch”, will be partially reproducible in a meaningful way, in 2017.
- What’s beyond (rebuilding, .buildinfo file handling, user tools) still needs design and code.
- Will Debian 10, ”buster”, be 100% reproducible?
Tell the world & collaborate

”We don’t care about Debian (only), we care about free and open source software.”
Tell the world & collaborate

- "We don’t care about Debian (only), we care about free and open source software."
- Weekly reports since May 2015
Tell the world & collaborate

- "We don’t care about Debian (only), we care about free and open source software."
- Weekly reports since May 2015
- First Reproducible World Summit in December 2015 (Athens, Greece)
  - 40 people from 16 projects
  - reproducible.debian.net has become tests.reproducible-builds.org
Tell the world & collaborate

- "We don’t care about Debian (only), we care about free and open source software."
- Weekly reports since May 2015
- First Reproducible World Summit in December 2015 (Athens, Greece)
  - 40 people from 16 projects
  - reproducible.debian.net has become tests.reproducible-builds.org
- Second Reproducible World Summit in December 2016 in Berlin
  - Talk to h01ger if you want to attend.
Skipping some...

- https://tests.r-b.org/coreboot
- https://tests.r-b.org/netbsd
- https://tests.r-b.org/freebsd
- paused: https://tests.r-b.org/archlinux
- paused: https://tests.r-b.org/fedora
- not yet: https://tests.r-b.org/f-droid
Skipping some more…

- Bitcoin (2011)
- Tor (2013)
- NixOS, Guix, ElectroBSD
- Qubes, Tails
- very few commercial, proprietary software (guess where!)
- ?
Skipping some more...

- Bitcoin (2011)
- Tor (2013)
- NixOS, Guix, ElectroBSD
- Qubes, Tails
- very few commercial, proprietary software (gambling machines!)
- ?

h01ger and lynxis
Reproducible builds everywhere
OpenWrt Summit, Berlin
OpenWrt and LEDE tested for reproducible builds

reproducible_(openwrt_common|openwrt|lede).sh scripts in jenkins.debian.net.git

1,073/1,089 packages and 12/1 (OpenWrt/LEDE) images tested each week

variations: TZ, LANG, LC_ALL, PATH, (umask), make -j, linux64 –uname-2.6, CAPTURE_ENVIRONMENT

h01ger and lynxis

Reproducible builds everywhere

OpenWrt Summit, Berlin
OpenWrt and LEDE tested for reproducible builds

- https://tests.r-b.org/openwrt
- https://tests.r-b.org/lede
- reproducible_(openwrt_common|openwrt|lede).sh scripts in jenkins.debian.net.git
- 1,073/1,089 packages and 12/1 (OpenWrt/LEDE) images tested each week
- variations: TZ, LANG, LC_ALL, PATH, (umask), make -j, linux64 --uname-2.6, CAPTURE_ENVIRONMENT
Thanks to these OpenWrt / LEDE reproducible builds contributors

Alexander Couzens
Bryan Newbold
Dirk Neukirchen
Felix Fietkau
Jonas Gorski
Jo-Philipp Wich
Nathan Hintz
Reiner Herrmann
TODO for tests.r-b.org/(openwrt|lede)

- we should add more variations (date, time, build path, hostname, domain, use disorderfs, CPU type, kernel, USER, HOME, SHELL, the base system).
- we should test more targets.
TODO for tests.r-b.org/(openwrt|lede)

- we should add more variations (date, time, build path, hostname, domain, use disorderfs, CPU type, kernel, USER, HOME, SHELL, the base system).
- we should test more targets.
- we could build other branches too...
- we could build OpenWrt + LEDE at least every day, thanks again to Profitbricks.com.
TODO for tests.r-b.org/(openwrt|lede)

- we should add more variations (date, time, build path, hostname, domain, use disorderfs, CPU type, kernel, USER, HOME, SHELL, the base system).
- we should test more targets.
- we could build other branches too...
- we could build OpenWrt + LEDE at least every day, thanks again to Profitbricks.com.
- we want to make you look at these pages every day!
TODO: design `.buildinfo` files for OpenWrt and LEDE

- rfc822 format
- needs to define the environment
- needs to define the sources (input)
- needs to define the binaries (output)
TODO: design `.buildinfo` files for OpenWrt and LEDE

- rfc822 format
- needs to define the environment
- needs to define the sources (input)
- needs to define the binaries (output)
- Debian has only `.deb` files as output, while OpenWrt/LEDE have packages and images...
Rebuilds and sharing signed checksums

- Almost no work has been done here yet. We are just at the first step: being able to rebuild reproducibly…
- Different projects, different solutions?
Rebuilds and sharing signed checksums

- Almost no work has been done here yet. We are just at the first step: being able to rebuild reproducibly...
- Different projects, different solutions?
  - something like `.buildinfo` files (defining the environment, the input and the output(s)) will be needed everywhere, but so far we only have them for Debian...
Rebuilders and sharing signed checksums, cont.

- Individually signed checksums (think web of trust) could work in the Debian case (we have a gpg web of trust), but IMO won’t scale.
- Another idea: rebuilders, run by large organisations (ACLU, CCC, CERN, Deutsche Bank, EDF, EON, Greenpeace, NASA, NSA, XYZ).
- Fedora rebuilds Debian, Debian rebuilds OpenSUSE, OpenSUSE rebuilds NetBSD, etc…
- Big customers could just rebuild everything themselves.
Integration in user tools

- "Do you really want to install this unreproducible software (y/N)"

h01ger and lynxis
Reproducible builds everywhere
OpenWrt Summit, Berlin
Integration in user tools

- "Do you really want to install this unreproducible software (y/N)"
- "Do you want to build those packages which have unconfirmed checksums, before installing? (Y/n)"
Integration in user tools

- "Do you really want to install this unreproducible software (y/N)"
- "Do you want to build those packages which have unconfirmed checksums, before installing? (Y/n)"
- "How many signed checksums do you require to call a package 'reproducible'?" - and whom do you trust?
Motivation
Common resources
Status Debian
Status Non-Debian World
Future work
Getting involved
Questions, comments, ideas?
As a software developer

- Stop using build dates
- Use SOURCE_DATE_EPOCH instead
- See https://reproducible-builds.org/specs/
Form your reproducible builds team!

**Why?**
- Every distribution should be reproducible!
- Learn something new every day!
- Change the (software) world!
- https://tests.reproducible-builds.org/openwrt needs your help
- https://tests.reproducible-builds.org/lede needs your help

**How to get started?**
- Build something twice, run diffoscope on the results.
- Talk to lynxis or h01ger here or talk to us on IRC or via mail.
- RTFM, there is lots of documentation
- Experiment - learning by doing
1 Motivation
2 Common resources
3 Status Debian
4 Status Non-Debian World
5 Future work
6 Getting involved
7 Questions, comments, ideas?
Thanks to...! ...and thank you, too!

- All “Reproducible Builds” contributors
  (you are just so awesome!)
- OpenWrt Summit and ELCE

holger@debian.org  B8BF  5413  7B09  D35C  F026
                    FE9D  091A  B856  069A  AA1C
lynxis@fe80.eu     390D  CF78  8BF9  AA50  4F8F
                    F1E2  C29E  9DA6  A0DF  8604
Questions, comments, ideas?

- https://reproducible-builds.org/
- #reproducible-builds on irc.OFTC.net
- https://lists.reproducible-builds.org
- twitter: @ReproBuild
Questions, comments, ideas?

- https://reproducible-builds.org/
- #reproducible-builds on irc.OFTC.net
- https://lists.reproducible-builds.org
- twitter: @ReproBuild
- Mike and Seth’s talk from 31c3 about motivations
- Lunar’s talk about fixing reproducible issues from CCCamp 15
- h01ger’s talk ”the Reproducible builds ecosystem” from FOSDEM 16