How to make your software build reproducibly

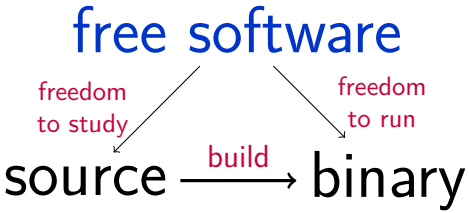
Provide a verifiable path from source to binary

Lunar
lunar@debian.org

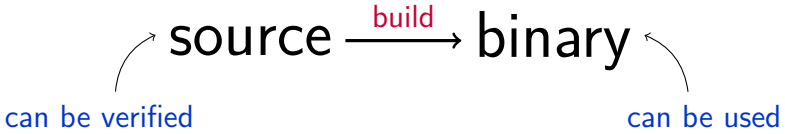
Chaos Communication Camp
2015-08-13

The problem

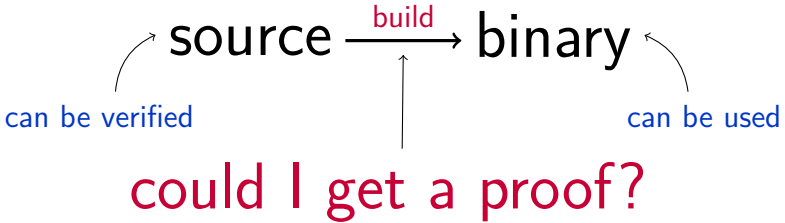
The problem


free software
freedom
to study
freedom
to run
source
build
binary

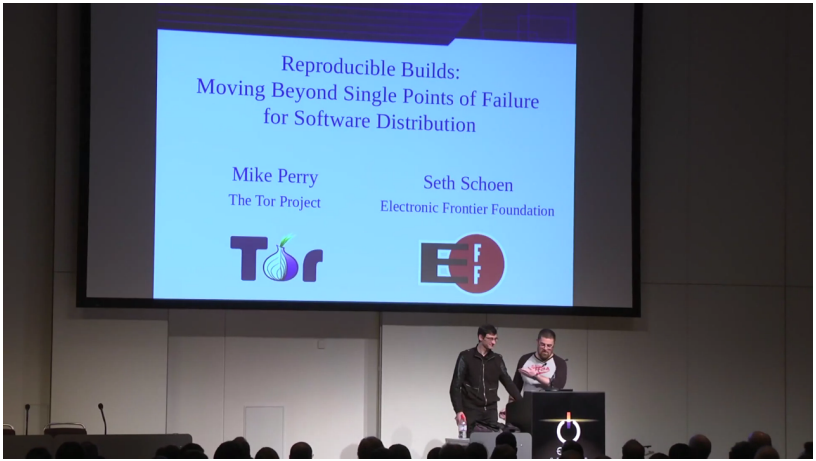
The problem



The problem


source
build
binary
can be verified
can be used
could I get a proof?

Why does it matter?



Available on `media.ccc.de`, 31c3

Just one example

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 securityd to load our own library - which rewrites securityd 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/ispy-cia-campaign-steal-apples-secrets/`

The solution

enable anyone to reproduce
identical binary packages
from a given source

The solution

We call this:

“reproducible builds”

It's trendy!

- Bitcoin (**done**)
- Tor (**done**)
- Debian (*in progress*)
- FreeBSD (*in progress*)
- NetBSD (*in progress*)
- Coreboot (**done**)
- OpenWrt (*in progress*)
- ...

It should become the **norm**.

Multiple aspects

- Deterministic build system
  for those who write source code
- Reproducible build environment
  for those who create binaries for others
- Distributing the build environment
  for those who distribute binaries to the world
- Performing a rebuild and checking the results
  for every one of us

Deterministic build system

In a nutshell:

- Stable inputs
- Stable outputs
- Capture as little as possible from the environment

Common problems

- Timestamps (recording current time)
- File order
- (Pseudo-)randomness:
  - Temporary file paths
  - UUID
  - Protection against complexity attacks
- CPU and memory related:
  - Code optimizations for current CPU class
  - Recording of memory addresses
- Build path
- Locale and timezone settings

```
$ gem install hpricot --source http://code.whytheluckystiff.net
ERROR:  Could not find a valid gem 'hpricot' (>= 0) in any repository
ERROR:  While executing gem ... (Gem::RemoteFetcher::FetchError)
    SocketError: getaddrinfo: Name or service not known (http://code.whyth
eluckystiff.net/latest_specs.4.8.gz)
$
```

Volatile inputs can disappear

- Don't rely on the network
- If you do:
  - Verify content using checksums
  - Have a backup
- The binary distributor should provide a fallback

FreeBSD does it right

```
$ grep MASTER_SITES Makefile
MASTER_SITES= http://gondor.apana.org.au/~herbert/dash/files/
$ cat distinfo
SHA256 (dash-0.5.8.tar.gz) = c6db3a237747b02d20382a761397563d813b306c020ae28ce25…
SIZE (dash-0.5.8.tar.gz) = 223028
$ wget http://distcache.freebsd.org/ports-distfiles/distfiles/dash-0.5.8.tar.gz
```

archive.tar

metadata

Offset 1, 4 lines modified		Offset 1, 4 lines modified	
1	-rwxr-xr-x lunar/lunar 0 2015-08-04 19:01:52 src/	1	-rwxr-xr-x lunar/lunar 0 2015-08-04 19:01:52 src/
2	-rw-r--r-- lunar/lunar 88191 2015-08-04 19:01:52 src/helper.c		
3	-rw-r--r-- lunar/lunar 12780 2015-08-04 19:01:37 src/util.c	2	-rw-r--r-- lunar/lunar 12780 2015-08-04 19:01:37 src/util.c
		3	-rw-r--r-- lunar/lunar 88191 2015-08-04 19:01:52 src/helper.c
4	-rw-r--r-- lunar/lunar 1173 2015-08-04 19:01:24 src/main.c	4	-rw-r--r-- lunar/lunar 1173 2015-08-04 19:01:24 src/main.c

Stable order for inputs

- Always process multiple inputs in the same order
- Directory listings are not stable!

Example

```
tar -cf archive.tar src
```

Stable order for inputs

- Always process multiple inputs in the same order
- Directory listings are not stable!
- Solutions:
  - List inputs explicitely

Example

```
tar -cf archive.tar \
  src/util.c src/helper.c src/main.c
```

Stable order for inputs

- Always process multiple inputs in the same order
- Directory listings are not stable!
- Solutions:
  - List inputs explicitely
  - Use sorting

Example

```
find src -print0 | sort -z |
  tar --null -T - --no-recursion -cf archive.tar
```

Stable order for inputs

- Always process multiple inputs in the same order
- Directory listings are not stable!
- Solutions:
  - List inputs explicitely
  - Use sorting
  - But watch out for difference between locales.

Example

```
find src -print0 | LC_ALL=C sort -z |
  tar --null -T - --no-recursion -cf archive.tar
```

build/cbfs/fallback/bootblock.bin

Offset 1, 10 lines modified		Offset 1, 10 lines modified	
1	00000000: bada baab 0200 0000 5845 0000 0000 009a ....XE......	1	00000000: bada baab 0200 0000 5845 0000 0000 009a ....XE......
2	00000010: 1200 0000 9c5e 0000 4b1d 0000 4045 0000 .....^..K...@E..	2	00000010: 1200 0000 9c5e 0000 4b1d 0000 4045 0000 .....^..K...@E..
3	00000020: 0000 009a 17b9 e22a 009b 1d3c 0020 bd27 ....*...<. .'	3	00000020: 0000 009a 1719 e252 009b 1d3c 0020 bd27 ....R...<. .'
4	00000030: 009b 083c 0000 0825 fcff a923 adde 0a3c ...<...%...#...<	4	00000030: 009b 083c 0000 0825 fcff a923 adde 0a3c ...<...%...#...<
5	00000040: efbe 4a35 0000 0aad feff 0915 0400 0821 ..J5...........!	5	00000040: efbe 4a35 0000 0aad feff 0915 0400 0821 ..J5...........!
6	00000050: 2700 0010 0000 0000 feff 0010 0000 0000 '...............	6	00000050: 2700 0010 0000 0000 feff 0010 0000 0000 '...............
7	00000060: 1800 998c 0800 2003 1c00 848c e8ff bd27 ....'	7	00000060: 1800 998c 0800 2003 1c00 848c e8ff bd27 ....'
8	00000070: 5a00 822c 1000 b0af 2180 8000 0800 4014 Z..,....!.....@.	8	00000070: 5a00 822c 1000 b0af 2180 8000 0800 4014 Z..,....!.....@.
9	00000080: 1400 bfaf 009a 053c 009a 063c 2120 0000 ....<...<!...	9	00000080: 1400 bfaf 009a 053c 009a 063c 2120 0000 ....<...<!...
10	00000090: 4839 a524 7039 c624 e201 800e 4200 0724 H9.$p9.$....B..$	10	00000090: 4839 a524 7039 c624 e201 800e 4200 0724 H9.$p9.$....B..$

Controlled value initialization

- Don't record memory by accident

Example

```
static int write_binary(FILE *out, FILE *in, struct bimg_header *hdr)
{
        static uint8_t file_buf[MAX_RECORD_BYTES];
        struct bimg_data_header data_hdr;
        size_t n_written;

        data_hdr.dest_addr = hdr->entry_addr;
        …
```

Controlled value initialization

- Don't record memory by accident
- Always initialize to a known value

Example

```
static int write_binary(FILE *out, FILE *in, struct bimg_header *hdr)
{
        static uint8_t file_buf[MAX_RECORD_BYTES];
        struct bimg_data_header data_hdr = { 0 };
        size_t n_written;

        data_hdr.dest_addr = hdr->entry_addr;
        …
```

./usr/lib/aspell/de_affix.dat

Offset 1, 11 lines modified		Offset 1, 11 lines modified	
1	# this is the affix file of the de_DE Myspell dictionary	1	# this is the affix file of the de_DE Myspell dictionary
2	# derived from the igerman98 dictionary	2	# derived from the igerman98 dictionary
3	#	3	#
4	# Version: 20131206 (build 20150801)	4	# Version: 20131206 (build 20150802)
5	#	5	#
6	# Copyright (C) 1998-2011 Bjoern Jacke <bjoern@j3e.de>	6	# Copyright (C) 1998-2011 Bjoern Jacke <bjoern@j3e.de>
7	#	7	#
8	# License: GPLv2, GPLv3 or OASIS distribution license agreement	8	# License: GPLv2, GPLv3 or OASIS distribution license agreement
9	# There should be a copy of all of this licenses included	9	# There should be a copy of all of this licenses included
10	# with every distribution of this dictionary. Modified	10	# with every distribution of this dictionary. Modified
11	# versions using the GPL may only include the GPL	11	# versions using the GPL may only include the GPL

Use deterministic version information

- Don't generate a version number on each build

Use deterministic version information

- Don't generate a version number on each build
- Instead extract information from the source:
  - Version control system revision
  - Hash of the source code
  - Changelog entry

Example

```
VERSION=$(shell dpkg-parsechangelog | sed -n 's/^Version: *//p')

SCONSOPTS = $(SCONSFLAGS) VERSION=$(VERSION) \
  PREFIX=$(PREFIX) PREFIX_CONF=$(SYSCONF) CHMDOCS=0 \
  STRIP_CP=no \
  $(if $(findstring nostripfull,$(DEB_BUILD_OPTIONS)),STRIP_W32=no,)
```

```
18  Contents of section .dynsym:                                  18  Contents of section .dynsym:
19   4002c0 00000000 00000000 00000000 00000000  ................  19   4002c0 00000000 00000000 00000000 00000000  ................
20   4002d0 00000000 00000000 ce010000 12000000  ................  20   4002d0 00000000 00000000 ce010000 12000000  ................
Offset 15584, 15 lines modified                                    Offset 15584, 15 lines modified
15584 43cee0 2acd4300 00000000 30cd4300 00000000  *.C.....0.C.....  15584 43cee0 2acd4300 00000000 30cd4300 00000000  *.C.....0.C.....
15585 43cef0 35cd4300 00000000 05f24d00 00000000  5.C.......M.....  15585 43cef0 35cd4300 00000000 05f24d00 00000000  5.C.......M.....
15586 43cf00 3bcd4300 00000000 44cd4300 00000000  ;.C.....D.C.....  15586 43cf00 3bcd4300 00000000 44cd4300 00000000  ;.C.....D.C.....
15587 43cf10 05224e00 00000000 00000000 00000000  ."N.............  15587 43cf10 05224e00 00000000 00000000 00000000  ."N.............
15588 43cf20 4e41534d 20322e31 312e3035 00000000  NASM 2.11.05....  15588 43cf20 4e41534d 20322e31 312e3035 00000000  NASM 2.11.05....
15589 43cf30 54686520 4e657477 69646520 41737365  The Netwide Asse  15589 43cf30 54686520 4e657477 69646520 41737365  The Netwide Asse
15590 43cf40 6d626c65 7220322e 31312e30 3500004a  mbler 2.11.05..J  15590 43cf40 6d626c65 7220322e 31312e30 3500004a  mbler 2.11.05..J
15591 43cf50 756c2032 39203230 31350032 2e31312e  ul 29 2015.2.11.  15591 43cf50 756c2033 30203230 31350032 2e31312e  ul 30 2015.2.11.
15592 43cf60 30350070 6f736e20 3e3d2030 00726161  05.posn >= 0.raa  15592 43cf60 30350070 6f736e20 3e3d2030 00726161  05.posn >= 0.raa
15593 43cf70 2e630028 732d3e77 706f7320 2520732d  .c.(s->wpos %s-   15593 43cf70 2e630028 732d3e77 706f7320 2520732d  .c.(s->wpos %s-
15594 43cf80 3e656c65 6d5f6c65 6e29203d 3d203000  >elem_len) == 0.  15594 43cf80 3e656c65 6d5f6c65 6e29203d 3d203000  >elem_len) == 0.
15595 43cf90 7361612e 6300732d 3e7706f7 73203d3d  saa.c.s->wpos ==  15595 43cf90 7361612e 6300732d 3e7706f7 73203d3d  saa.c.s->wpos ==
15596 43cfa0 20732d3e 626c6b5f 6c656e00 28732d3e   s->blk_len.(s->  15596 43cfa0 20732d3e 626c6b5f 6c656e00 28732d3e   s->blk_len.(s->
15597 43cfb0 72706f73 20252073 2d3e656c 656d5f6c  rpos %s->elem_l  15597 43cfb0 72706f73 20252073 2d3e656c 656d5f6c  rpos %s->elem_l
15598 43cfc0 656e2920 3d3d2030 00732d3e 72707472  en) == 0.s->rptr  15598 43cfc0 656e2920 3d3d2030 00732d3e 72707472  en) == 0.s->rptr
Offset 63278, 18 lines modified                                    Offset 63278, 18 lines modified
63278 6f76e0 00000000 00000000 00000000 00000000  ................  63278 6f76e0 00000000 00000000 00000000 00000000  ................
63279 6f76f0 00000000 00000000 00000000 00000000  ................  63279 6f76f0 00000000 00000000 00000000 00000000  ................
63280 6f7700 a0b04300 00000000 80ae4300 00000000  ..C.......C.....  63280 6f7700 a0b04300 00000000 80ae4300 00000000  ..C.......C.....
63281 6f7710 50ae4300 00000000 00ae4300 00000000  P.C.......C.....  63281 6f7710 50ae4300 00000000 00ae4300 00000000  P.C.......C.....
63282 6f7720 10ae4300 00000000 20ae4300 00000000  ..C.......C.....  63282 6f7720 10ae4300 00000000 20ae4300 00000000  ..C.......C.....
63283 6f7730 30ae4300 00000000 40ae4300 00000000  0.C.....@.C.....  63283 6f7730 30ae4300 00000000 40ae4300 00000000  0.C.....@.C.....
63284 Contents of section .gnu_debuglink:                          63284 Contents of section .gnu_debuglink:
63285 0000 66393431 62646363 64386532 65383634  f941bdccd8e2e864  63285 0000 36626337 64653462 64346536 39326463  6bc7de4bd4e692dc
63286 0010 36346238 65613239 31386363 37313531  64b8ea2918cc7151  63286 0010 63343732 34376236 39636461 30346663  c47247b69cda04fc
63287 0020 61643235 63632e64 65627567 00000000  ad25cc.debug....  63287 0020 31303731 34382e64 65627567 00000000  107148.debug....
63288 0030 55e33e88                              U.>.              63288 0030 0afc1a7b                              ...{
63289                                                              63289
```

Don't record the current date and time

- Avoid timestamps

Don't record the current date and time

- Avoid timestamps
- If you need one:
  - Use date of last commit in VCS
  - Extract from changelog

Don't record the current date and time

- Avoid timestamps
- If you need one:
  - Use date of last commit in VCS
  - Extract from changelog
  - Don't forget the timezone

Don't record the current date and time

- Avoid timestamps
- If you need one:
  - Use date of last commit in VCS
  - Extract from changelog
  - Don't forget the timezone
- `faketime` is an option but has serious drawbacks

  `https://bugs.torproject.org/12240`

Don't record the current date and time

- Avoid timestamps
- If you need one:
  - Use date of last commit in VCS
  - Extract from changelog
  - Don't forget the timezone
- `faketime` is an option but has serious drawbacks

  `https://bugs.torproject.org/12240`
- Implement `SOURCE_DATE_EPOCH`

SOURCE_DATE_EPOCH

- What is it?
  - Environment variable with a reference time
  - Number of seconds since the Epoch (1970-01-01 00:00:00 +0000 UTC)
  - If set, replace "current time of day"
  - Implemented by `help2man`, Epydoc, Doxygen, Ghostscript (in Debian)
  - Patches ready for GCC, `txt2man`, `libxslt`, Gettext...

`https://wiki.debian.org/ReproducibleBuilds/TimestampsProposal`

SOURCE_DATE_EPOCH

- What is it?
  - Environment variable with a reference time
  - Number of seconds since the Epoch (1970-01-01 00:00:00 +0000 UTC)
  - If set, replace "current time of day"
  - Implemented by `help2man`, Epydoc, Doxygen, Ghostscript (in Debian)
  - Patches ready for GCC, `txt2man`, `libxslt`, Gettext...
- Set `SOURCE_DATE_EPOCH` in your build system

`https://wiki.debian.org/ReproducibleBuilds/TimestampsProposal`

SOURCE_DATE_EPOCH

- What is it?
  - Environment variable with a reference time
  - Number of seconds since the Epoch (1970-01-01 00:00:00 +0000 UTC)
  - If set, replace "current time of day"
  - Implemented by `help2man`, Epydoc, Doxygen, Ghostscript (in Debian)
  - Patches ready for GCC, `txt2man`, `libxslt`, Gettext...
- Set `SOURCE_DATE_EPOCH` in your build system
- Please add support in any tool writing timestamps

`https://wiki.debian.org/ReproducibleBuilds/TimestampsProposal`

data.tar.xz

data.tar

./usr/share/doc/git-all/changelog.Debian.gz

metadata

Offset 1, 1 lines modified		Offset 1, 1 lines modified	
1	gzip compressed data, was "changelog.Debian", last modified: Wed Jul 29 00:54:37 2015, max compression, from Unix	1	gzip compressed data, was "changelog.Debian", last modified: Wed Jul 29 01:03:50 2015, max compression, from Unix

./usr/share/doc/git-all/changelog.gz

metadata

Offset 1, 1 lines modified		Offset 1, 1 lines modified	
1	gzip compressed data, was "changelog", last modified: Wed Jul 29 00:54:37 2015, max compression, from Unix	1	gzip compressed data, was "changelog", last modified: Wed Jul 29 01:03:50 2015, max compression, from Unix

metadata

Offset 1, 8 lines modified		Offset 1, 8 lines modified	
1	-rwxr-xr-x root/root 0 2015-07-29 00:54:38 ./	1	-rwxr-xr-x root/root 0 2015-07-29 01:03:51 ./
2	-rwxr-xr-x root/root 0 2015-07-29 00:54:37 ./usr/	2	-rwxr-xr-x root/root 0 2015-07-29 01:03:50 ./usr/
3	-rwxr-xr-x root/root 0 2015-07-29 00:54:37 ./usr/share/	3	-rwxr-xr-x root/root 0 2015-07-29 01:03:50 ./usr/share/
4	-rwxr-xr-x root/root 0 2015-07-29 00:54:37 ./usr/share/doc/	4	-rwxr-xr-x root/root 0 2015-07-29 01:03:50 ./usr/share/doc/

Don't record current time (really)

- Archives keep modification times in metadata
- Storing a file can record build time

Example

```
tar -cf product.tar build
```

Don't record current time (really)

- Archives keep modification times in metadata
- Storing a file can record build time
- Solutions:
  - Store an arbitrary value

Example

```
tar --mtime='2015-08-13 00:00Z' -cf product.tar build
```

Don't record current time (really)

- Archives keep modification times in metadata
- Storing a file can record build time
- Solutions:
  - Store an arbitrary value
  - Pre-process file modification time

Example

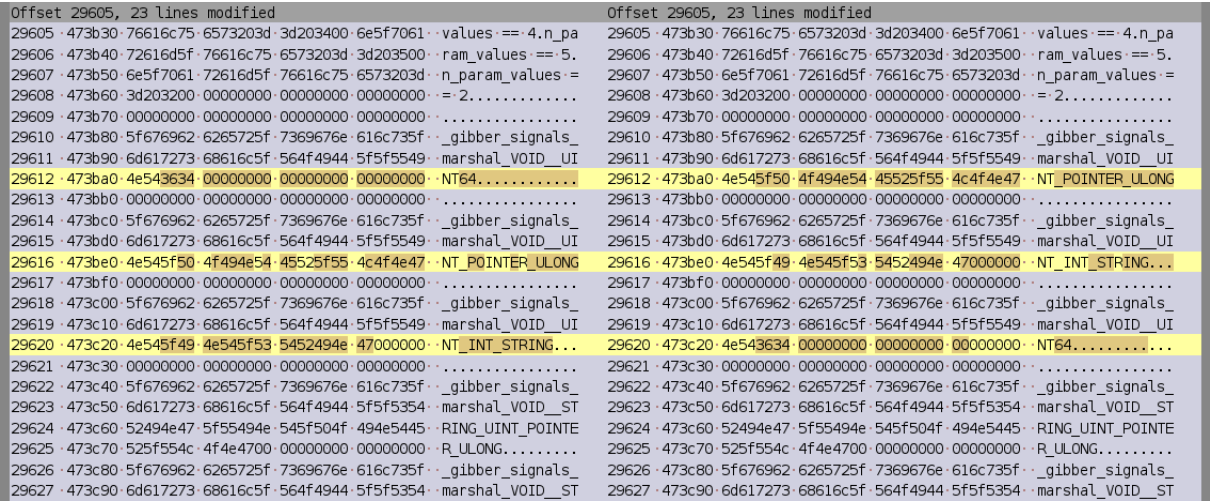
```
touch --date="2015-08-13 00:00Z" build/*
tar -cf product.tar build
```

Don't record current time (really)

- Archives keep modification times in metadata
- Storing a file can record build time
- Solutions:
  - Store an arbitrary value
  - Pre-process file modification time
  - Post-process archive

Example

```
# zip has no equivalent of --mtime
zip product.zip build
strip-nondeterminism product.zip
```



```
Offset 29605, 23 lines modified                    Offset 29605, 23 lines modified
29605 473b30 76616c75 6573203d 3d203400 6e5f7061  values == 4.n_pa  29605 473b30 76616c75 6573203d 3d203400 6e5f7061  values == 4.n_pa
29606 473b40 72616d5f 76616c75 6573203d 3d203500  ram_values == 5.  29606 473b40 72616d5f 76616c75 6573203d 3d203500  ram_values == 5.
29607 473b50 6e5f7061 72616d5f 76616c75 6573203d  n_param_values =  29607 473b50 6e5f7061 72616d5f 76616c75 6573203d  n_param_values =
29608 473b60 3d203200 00000000 00000000 00000000  = 2.............  29608 473b60 3d203200 00000000 00000000 00000000  = 2.............
29609 473b70 00000000 00000000 00000000 00000000  ................  29609 473b70 00000000 00000000 00000000 00000000  ................
29610 473b80 5f676962 6265725f 7369676e 616c735f  _gibber_signals_  29610 473b80 5f676962 6265725f 7369676e 616c735f  _gibber_signals_
29611 473b90 6d617273 68616c5f 564f4944 5f5f5549  marshal_VOID__UI  29611 473b90 6d617273 68616c5f 564f4944 5f5f5549  marshal_VOID__UI
29612 473ba0 4e543634 00000000 00000000 00000000  NT64............  29612 473ba0 4e545f50 4f494e54 45525f55 4c4f4e47  NT_POINTER_ULONG
29613 473bb0 00000000 00000000 00000000 00000000  ................  29613 473bb0 00000000 00000000 00000000 00000000  ................
29614 473bc0 5f676962 6265725f 7369676e 616c735f  _gibber_signals_  29614 473bc0 5f676962 6265725f 7369676e 616c735f  _gibber_signals_
29615 473bd0 6d617273 68616c5f 564f4944 5f5f5549  marshal_VOID__UI  29615 473bd0 6d617273 68616c5f 564f4944 5f5f5549  marshal_VOID__UI
29616 473be0 4e545f50 4f494e54 45525f55 4c4f4e47  NT_POINTER_ULONG  29616 473be0 4e545f49 4e545f53 5452494e 47000000  NT_INT_STRING...
29617 473bf0 00000000 00000000 00000000 00000000  ................  29617 473bf0 00000000 00000000 00000000 00000000  ................
29618 473c00 5f676962 6265725f 7369676e 616c735f  _gibber_signals_  29618 473c00 5f676962 6265725f 7369676e 616c735f  _gibber_signals_
29619 473c10 6d617273 68616c5f 564f4944 5f5f5549  marshal_VOID__UI  29619 473c10 6d617273 68616c5f 564f4944 5f5f5549  marshal_VOID__UI
29620 473c20 4e545f49 4e545f53 5452494e 47000000  NT_INT_STRING...  29620 473c20 4e543634 00000000 00000000 00000000  NT64............
29621 473c30 00000000 00000000 00000000 00000000  ................  29621 473c30 00000000 00000000 00000000 00000000  ................
29622 473c40 5f676962 6265725f 7369676e 616c735f  _gibber_signals_  29622 473c40 5f676962 6265725f 7369676e 616c735f  _gibber_signals_
29623 473c50 6d617273 68616c5f 564f4944 5f5f5354  marshal_VOID__ST  29623 473c50 6d617273 68616c5f 564f4944 5f5f5354  marshal_VOID__ST
29624 473c60 52494e47 5f55494e 545f504f 494e5445  RING_UINT_POINTE  29624 473c60 52494e47 5f55494e 545f504f 494e5445  RING_UINT_POINTE
29625 473c70 525f554c 4f4e4700 00000000 00000000  R_ULONG.........  29625 473c70 525f554c 4f4e4700 00000000 00000000  R_ULONG.........
29626 473c80 5f676962 6265725f 7369676e 616c735f  _gibber_signals_  29626 473c80 5f676962 6265725f 7369676e 616c735f  _gibber_signals_
29627 473c90 6d617273 68616c5f 564f4944 5f5f5354  marshal_VOID__ST  29627 473c90 6d617273 68616c5f 564f4944 5f5f5354  marshal_VOID__ST
```

Stable order for outputs

- Always output lists in the same order
- Typical issue: key order with hash tables

  `perldoc.perl.org/perlsec.html#Algorithmic-Complexity-Attacks`

Example

```
for module in dependencies.keys():
    version = dependencies[module]
    print('%s (>= %s)' % (module, version))
```

Stable order for outputs

- Always output lists in the same order
- Typical issue: key order with hash tables

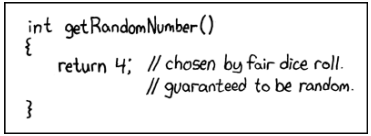
  `perldoc.perl.org/perlsec.html#Algorithmic-Complexity-Attacks`
- Sort!

Example

```
for module in sorted(dependencies.keys()):
    version = dependencies[module]
    print('%s (>= %s)' % (module, version))
```

Avoid (true) randomness

- Randomness is not deterministic



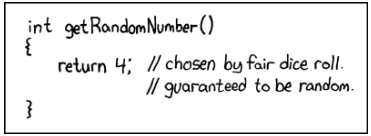
XKCD #221

Example

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

Avoid (true) randomness

- Randomness is not deterministic
- Seed your PRNG from known value
  - Use a fixed value

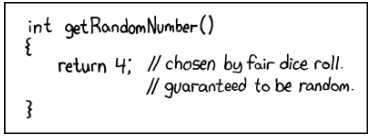


XKCD #221

Example

```
$ gcc -flto -c -frandom-seed=0 utils.c
$ nm -a utils.o | grep inline
0000000000000000 n .gnu.lto_.inline.0
```

Avoid (true) randomness



XKCD #221

- Randomness is not deterministic
- Seed your PRNG from known value
  - Use a fixed value
  - Extract from source code (filename, content hash)

Example

```
$ gcc -flto -c -frandom-seed=utils.o utils.c
$ nm -a utils.o | grep inline
0000000000000000 n .gnu.lto_.inline.a108e942
```

Define environment variable affecting outputs

- Some environment variables will affect software outputs. E.g:
  - `LC_CTIME` for time strings
  - `LC_CTYPE` for text encoding
  - `TZ` for times

Define environment variable affecting outputs

- Some environment variables will affect software outputs. E.g:
  - `LC_CTIME` for time strings
  - `LC_CTYPE` for text encoding
  - `TZ` for times
- Set them to a controlled value

Define environment variable affecting outputs

- Some environment variables will affect software outputs. E.g:
  - `LC_CTIME` for time strings
  - `LC_CTYPE` for text encoding
  - `TZ` for times
- Set them to a controlled value
- *Please don't force the language*

Stop recording build system information

- Don't record information about the build system, like:
  - date and time of the build
  - hostname
  - path
  - network configuration
  - CPU
  - environment variables
  - ...

Stop recording build system information

- Don't record information about the build system, like:
  - date and time of the build
  - hostname
  - path
  - network configuration
  - CPU
  - environment variables
  - ...
- If you really want to record them, do it outside the binaries

What's in a build environment?

- At least: build tools and their specific versions

What's in a build environment?

- At least: build tools and their specific versions
- Up to you, depending on the build system:
  - build architecture
  - operating system
  - *build path*
  - *build date and time*
  - ...

Build from source

- Build tools affecting the output from source
- Record version / tag / git commit
- Approach used by Coreboot, OpenWrt, *Tor Browser*

Reference distribution

- Use a stable distribution (e.g. Debian, CentOS)
- Record package version
- Hope the old package will stay available / record
- Approach used by Bitcoin, *Tor Browser*

Virtual machines / containers

- Using a VM saves some problems:
  - Same user
  - Same hostname
  - Same network configuration
  - *Same CPU*
  - ...
- Introduce new things that need to be trusted

Proprietary operating systems

- Cross-compiling to the rescue!
- For Windows:
  - mingw-w64: build Windows binaries on *nix
  - NSIS (Nullsoft Scriptable Install System)
- For Mac OS X:
  - Used to be quite complicated
    `https://github.com/bitcoin/bitcoin/blob/master/doc/README_osx.txt`
  - Now pretty straightforward with `crosstool-ng`
    `https://bugs.torproject.org/9711#comment:73`
  - Need a non-redistributable SDK extracted from XCode
  - `.dmg` are a bit tricky

Good ol'Makefile

- Download known toolchain archives
- Compare reference checksums
- Build and setup
- Coreboot: `make crossgcc`

Check-in everything

- Check-in all the toolchain source code in VCS
- Approach used for the base system in *BSD, and Google
- Make sure everything is checked in (*use sandbox on Linux*)
- Recently open-sourced: Bazel
  `http://bazel.io/`
- Can be hard to ask everyone to download everything all the time

Ship the toolchain as a build product

- Make the toolchain as a build product
- OpenWrt:
  `http://wiki.openwrt.org/doc/howto/obtain.firmware.sdk`

Example

```
$ wget https://downloads.openwrt.org/…/14.07/…OpenWrt-SDK-atheros-….tar.bz2
$ svn export svn://…/branches/packages_14.07/utils/xz package/xz
$ make package/xz/compile
```

Gitian

- Used by Bitcoin, Tor Browser
- Drives LXC or KVM
- "Descriptors" describing the build using:
  - Base distribution
  - Packages
  - Git remotes
  - Other input files
  - Build script

Resources

`https://gitian.org/`
`https://github.com/bitcoin/bitcoin/blob/master/doc/gitian-building.md`
`https://github.com/bitcoin/bitcoin/blob/master/contrib/gitian-descriptors/`

Docker

- Provide a way to describe specialized Linux container images
- Build in a controlled environment
- Docker images can be addressed with a hash of their content
- Bazel has support to build Docker images reproducibly

https://github.com/tianon/gosu/blob/master/Dockerfile

```
FROM golang:1.4-cross
[…]
# disable CGO for ALL THE THINGS (to help ensure no libc)
ENV CGO_ENABLED 0
COPY *.go /go/src/github.com/tianon/gosu/
WORKDIR /go/src/github.com/tianon/gosu
RUN GOARCH=amd64 go build -v -ldflags -d -o /go/bin/gosu-amd64
```

Vagrant

- Drive VirtualBox using Ruby and other scripts
- Build in a controlled environment
- Also works under OS X and Windows

`https://www.vagrantup.com/`

Debian .buildinfo

- Tie in the same file:
  - Sources
  - Generated binaries
  - Packages used to build (with specific version)
- Can be later processed to reinstall environment
- All versions are available from `snapshot.debian.org`

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),
 …
```

Testing for variations

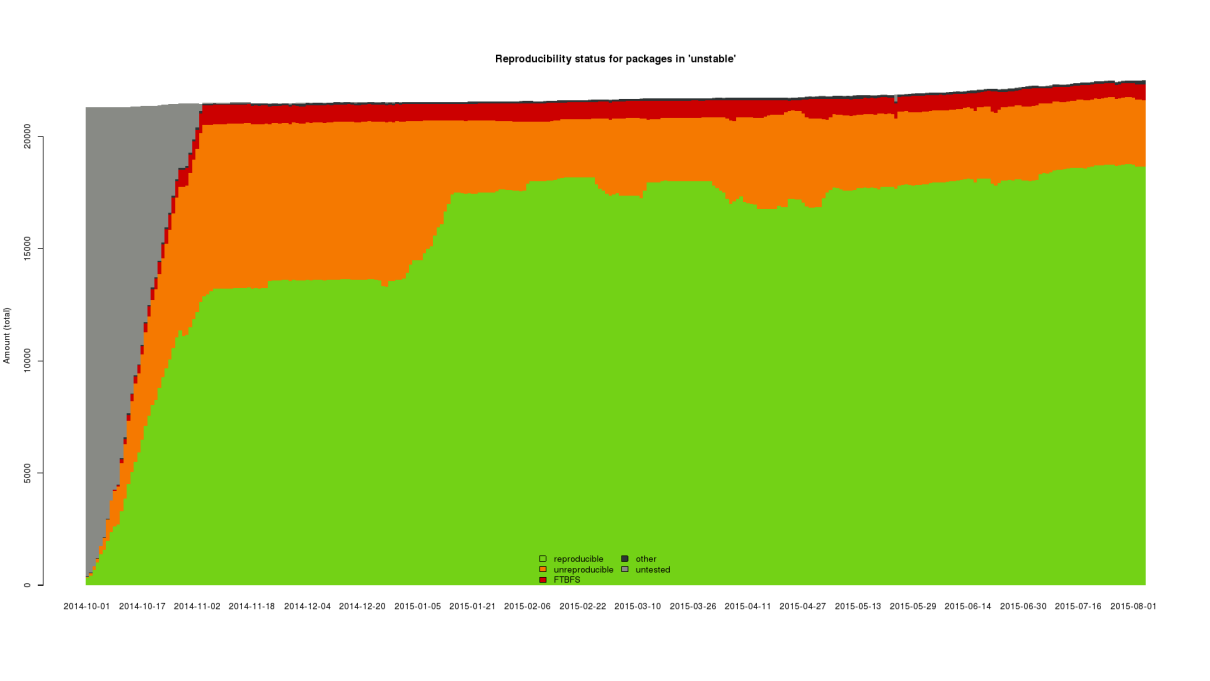
- Build a first time
- Save the result
- Perform change(s) to the environment
- Build a second time
- Compare results

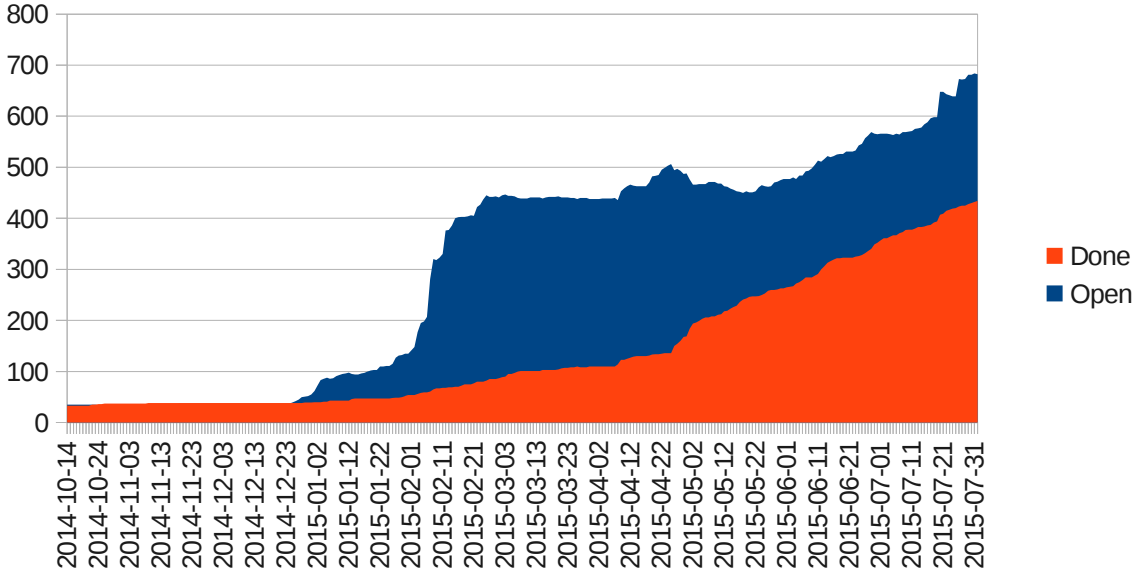
reproducible.debian.net

- Continuous test system driven by Jenkins
- Bad-ass hardware sponsored by ProfitBricks
- Tests about 1300 Debian source packages per day on average
- Results are visible on a website
- Other projects: Coreboot, OpenWrt, *yours?*

Variations on reproducible.debian.net

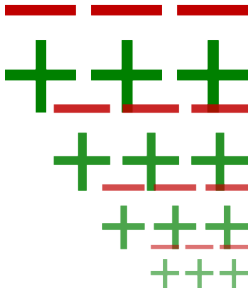
variation	first build	second build
hostname	`jenkins`	`i-capture-the-hostname`
domainname	`debian.net`	`i-capture-the-domainname`
env TZ	`GMT+12`	`GMT-14`
env LANG	`en_GB.UTF-8`	`fr_CH.UTF-8`
env LC_ALL	not set	`fr_CH.UTF-8`
env USER	`pbuilder1`	`pbuilder2`
uid	1111	2222
gid	1111	2222
UTS namespace	shared with the host	*modified using `/usr/bin/unshare --uts`*
kernel version	Linux 3.16.0-4-amd64	Linux 2.6.56-4-amd64
umask	0022	0002
CPU type	same for both builds *(work in progress)*	
year, month, date	same for both builds *(work in progress)*	
hour, minute	hour is usually the same... usually, the minute differs... *(work in progress)*	
everything else	*is likely the same...*	


Reproducibility status for packages in 'unstable'
Amount (total)
0
5000
10000
15000
20000
2014-10-01
2014-10-17
2014-11-02
2014-11-18
2014-12-04
2014-12-20
2015-01-05
2015-01-21
2015-02-06
2015-02-22
2015-03-10
2015-03-26
2015-04-11
2015-04-27
2015-05-13
2015-05-29
2015-06-14
2015-06-30
2015-07-16
2015-08-01
reproducible
unreproducible
FTBFS
other
untested


800
700
600
500
400
300
200
100
0
2014-10-14
2014-10-24
2014-11-03
2014-11-13
2014-11-23
2014-12-03
2014-12-13
2014-12-23
2015-01-02
2015-01-12
2015-01-22
2015-02-01
2015-02-11
2015-02-21
2015-03-03
2015-03-13
2015-03-23
2015-04-02
2015-04-12
2015-04-22
2015-05-02
2015-05-12
2015-05-22
2015-06-01
2015-06-11
2015-06-21
2015-07-01
2015-07-11
2015-07-21
2015-07-31
Done
Open

Debugging problems: diffoscope

- Examines differences **in depth**
- Outputs HTML or plain text showing the differences
- Recursively unpacks archives
- Seeks human readability:
  - uncompresses PDF
  - disassembles binaries
  - unpacks Gettext files
  - ... *easy to extend to new file formats*
- Falls back to binary comparison



`http://diffoscope.org/`

(formely known as `debbindiff`)

diffoscope example (HTML output)

51431	INSERT INTO "targets" VALUES('ttu.ee',13611);	51438	INSERT INTO "targets" VALUES('ttu.ee',13542);
51432	INSERT INTO "targets" VALUES('ttu.ee',13611);	51439	INSERT INTO "targets" VALUES('ttu.ee',13542);
51433	[9300 lines removed]	51440	[9314 lines removed]
60733	CREATE TABLE git_commit	60754	CREATE TABLE git_commit
60734	(git_commit TEXT);	60755	(git_commit TEXT);
60735	INSERT INTO "git_commit" VALUES('cd09fb8c2161a8d1280b848eaab3b14d35fe3044');	60756	INSERT INTO "git_commit" VALUES('e78fe5d803208bf6c877dc675cdb4f1b719e7519');
60736	COMMIT;	60757	COMMIT;

install.rdf

Offset 5, 15 lines modified		Offset 5, 15 lines modified	
5	<Description about="urn:mozilla:install-manifest">	5	<Description about="urn:mozilla:install-manifest">
6	<em:name>HTTPS-Everywhere</em:name>	6	<em:name>HTTPS-Everywhere</em:name>
7	<em:creator>Mike Perry, Peter Eckersley, & Yan Zhu</em:creator>	7	<em:creator>Mike Perry, Peter Eckersley, & Yan Zhu</em:creator>
8	<em:aboutURL>chrome://https-everywhere/content/about.xul</em:aboutURL>	8	<em:aboutURL>chrome://https-everywhere/content/about.xul</em:aboutURL>
9	<em:id>https-everywhere@eff.org</em:id>	9	<em:id>https-everywhere@eff.org</em:id>
10	<em:type>2</em:type> <!-- type: Extension -->	10	<em:type>2</em:type> <!-- type: Extension -->
11	<em:description>Encrypt the Web! Automatically use HTTPS security on many sites. </em:description>	11	<em:description>Encrypt the Web! Automatically use HTTPS security on many sites. </em:description>
12	<em:version>5.0.6</em:version>	12	<em:version>5.0.7</em:version>

diffoscope example (text output)

```
├── myspell-de-de_20131206-5_all.deb
│   ├── metadata
│   @@ -1,3 +1,3 @@
│    rw-r--r-- 0/0      4 Jun 11 16:19 2014 debian-binary
│   -rw-r--r-- 0/0    775 Jun 11 16:19 2014 control.tar.gz
│   +rw-r--r-- 0/0    777 Jun 11 16:19 2014 control.tar.gz
│    rw-r--r-- 0/0 325128 Jun 11 16:19 2014 data.tar.xz
│   ├── control.tar.gz
│   │   ├── control.tar
│   │   │   ├── md5sums
│   │   │   ⋯ Files in package differs
│   │   │
│   │
│   ├── data.tar.xz
│   │   ├── data.tar
│   │   │   ├── ./usr/share/hunspell/de_DE.aff
│   │   │    @@ -1,11 +1,11 @@
│   │   │     # this is the affix file of the de_DE Myspell dictionary
│   │   │     # derived from the igerman98 dictionary
│   │   │     #
│   │   │    -# Version: 20131206 (build 20150801)
│   │   │    +# Version: 20131206 (build 20150802)
│   │   │     #
│   │   │     # Copyright (C) 1998-2011 Bjoern Jacke <bjoern@j3e.de>
│   │   │     #
│   │   │     # License: GPLv2, GPLv3 or OASIS distribution license agreement
│   │   │     # There should be a copy of all of this licenses included
│   │   │     # with every distribution of this dictionary. Modified
│   │   │     # versions using the GPL may only include the GPL
```

strip-nondeterminism

- Normalizes various file formats
- Currently handles:
  - ar archives (`.a`)
  - gzip
  - Java jar
  - Javadoc HTML
  - Maven `pom.properties`
  - PNG
  - ZIP archives
  - ... *extensible to new formats*
- Written in Perl (like `dpkg-dev`)

`git://git.debian.org/reproducible/strip-nondeterminism.git`

Resources

- Reproducible Builds HOWTO (*work in progress*)
  `https://reproducible.debian.net/howto/`

Resources

- Reproducible Builds HOWTO (*work in progress*)
  `https://reproducible.debian.net/howto/`
- Debian “Reproducible Builds” wiki
  `https://wiki.debian.org/ReproducibleBuilds`

Resources

- Reproducible Builds HOWTO (*work in progress*)
  `https://reproducible.debian.net/howto/`
- Debian “Reproducible Builds” wiki
  `https://wiki.debian.org/ReproducibleBuilds`
- Diverse Double-Compilation
  `http://www.dwheeler.com/trusting-trust/`

Thanks!

- Debian “Reproducible Builds” team
  (you are just **so** awesome!)
- Mike Perry, Georg Koppen, David A. Wheeler
- Linux Foundation and the Core Infrastructure Initiative

lunar@debian.org 0603 CCFD 9186 5C17 E88D
4C79 8382 C95C 2902 3DF9

clothes: Elhonna Sombrefeuille — hair: igor