Stretching out for trustworthy reproducible builds

Holger Levsen
holger@debian.org

Lunar
lunar@debian.org

FOSDEM'15
2015-01-31

What?

What are reproducible builds?

"reproducible builds"
enable anyone to reproduce
identical binary packages
from a given source

"I want to believe"

$$\text{source} \xrightarrow{\text{build}} \text{binary}$$

"I want to believe"

free software

freedom to study

freedom to run

source —build→ binary

"I want to believe"

$$\text{source} \xrightarrow{\text{build}} \text{binary}$$

can be verified

can be used

"I want to believe"

source $\xrightarrow{\text{build}}$ binary

can be verified

can be used
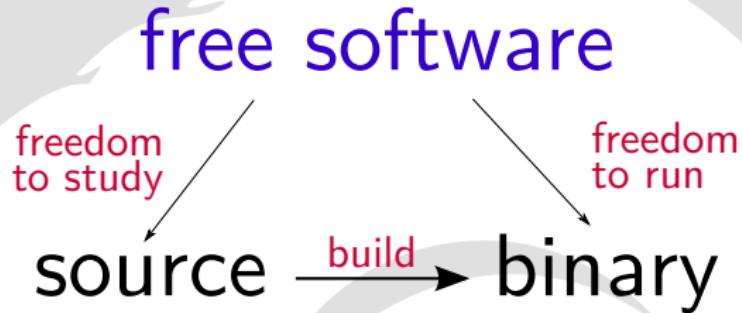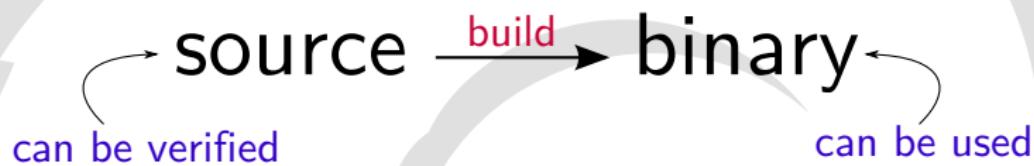
prove it
to me!
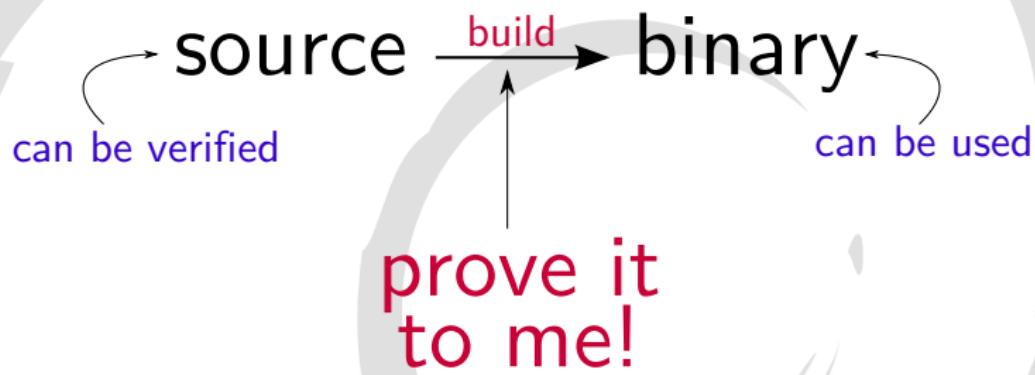
Why?

Why?

Reproducible builds allow for **independent verifications**
that a binary matches what the source intended to
produce.

... and other nice things.

But I'm the developer!

"*I know what's in the binary because I compiled it myself!*"

"*I'm an upstanding, careful, and responsible individual!*"

"*Why should I have to worry about hypothetical risks about the contents of my binaries?*"

But the build machines are secure

- How can you be sure?

Unpleasant thoughts

- We think of software development as a fundamentally benign activity.
  - "*I'm not that interesting.*"

- Users can be targeted through developers
- Known successful attacks against infrastructure used by Linux (2003), FreeBSD (2013)

Strong motivations

- Compromise one computer to get:
  - ▸ Hundreds of millions of other computers?
  - ▸ Every bank account in the world?
  - ▸ Every Windows computer in the world?
  - ▸ Every Linux server in the world?

- Compromise one computer is worth:
  - ▸ $100k USD? (Market price of remote 0day)
  - ▸ $100M USD? (Censorship budget of Iran per year)
  - ▸ $4B USD? (Bitcoin market cap)

How small can a backdoor be?

OpenSSH 3.0.2 (CVE-2002-0083) – exploitable security bug (privilege escalation: user can get root)

```
 {
    Channel *c;
-   if (id < 0 || id > channels_alloc) {
+   if (id < 0 || id >= channels_alloc) {
      log("channel_lookup: %d: bad id", id);
      return;
    }
```

Result of fixing the bug (asm)

```
cmpl $0x0,0x8(%ebp)        cmpl $0x0,0x8(%ebp)
js 16                      js 16
mov 0x4,%eax               mov 0x4,%eax
cmp %eax,0x8(%ebp)         cmp %eax,0x8(%ebp)
jle 30                     jl 30
mov 0x8(%ebp),%eax         mov 0x8(%ebp),%eax
mov %eax,0x4(%esp)         mov %eax,0x4(%esp)
movl $0x4c,(%esp)          movl $0x4c,(%esp)
call 25                    call 25
```

Result of fixing the bug (asm)

```
cmpl $0x0,0x8(%ebp)        cmpl $0x0,0x8(%ebp)
js 16                       js 16
mov 0x4,%eax                mov 0x4,%eax
cmp %eax,0x8(%ebp)          cmp %eax,0x8(%ebp)
jle 30                      jl 30
mov 0x8(%ebp),%eax          mov 0x8(%ebp),%eax
mov %eax,0x4(%esp)          mov %eax,0x4(%esp)
movl $0x4c,(%esp)           movl $0x4c,(%esp)
call 25                     call 25
```

Resulting difference in the binary

What's the difference between `if (a > b)`
and `if (a >= b)` in x86 assembly?

assembly:	JLE	JL
opcode:	0x7E	0x7C
binary:	01111110	01111100

A single bit!

Other corresponding opcode pairs also differ by just a
single bit (JGE=0x7D, JG=0x7F)

Result of fixing the bug (hex)

Vulnerable	Fixed
55 89 e5 83 ec	55 89 e5 83 ec
28 83 7d 08 00	28 83 7d 08 00
78 0a a1 04 00	78 0a a1 04 00
00 00 39 45 08	00 00 39 45 08
7e 1a 8b 45 08	7c 1a 8b 45 08
89 44 24 04 c7	89 44 24 04 c7
04 24 4c 00 00	04 24 4c 00 00
00 e8 fc ff ff	00 e8 fc ff ff
ff b8 00 00 00	ff b8 00 00 00
00 eb 35	00 eb 35

Overall file size: approx. 500 kB

Result of fixing the bug (hex)

Vulnerable	Fixed
55 89 e5 83 ec	55 89 e5 83 ec
28 83 7d 08 00	28 83 7d 08 00
78 0a a1 04 00	78 0a a1 04 00
00 00 39 45 08	00 00 39 45 08
7e 1a 8b 45 08	7c 1a 8b 45 08
89 44 24 04 c7	89 44 24 04 c7
04 24 4c 00 00	04 24 4c 00 00
00 e8 fc ff ff	00 e8 fc ff ff
ff b8 00 00 00	ff b8 00 00 00
00 eb 35	00 eb 35

Overall file size: approx. 500 kB

How did this start?

Bitcoin's motivation

- Malicious modifications to binaries could result in irrevocable unwanted transfers of bitcoins
- Individual developers could be blamed for such modifications
- Users might not believe that a developer's machine was hacked
- Reproducible builds therefore protect developers

Nothing new though

```
From: Martin Uecker <muecker@gmx.de>
Cc: debian-devel@lists.debian.org
Date: Sun, 23 Sep 2007 23:32:59 +0200
```

I think it would be really cool if the Debian policy required that packages could be rebuild bit-identical from source. At the moment, it is impossible to independly verify the integricity of binary packages.

https://lists.debian.org/debian-devel/2007/09/msg00746.html

Wouldn't it be cool?

- Debian is the largest collection of free software
- More than 21,000 source packages
- "Our priorities are our **users** and **free software**"

How?

How?

- Record the build environment
- Reproduce the build environment
- Eliminate unneeded variations

How:

Record the build environment

*.buildinfo

New control file *.buildinfo which records:

- Versions of build dependencies
  - ... and their dependencies
- Checksum of the source package.
- Checksums of the binary packages.

Example *.buildinfo

```
Format: 1.9
Build-Architecture: amd64
Source: txtorcon
Binary: python-txtorcon
Architecture: all
Version: 0.11.0-1
Build-Path: /usr/src/debian/txtorcon-0.11.0-1
Checksums-Sha256:
 a26549d9…7b 125910 python-txtorcon_0.11.0-1_all.deb
 28f6bcbe…69 2039 txtorcon_0.11.0-1.dsc
Build-Environment:
 base-files (= 8),
 base-passwd (= 3.5.37),
 bash (= 4.3-11+b1),
 …
```

How:
Reproduce the build environment

snapshot.debian.org

- `snapshot.debian.org` archives every state of the Debian archive.
- 2015-01-29: 28 terabytes of data in almost 17 million files.

srebuild

- Thin wrapper around sbuild
- Find the right archive snapshot
- Install packages listed in the *.buildinfo file
- Start the build

Status: work in progress

How:

Eliminate unneeded variations

General approach

- Gitian (Bitcoin, Tor Browser):
  - ▸ Use a VM: same kernel, same user, same build path
  - ▸ `libfaketime`

- Debian:
  - ▸ Fix the tools
  - ▸ Fix the build systems
  - ▸ Work-arounds as last resort

strip-nondeterminism

- Normalize various file formats
- Currently handle:
  - ar archives (`.a`)
  - gzip
  - Java jar
  - Javadoc HTML
  - ZIP archives

- Written in Perl (like `dpkg-dev`)

Investigating packages

debbindiff :

- Two files → HTML
- Recursively unpack archives
- Seeks human readability:
  - ▸ uncompress PDF
  - ▸ disassemble binaries,
  - ▸ unpack Gettext files,
  - ▸ *... easy to extend to new file formats*
- Falls back to binary comparison
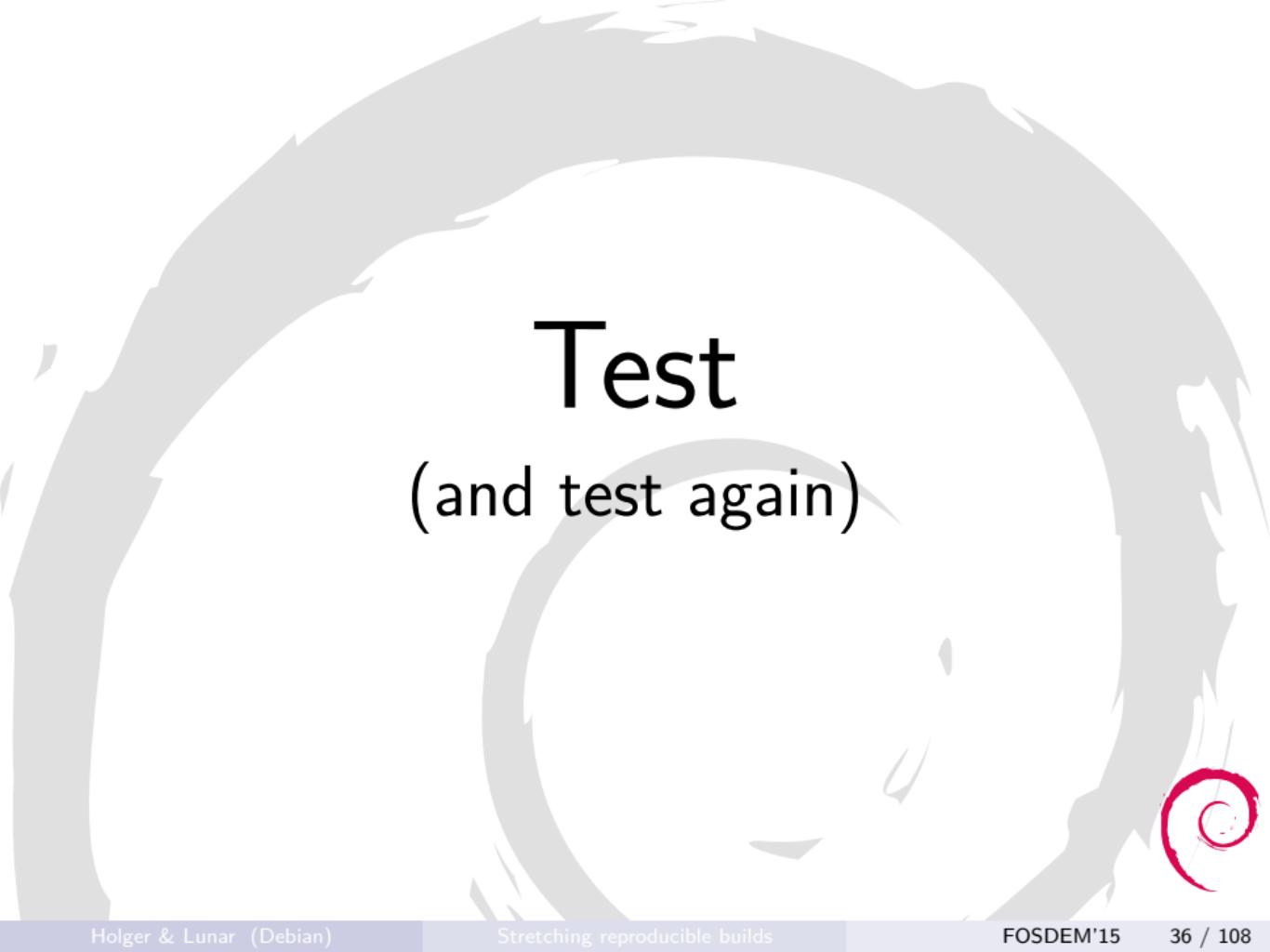
Test
(and test again)

reproducible.debian.net: idea

- Build the package
- *Re*build the package
- Compare the results

Initially a 10 lines shell script...

reproducible.debian.net: today

- Driven by jenkins.debian.net
  - announced 2014-09-26

- 14 jobs defined in 263 LOC of YAML
  - 3 build jobs, rescheduled every 2 minutes
  - Job for auto-scheduling
  - Jobs to recreate pbuilder base.tgz and debbindiff schroot
  - Job to update notes and issues from Git
  - JSON output
  - Remaining jobs mostly to create HTML and graphs

- 13 Shell scripts with 2049 LOC
- 4 Python scripts with 948 LOC

reproducible.debian.net: variations

The second build is different in:

- time
- file ordering
- CPU ordering and availability
- hostname
- user & group
- locale

reproducible.debian.net: more goodies

- `https://reproducible.debian.net/$PACKAGE`
- Reproducibility status in `tracker.debian.org`
- Package sets:
  - `Essential:yes, build-essential`
  - 1337 most frequently installed packages (`popcon`)
  - Packages installed on Debian machines
  - All packages in Tails and their build dependencies
  - All packages in Grml and their build dependencies
- List of unreproducible packages per maintainer

reproducible.debian.net: not reproducible

Overview of packages which failed to build reproducibly

21557 packages have been attempted to be build so far, that's 100.0% of 21557 source packages in Debian sid currently. Out of these, 17632 packages (81.7%) could be built reproducible!

Have a look at: ☀ 🚜 🌩 ⚠ ☁ ⛔ issues | packages with notes | currently scheduled | packages tested in the last 24h | packages tested in the last 48h | all tested packages (sorted alphabetically) | maintainers of unreproducible packages | stats | package sets stats

---

🐭 3058 (14.1%) packages which failed to build reproducibly: **java-xmlbuilder** gfpoken **freehep-graphicsio-tests munge-maven-plugin maven** nvidia-settings camping **libmodule-build-perl** octave-secs1d alienblaster primrose **gpgme1.0 webassets** vmm ubuntu-packaging-guide **txtorcon tsung** taurus statsmodels spykeviewer skimage **shinken** sardana **salt** rdflib quodlibet **qiime** pythondialog **python-tidylib** python-shapely python-sfml python-secretstorage python-repoze.who python-reconfigure python-qt4 **python-pyaudio** python-mzml **python-fitbit** python-expyriment python-django-mptt **python-cogent** pytables pyqt5 pyotherside pyopencl pymvpa2 **pymvpa pycurl pycairo py-postgresql** py3cairo psi4 powerline plainbox phatch **php-guzzle pebl paramiko** owncloud-doc owncloud-client **opencolorio**

reproducible.debian.net: package

gdb 7.7.1+dfsg-5 at 2015-01-24 15:50 UTC: notes debbindiff buildinfo rbuild (4MB) PTS BTS sources debian/rules   reproducible builds

Version annotated:	7.7.1+dfsg-5		
Identified issues:			
	Identifier	timestamps_in_jar	
	URL	https://wiki.debian.org/ReproducibleBuilds/TimestampsInJarFiles	
	Description	jar (Java Archive) file stores mtimes. Most often of files generated during the build.	
	Identifier	timestamps_in_dvi_generated_by_latex	
	URL	https://wiki.debian.org/ReproducibleBuilds/TimestampsInDviGeneratedByLaTeX	
	Description	The creation date of the DVI file is the current time.	
Comments:	Capture build time and hostname in /usr/share/doc/gdb/check.log.gz		

reproducible.debian.net: package

reproducible.debian.net: package

gdb 7.7.1+dfsg-5 🐌 at 2015-01-24 15:50 UTC: notes debbindiff buildinfo rbuild (4MB) PTS BTS sources debian/rules   reproducible builds

```
Starting to build gdb on 2015-01-24 15:50
The jenkins build log is/was available at https://jenkins.debian.net/job/reproducible_builder_alpha/26194//console
Reading package lists...
Building dependency tree...
Reading state information...
NOTICE: 'gdb' packaging is maintained in the 'Git' version control system at:
git://anonscm.debian.org/pkg-gdb/gdb.git
Need to get 17.7 MB of source archives.
Get:1 http://ftp.de.debian.org/debian/ sid/main gdb 7.7.1+dfsg-5 (dsc) [2783 B]
Get:2 http://ftp.de.debian.org/debian/ sid/main gdb 7.7.1+dfsg-5 (tar) [17.7 MB]
Get:3 http://ftp.de.debian.org/debian/ sid/main gdb 7.7.1+dfsg-5 (diff) [71.0 kB]
Fetched 17.7 MB in 0s (21.5 MB/s)
Download complete and in download only mode
-----BEGIN PGP SIGNED MESSAGE-----
Hash: SHA512

Format: 3.0 (quilt)
Source: gdb
Binary: gdb, gdb-minimal, gdb64, gdb-multiarch, gdbserver, gdb-source, gdb-dbg
Architecture: any all
```

Experimental "reproducible" toolchain

- Test and refine changes to tools used by build systems
- Currently 20 source packages
- Examples : `dpkg`, **`debhelper`**, `cdbs`, `libxslt`, `r-base`, `gtk-doc`, `fontforge`, …
- Most patches already submitted
- `git://git.debian.org/reproducible/$pkg.git` in branch `pu/reproducible_builds`

reproducible.debian.net overall stats

Package reproducibility status

For those in the back

$$81.7\%$$
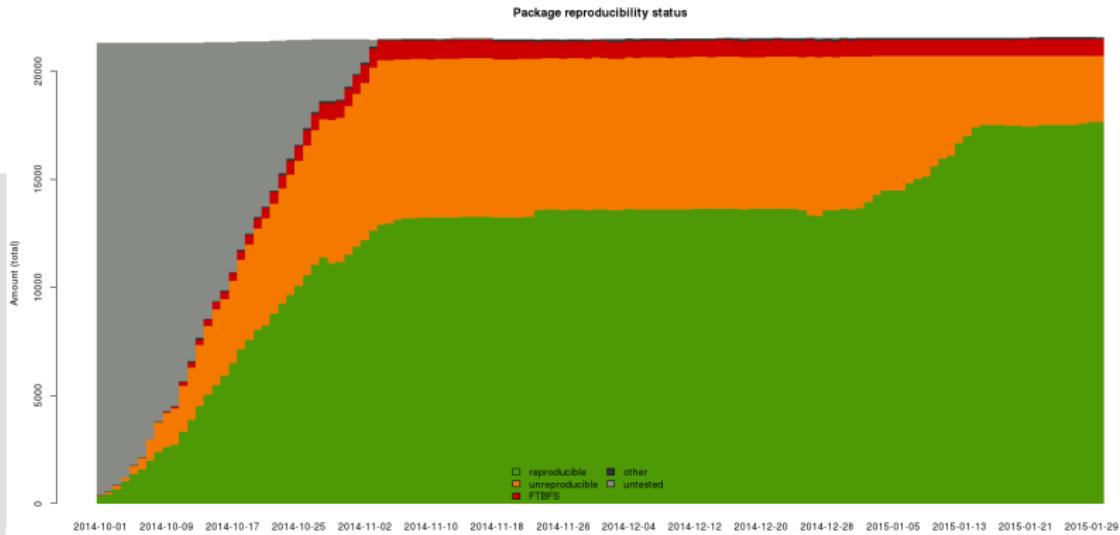
More than 17,600 source packages!

For those in the back

81.7%

More than 17,600 source packages!

(in our test environment!)

What is not tested (yet)

- `uname` output
- Day/month/year
- `/proc/cpuinfo`
- *More?*

Investigated packages

Packages which have notes
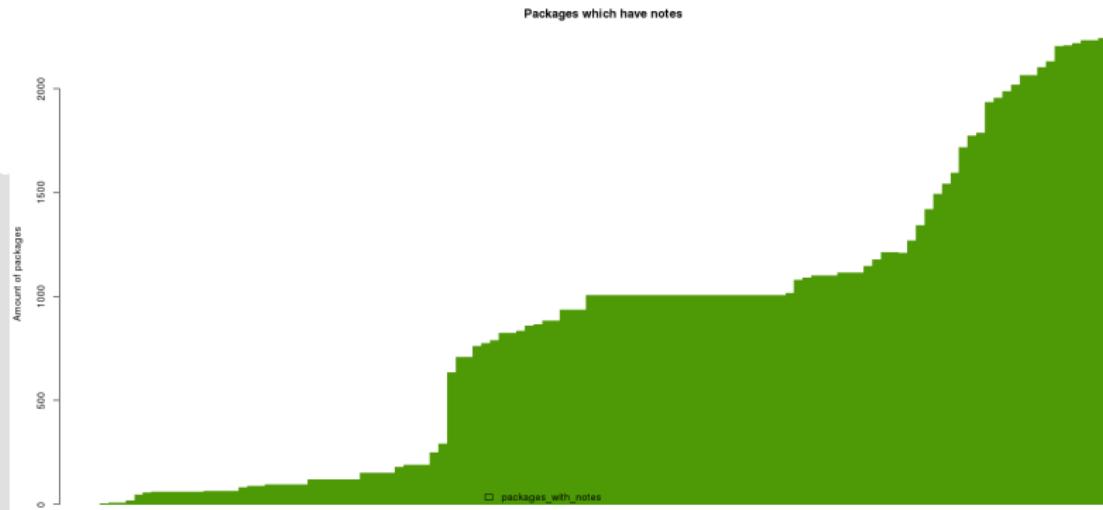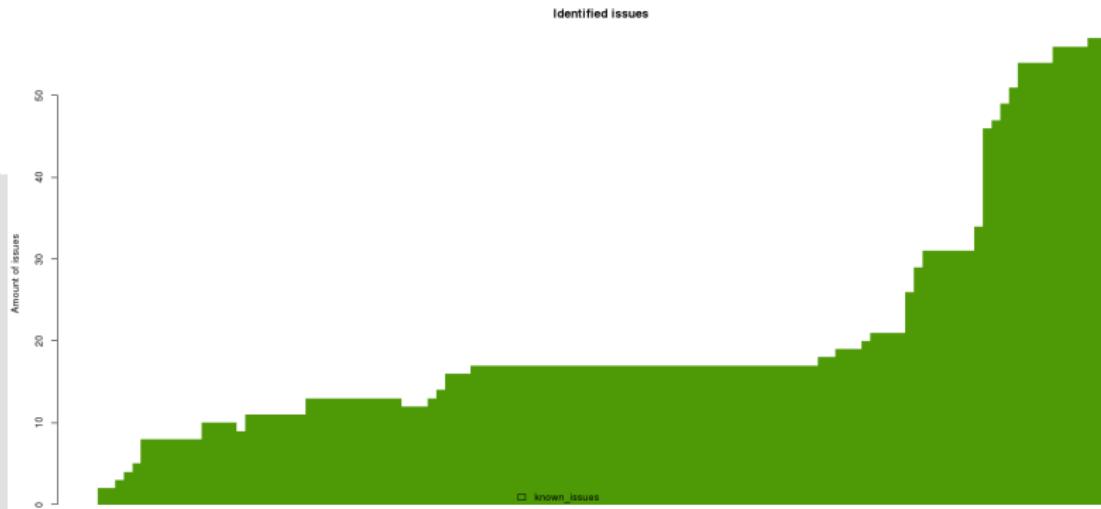
Identified issues

Findings

Identified issues

- Timestamps (recording current time)
- File order
- (Pseudo-)randomness:
    - Temporary file paths
    - UUID
    - Protection against complexity attacks

Identified issues (cont.)

- CPU and memory related:
  - ▸ Code optimizations for current CPU class
  - ▸ Recording of memory addresses

- ~~Build-path~~

- Others, eg. locale settings

Examples
Timestamps added by build systems

Timestamps in static libraries

Timestamps in static libraries (cont.)

data.tar

./usr/lib/libuninum.a
No differences found inside, yet data differs

```
2f20 2020 2020 2020   !<arch>./           2f20 2020 2020 2020   !<arch>./
3134 3231 3837 3334          14218734     3134 3231 3837 3334          14218734
2020 3020 2020 2020   06  0     0          2020 3020 2020 2020   51  0     0
3239 3634 2020 2020   0       2964         3239 3634 2020 2020   0       2964
0000 0bd8 0000 0bd8   `.............       0000 0bd8 0000 0bd8   `.............
0000 0bd8 0000 0bd8   .............        0000 0bd8 0000 0bd8   .............
0000 0bd8 0000 0bd8   .............        0000 0bd8 0000 0bd8   .............
0000 0bd8 0000 0bd8   .............        0000 0bd8 0000 0bd8   .............
0000 0bd8 0000 0bd8   .............        0000 0bd8 0000 0bd8   .............
0bd8 0000 0bd8 0000 0bd8 0000 0bd8  .....  0bd8 0000 0bd8 0000 0bd8 0000 0bd8  .....
6e67 6433 0074 656e   engd2.tengd3.ten     6e67 6433 0074 656e   engd2.tengd3.ten
6435 0074 656e 6764   gd4.tengd5.tengd     6435 0074 656e 6764   gd4.tengd5.tengd
0074 656e 6764 3800   6.tengd7.tengd8.     0074 656e 6764 3800   6.tengd7.tengd8.
656e 6764 3130 0074   tengd9.tengd10.t     656e 6764 3130 0074   tengd9.tengd10.t
756e 696e 756d 2e6f   engd11..uninum.o     756e 696e 756d 2e6f   engd11..uninum.o
3134 3231 3837 3334   /        14218734    3134 3231 3837 3334   /        14218734
2020 3132 3334 2020   05  1234  1234        2020 3132 3334 2020   50  1234  1234
3336 3636 3038 2020   100644  366608       3336 3636 3038 2020   100644  366608
```

Timestamps written by Maven

Timestamps written by PyQt4

Timestamps written by Erlang compiler

Timestamps in PE binaries

Windows, UEFI, Mono…

Timestamps in Java jar

data.tar

./usr/share/java/flute-1.1-SNAPSHOT.jar

metadata

```
1.1-SNAPSHOT.jar                          1.1-SNAPSHOT.jar
4568 bytes, number of entries: 45         4568 bytes, number of entries: 45
unx       0 bx stor 15-Jan-07 20:23 META  unx       0 bx stor 15-Jan-07 20:25 META
unx     259 b- defN 15-Jan-07 20:23 META  unx     259 b- defN 15-Jan-07 20:25 META
unx       0 b- stor 15-Jan-07 20:23 org/  unx       0 b- stor 15-Jan-07 20:25 org/
unx       0 b- stor 15-Jan-07 20:23 org/  unx       0 b- stor 15-Jan-07 20:25 org/
unx       0 b- stor 15-Jan-07 20:23 org/  unx       0 b- stor 15-Jan-07 20:25 org/
unx       0 b- stor 15-Jan-07 20:23 org/  unx       0 b- stor 15-Jan-07 20:25 org/
unx       0 b- stor 15-Jan-07 20:23 org/  unx       0 b- stor 15-Jan-07 20:25 org/
unx     585 b- defN 15-Jan-07 20:23 org/  unx     585 b- defN 15-Jan-07 20:25 org/
unx    5744 b- defN 15-Jan-07 20:23 org/  unx    5744 b- defN 15-Jan-07 20:25 org/
unx     320 b- defN 15-Jan-07 20:23 org/  unx     320 b- defN 15-Jan-07 20:25 org/
```

They are ZIP archives, really.

Timestamps in PHP registry

data.tar.xz

  data.tar

    ./usr/share/php/.registry/.channel.pecl.php.net/oauth.reg

`"2.0";s:13:"_lastmodified";i:1419537766;}`   `"2.0";s:13:"_lastmodified";i:1419537812;}`
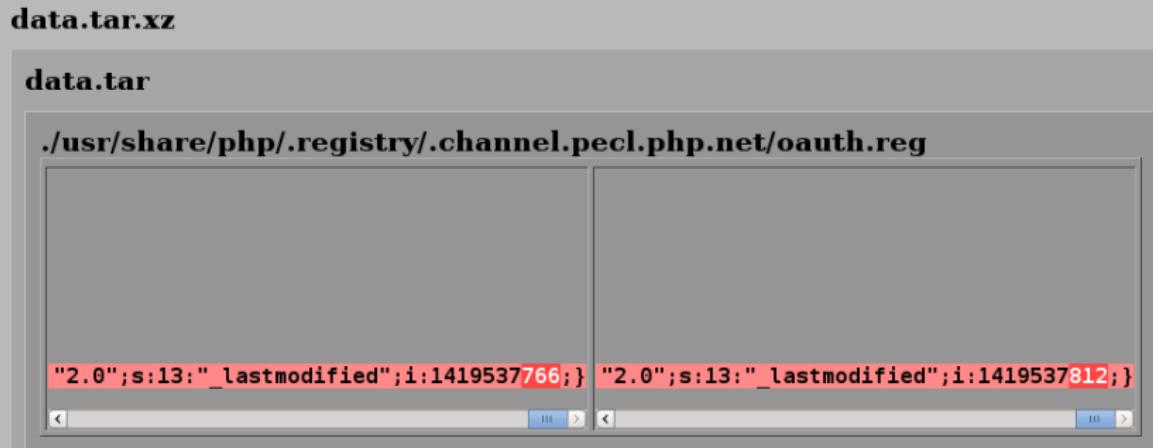
Timestamps by a template engine

Identified issues (cont.)

Examples

Archives

Timestamps in gzip headers

data.tar.xz

data.tar

./usr/share/ispell/brasileiro.mwl.gz

metadata

```
1  gzip compressed data, was "br.ispell",  | last modified: Sat Jan  3 22:47:40 2015
```

Timestamps in ZIP archives


```
data.tar.xz

  data.tar

  ./usr/share/boa-constructor/Docs/boa.apphelp.htb

    metadata

    +--249 lines: Archive:  boa.apphelp.htb-----------------------------------
      minimum file system compatibility required:    MS-DOS, OS/2 or NT FAT
      minimum software version required to extract:  2.0
      compression method:                            deflated
      compression sub-type (deflation):              normal
      file security status:                          not encrypted
      extended local header:                         no
      file last modified on (DOS date/time):         2015 Jan 16 09:55:52
      file last modified on (UT extra field modtime): 2015 Jan 16 09:55:52 local
      file last modified on (UT extra field modtime): 2015 Jan 16 09:55:52 UTC
      32-bit CRC value (hex):                        5bb8662c
      compressed size:                               1660 bytes
      uncompressed size:                             5682 bytes
      length of filename:                            19 characters
      length of extra field:                         24 bytes
      length of file comment:                        0 characters
```

Timestamps in tarballs

Users and groups in tarballs

Identified issues (cont.)

Examples

Timestamps in documentation

Timestamps written by Doxygen

Timestamps written by docbook-to-man

Timestamps written by Ghostscript

data.tar.xz

data.tar

./usr/share/doc/c++-annotations/cplusplus.pdf

pdftk cplusplus.pdf output - uncompress

```
                                              ----------------------------------------

                                              d='W5M0MpCehiHzreSzNTczkc9d'?>

                                              a/' x:xmptk='XMP toolkit 2.9.1-13, fram
                                              3.org/1999/02/22-rdf-syntax-ns#' xmlns:
er>dvips + GPL Ghostscript 9.06</pdf:Pr       :2b7e66bf-d448-11ef-0000-a06d27858a43'

Date>2015-01-14T20:23:14Z</xmp:ModifyDa       :2b7e66bf-d448-11ef-0000-a06d27858a43'
                                              :29Z</xmp:CreateDate>
                                              rref package</xmp:CreatorTool></rdf:Des
DocumentID='uuid:b743c33d-d447-11ef-000       :2b7e66bf-d448-11ef-0000-a06d27858a43'
t='application/pdf'><dc:title><rdf:Alt>       :2b7e66bf-d448-11ef-0000-a06d27858a43'
```

Timestamps written by Ghostscript (cont.)

Timestamps written by GNU groff

Timestamps written by Javadoc

data.tar.xz

  data.tar

   ./usr/share/doc/libconstantine-java-doc/javadoc/allclasses-frame.html

```
DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN" "http://www.w3.org/TR/h
-- NewPage -->
ml lang="en">
ead>
-- Generated by javadoc (version 1.7.0_65) on Wed Jan 07 12:39:34 UTC 2015 -->
eta http-equiv="Content-Type" content="text/html" charset="UTF-8">
tle>All Classes</title>
eta name="date" content="2015-01-07">
nk rel="stylesheet" type="text/css" href="stylesheet.css" title="Style">
head>
ody>
147 lines: <h1 class="bar">All Classes</h1>---------------------------------
```

Timestamps written by man2html

Timestamps in TeX output (.dvi)

Identified issues (cont.)

Examples
"Compiled at/on/by"

Build time via C preprocessor macros

```
80    {"to",1,NULL,'t'},
81     {0,0,0,0}
82  };
83
84  #define MSGSIZE 512
85  #define INPUT_TEXT_SIZE_INIT 32
86
87  char *pgname ="numconv";
88  char compdate[]="Compiled " __DATE__ " " __TIME__ ;
89
90  void ShowNumberSystems(int which) {
```

package info (click to toggle)
libuninum 2.7-1.1
- links: PTS
- area: main
- in suites: jessie, sid, wheezy
- size: 1,972 kB
- SLOC: ansic: 9,968; sh: 8,715; tcl: 553; makefile: 78

Build time via C preprocessor macros

```
190 | z_version(void)
191 | {
192 |   printf("Zorp %s (%s)\n"
193 |          "Revision: %s\n"
194 |          "Compile-Date: %s %s\n"
195 |          "Config-Date: %s\n"
196 |          "Trace: %s\n"
197 |          "Debug: %s\n"
198 |          "IPOptions: %s\n"
199 |          "%s\n"
200 |          ,
201 |          BROCHURE_VERSION, VERSION,
202 |          ZORP_SOURCE_REVISION,
203 |          __DATE__, __TIME__,
204 |          ZORP_CONFIG_DATE,
205 |          ON_OFF_STR(ENABLE_TRACE),
206 |          ON_OFF_STR(ENABLE_DEBUG),
207 |          ON_OFF_STR(ENABLE_IPOPTIONS),
```

package info (click to toggle)

zorp 3.9.5-4

- links: PTS
- area: main
- in suites: jessie, sid, wheezy
- size: 5,412 kB
- ctags: 5,387

Build time recorded via Makefile

```
149  genversion.$(OBJEXT): $(genversion_SOURCES) genversion.h
150
151  genversion.h: $(top_builddir)/config.status
152          -rm -f $@ $@.new
153          echo '#define CC "$(CC)"' > $@.new
154          echo '#define BUILT_DATE "'`date`'"' >> $@.new
155          echo '#define BUILT_MACH "$(target)"' >> $@.new
156          mv $@.new $@
157
158  version.c: genversion$(EXEEXT)
159          -rm -f version.c
160          ./genversion$(EXEEXT) > version.c
161
162  BUILT_SOURCES += genversion.h version.c
```

package info (click to toggle)
amanda 1:3.3.6-4
- links: PTS, VCS
- area: main
- in suites: jessie, sid
- size: 25,248 kB
- ctags: 28,823
- SLOC: ansic: 225,464; perl:

Hostname recorded via ./configure

```
364      VERSION="$VERSION (svn$SVN_REV)"
365    fi
366  fi
367  HOSTNAME=`hostname`
368  DATE=`date +"%d.%m.%Y %H:%M:%S %Z"`
369
370  cat > version.h <<EOF
371  /*
372   * anytun version info
373   *
374   * this file was created automatically
375   * do not edit this file directly
376   * use ./configure instead
377   */
378
379  #ifndef ANYTUN_version_h_INCLUDED
380  #define ANYTUN_version_h_INCLUDED
381
382  #define VERSION_STRING_0 " version $VERSION"
383  #define VERSION_STRING_1 "built on $HOSTNAME, $DATE"
```

package info (click to toggle)
anytun 0.3.5-1
- links: PTS, VCS
- area: main
- in suites: jessie, sid
- size: 1,424 kB
- ctags: 1,339
- SLOC: cpp: 9,126; sh: 618;
  makefile: 367

Build time recorded via ./configure

```
364       VERSION="$VERSION (svn$SVN_REV)"
365     fi
366   fi
367   HOSTNAME=`hostname`
368   DATE=`date +"%d.%m.%Y %H:%M:%S %Z"`
369
370   cat > version.h <<EOF
371   /*
372    * anytun version info
373    *
374    * this file was created automatically
375    * do not edit this file directly
376    * use ./configure instead
377    */
378
379   #ifndef ANYTUN_version_h_INCLUDED
380   #define ANYTUN_version_h_INCLUDED
381
382   #define VERSION_STRING_0 " version $VERSION"
383   #define VERSION_STRING_1 "built on $HOSTNAME, $DATE"
```

package info (click to toggle)

anytun 0.3.5-1

- links: PTS, VCS
- area: main
- in suites: jessie, sid
- size: 1,424 kB
- ctags: 1,339
- SLOC: cpp: 9,126; sh: 618;
  makefile: 367

m4 macros for autoconf (build time)

```
75    AC_FLDIGI_SH_DQ([echo $ac_configure_args])
76    AC_DEFINE_UNQUOTED([BUILD_CONFIGURE_ARGS], [$ac_sh_dq], [Configure arguments])
77
78    AC_FLDIGI_SH_DQ([date])
79    AC_DEFINE_UNQUOTED([BUILD_DATE], [$ac_sh_dq], [Build date])
80
81    AC_FLDIGI_SH_DQ([whoami])
82    AC_DEFINE_UNQUOTED([BUILD_USER], [$ac_sh_dq], [Build user])
83
84    AC_FLDIGI_SH_DQ([hostname])
85    AC_DEFINE_UNQUOTED([BUILD_HOST], [$ac_sh_dq], [Build host])
86
87    AC_FLDIGI_SH_DQ([$CXX -v 2>&1 | tail -1])
88    AC_DEFINE_UNQUOTED([BUILD_COMPILER], [$ac_sh_dq], [Compiler])
```

m4 macros for autoconf (username)

```
75    AC_FLDIGI_SH_DQ([echo $ac_configure_args])
76    AC_DEFINE_UNQUOTED([BUILD_CONFIGURE_ARGS], [$ac_sh_dq], [Configure arguments])
77
78    AC_FLDIGI_SH_DQ([date])
79    AC_DEFINE_UNQUOTED([BUILD_DATE], [$ac_sh_dq], [Build date])
80
81    AC_FLDIGI_SH_DQ([whoami])
82    AC_DEFINE_UNQUOTED([BUILD_USER], [$ac_sh_dq], [Build user])
83
84    AC_FLDIGI_SH_DQ([hostname])
85    AC_DEFINE_UNQUOTED([BUILD_HOST], [$ac_sh_dq], [Build host])
86
87    AC_FLDIGI_SH_DQ([$CXX -v 2>&1 | tail -1])
88    AC_DEFINE_UNQUOTED([BUILD_COMPILER], [$ac_sh_dq], [Compiler])
```

m4 macros for autoconf (hostname)

```
75    AC_FLDIGI_SH_DQ([echo $ac_configure_args])
76    AC_DEFINE_UNQUOTED([BUILD_CONFIGURE_ARGS], [$ac_sh_dq], [Configure arguments])
77
78    AC_FLDIGI_SH_DQ([date])
79    AC_DEFINE_UNQUOTED([BUILD_DATE], [$ac_sh_dq], [Build date])
80
81    AC_FLDIGI_SH_DQ([whoami])
82    AC_DEFINE_UNQUOTED([BUILD_USER], [$ac_sh_dq], [Build user])
83
84    AC_FLDIGI_SH_DQ([hostname])
85    AC_DEFINE_UNQUOTED([BUILD_HOST], [$ac_sh_dq], [Build host])
86
87    AC_FLDIGI_SH_DQ([$CXX -v 2>&1 | tail -1])
88    AC_DEFINE_UNQUOTED([BUILD_COMPILER], [$ac_sh_dq], [Compiler])
```

Examples

File ordering

File ordering in python-support files

Identified issues (cont.)

Examples
Randomness

Random Perl hash order

See *Algorithmic complexity attacks* in perlsec(1).

Random serial numbers in Ogg streams

data.tar.xz

data.tar

./usr/share/games/kiki-the-nano-bot/sound/atom_digest.ogg

Identified issues (cont.)

Examples

Even more timestamps!

Timestamps in PNG

Even images!

# Timestamps in TrueType font files

And fonts!

Please help!

Please help!

- Do not record time, username, hostname, kernel version…
  - … or make it optional.
- Sort file paths.
- Sort dictionary keys.
- Merge our patches!

# Bugs filed
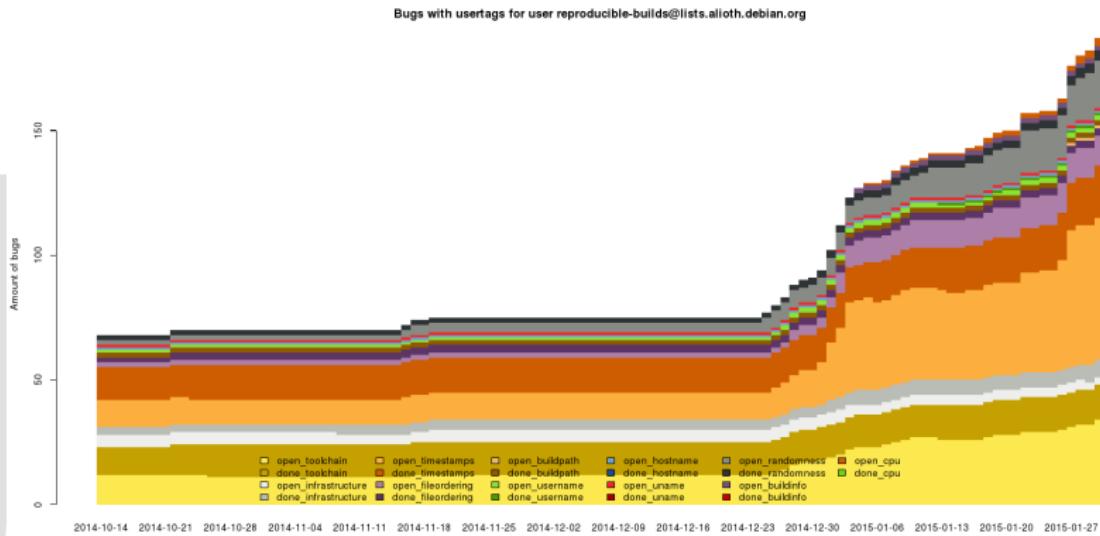
Bugs with usertags for user reproducible-builds@lists.alioth.debian.org
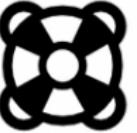
Help Debian?

Help Debian?

- Inventory issues
- Make packages build reproducibly
- Fix known common issues:
  - Get reproducible PE binaries;
  - Random filenames with GCC;
  - …

- Hack Debian archive infrastructure
  - Store and distribute `*.buildinfo` files (#763822)
  - …

- Hack tools to display local packages reproducibility status

Contributors

akira	Hans-Christoph Steiner	Peter De Wachter
Andrew Ayer	Helmut Grohne	Reiner Herrmann
Asheesh Laroia	Holger Levsen	Stefano Rivera
Chris Lamb (lamby)	josch	Stéphane Glondu
Chris West	Lunar	Steven Chamberlain
Daniel Kahn Gillmor	Mattia Rizzolo	Tom Fitzhenry
David Suarez	Niels Thykier	Tomasz Buchert
Drew Fisher	Niko Tyni	Wookey
Guillem Jover	Paul Wise	Ximin Luo

Stay in touch

- Subscribe to the `ReproducibleBuilds` wiki pages.
- Mailing lists:
    - `reproducible-builds@lists.alioth.debian.org`
    - `reproducible-commits@lists.alioth.debian.org`
- Join `#debian-reproducible` on `irc.oftc.net`

Other distributions

- Fedora
  `http://securityblog.redhat.com/2013/09/18/`
  `reproducible-builds-for-fedora/`
- OpenSUSE build-compare
  `https://build.opensuse.org/package/show/`
  `openSUSE:Factory/build-compare`
- NixOS
  `http://lists.science.uu.nl/pipermail/`
  `nix-dev/2013-June/011357.html`

Reproducible Fedora?

- No known activity after the initial blog post.
- Fedora is leading developments for key components.
- Can we help **reproducible Fedora**?

SLES? OpenBSD? Firefox? LibreOffice?

- We hope **reproducible builds** will become the norm.
- Our documentation should be helpful to others.
- Come talk to us!

Future

- Ready for the next Debian release (*stretch*)
- Reproducible installation media
- Reproducible live images (pure Debian, Grml, Tails)
- Reproducible cross-platform builds ("rebootstrap")
- Binary transparency logs: public log of all software releases

Thanks

- Mike Perry and Seth Schoen for their 31C3 talk
- Profitbricks for sponsoring `jenkins.debian.net`
- Globalsign for sponsoring X.509 certificates
- Designers of Tango icons
- Everyone who helped!

Questions? Comments?

?

https://wiki.debian.org/ReproducibleBuilds
https://reproducible.debian.net/
#debian-reproducible on OFTC

Holger	B8BF 5413 7B09 D35C F026 FE9D 091A B856 069A AA1C
Lunar	0603 CCFD 9186 5C17 E88D 4C79 8382 C95C 2902 3DF9