GNU Mes – Scheme-only bootstrap

janneke@gnu.org

FOSDEM’20

2020-02-02
Outline

1 Introduction

2 Reproducibility

3 Bootstrappability

4 Thanks
Why bootstrapping is important to you.

or

Why bootstrapping is something you wish to ignore.
GNU Mes

- A Scheme interpreter written in ~5,000LOC of simple C.
- A C compiler written in Scheme.
- Built on LISP: eval/apply, the Maxwell Equations of Software.
(define (apply fn x a)
  (cond
    ((atom fn)
      (cond
        ((eq fn CAR) (caar x))
        ((eq fn CDR) (cdar x))
        ((eq fn CONS) (cons (car x) (cadr x)))
        ((eq fn ATOM) (atom (car x)))
        ((eq fn EQ) (eq (car x) (cadr x)))
        (#t (apply (eval fn a) x a)))))
    ((eq (car fn) LAMBDA)
      (eval (caddr fn) (pairlis (cadr fn) x a)))
    ((eq (car fn) LABEL)
      (apply (caddr fn) x
        (cons (cons (cadr fn) (caddr fn)) a)))))

(define (eval e a)
  (cond
    ((atom e) (cdr (assoc e a)))
    ((atom (car e))
      (cond
       ((eq (car e) QUOTE) (cadr e))
       ((eq (car e) COND) (evcon (cdr e) a))
       (#t (apply (car e) (evlis (cdr e) a) a)))))
    (#t (apply (car e) (evlis (cdr e) a) a))))

eval and apply are mutual recursing functions that—using a few helper functions—describe the universe of computing.
Long path: Best Practice

- 500+ MB: no bootstrap
Guix
Pronounced Geeks
Reduce binary seeds to bare minimum

These big chunks of binary code are practically non-auditable which breaks the source to binary transparency that we get in the rest of the package dependency graph.

Every unauditable binary leaves us vulnerable to compiler backdoors as described by Ken Thompson in the 1984 paper Reflections on Trusting Trust.

Thus, our goal is to reduce the set of bootstrap binaries to the bare minimum. – Ludovic Courtès (GNU Guix documentation, December 2017)
Reflections on Trusting Trust

To what extent should one trust a statement that a program is free of Trojan horses? Perhaps it is more important to trust the people who wrote the software.
500+ MB: no bootstrap
- 500+ MB: no bootstrap
- 252 MB: GNU Guix System v1.0
Bitcoin Build System Security

Carl Dong, Chaincode Labs
What is a Bootstrap?

Impossible task: pull yourself up on your boot straps

Software: to create your first: kernel, shell, C compiler, ...

GCC

source + ?? = GCC

binary
Recipe for yoghurt: Add yoghurt to milk – Anonymous
How to Bootstrap: Create your second GCC

Traditional recipe: like yoghurt

\[ \text{source} + \text{GCC} - 1 = \text{binary} \]

... and done!
Pour milk
Add yoghurt
We’re Reproducible!
Add evil yoghurt
We’re Reproducible!
Evil yoghurt
We’re Reproducibly Malicious
Reproducibility is not enough
Reproducibility plus clean source code is not enough
Guix
Pronounced Geeks
Long path: Reduced Binary Seed bootstrap

- **500+ MB**: no bootstrap
- **252 MB**: GNU Guix System v1.0
- **145 MB**: Reduced Binary Seed
  - master branch
  - GCC, GLIBC, Binutils
  - + MesCC-Tools, + Mes
Long path: Scheme-only bootstrap

- **500+ MB**: no bootstrap
- **252 MB**: GNU Guix System v1.0
- **145 MB**: Reduced Binary Seed
  - master branch
  - GCC, GLIBC, Binutils
  - + MesCC-Tools, + Mes
- **57 MB**: Scheme-only
  - wip-bootstrap branch
  - Awk, Bash, Bzip2, GNU Core Utilities, Grep, Gzip, Make, Patch, Sed, Tar, and XZ.
  - + Gash (source only!)
**Scheme-only bootstrap: Gash Core Utils**

<table>
<thead>
<tr>
<th>awk</th>
<th>cp</th>
<th>gash</th>
<th>mv</th>
<th>sleep</th>
<th>uname</th>
</tr>
</thead>
<tbody>
<tr>
<td>basename</td>
<td>cut</td>
<td>grep</td>
<td>pwd</td>
<td>sort</td>
<td>uniq</td>
</tr>
<tr>
<td>bash</td>
<td>diff</td>
<td>gzip</td>
<td>reboot</td>
<td>tar</td>
<td>wc</td>
</tr>
<tr>
<td>cat</td>
<td>dir</td>
<td>head</td>
<td>rm</td>
<td>test</td>
<td>which</td>
</tr>
<tr>
<td>chmod</td>
<td>expr</td>
<td>ln</td>
<td>rmdir</td>
<td>touch</td>
<td></td>
</tr>
<tr>
<td>cmp</td>
<td>false</td>
<td>ls</td>
<td>sed</td>
<td>tr</td>
<td></td>
</tr>
<tr>
<td>compress</td>
<td>find</td>
<td>mkdir</td>
<td>sh</td>
<td>true</td>
<td></td>
</tr>
</tbody>
</table>
Cross distro reproducibility

The sha256sum for bin/mes-mescc on x86 shall be

722790ed261954eb53cf2cd2906c89c7589ef72b66171bbe2a9dce0f0af20232 v0.22
9e0bcb1633c58e7bc415f6ea27cee7951d6b0658e13cdc147e992b31a14625fb v0.21

only differing in the version number string.

For v0.21 this has been verified on Guix System, Debian GNU/Linux and NixOS.
The holy grail of bootstrappability will be connecting mes to hex0.
– Carl Dong, Chaincode Labs
Long path: Full Source Bootstrap

- **500+ MB**: no bootstrap
- **252 MB**: GNU Guix System v1.0
- **145 MB**: Reduced Binary Seed
  - master branch
  - GCC, GLIBC, Binutils
  - + MesCC-Tools, + Mes
- **57 MB**: Scheme-only
  - wip-bootstrap branch
  - Awk, Bash, Bzip2, GNU Core Utilities, Grep, Gzip, Make, Patch, Sed, Tar, and XZ.
  - + Gash (source only!)
- **357 bytes**: Full Source
  - MesCC-Tools, Mes
  - + Stage0: 357 bytes (x86)
Trusted Computing Base

- Source code
- Binary seeds
- Guix System
- Linux
I want code easy to reason about at the heart of this bootstrap, so that everyone will be able to sit down in the morning and be done by lunch time; understanding how every piece of it works. – Jeremiah Orians
Won’t your life be boring?

MesCC should optimize for the ease of convincing us of its correctness. – Mark H Weaver
Vulnerability to a trusting trust attack is a symptom of an unauditable or missing bootstrap story. – janneke
Thanks

- Carl Dong
- Danny Milosavljevic
- David Terry
- Jeremiah Orians
- Ludovic Courtès
- Matt Wette
- Pjotr Prins
- Rutger van Beusekom
- Timothy Sample
- Vagrant Cascadian
You can help

- make Guix run on Mes
- write a bootstrappable syntax-case
- simplify MesCC and target GCC-4.6
- bootstrap NixOS, Debian
- port MesCC to the Hurd, FreeBSD
- spread the message
- retweet @janneke-gnu janneke@octodon.social

Connect

- irc freenode.net #bootstrappable #guix
- mail bug-mes@gnu.org guix-devel@gnu.org
- git https://git.savannah.gnu.org/git/mes.git
- web bootstrappable.org