Breaking the Chains of Trusting Trust

Vagrant Cascadian <vagrant@reproducible-builds.org>

BSidesPDX 2022

Who am I

	Vagrant
debian user	2001
debian developer	2010
reproducible builds	2015

Trust

Different levels of trust:

- curl `http://example.net/hackme` | sudo sh

Trust

Different levels of trust:

- curl `http://example.net/hackme` | sudo sh
- curl –proto '=https' –tlsv1.2 -sSf `https://sh.rustup.rs` | sh

Trust

Different levels of trust:

- curl `http://example.net/hackme` | sudo sh
- curl –proto '=https' –tlsv1.2 -sSf `https://sh.rustup.rs` | sh
- download file, verify signature ... run code

Trust

Different levels of trust:

- curl `http://example.net/hackme` | sudo sh
- curl –proto '=https' –tlsv1.2 -sSf `https://sh.rustup.rs` | sh
- download file, verify signature ... run code
- download source, verify signature, compile from source

Trust

Different levels of trust:

- curl `http://example.net/hackme` | sudo sh
- curl –proto '=https' –tlsv1.2 -sSf `https://sh.rustup.rs` | sh
- download file, verify signature ... run code
- download source, verify signature, compile from source
- emerge –emptytree @world

Trust

Different levels of trust:

- curl `http://example.net/hackme` | sudo sh
- curl –proto '=https' –tlsv1.2 -sSf `https://sh.rustup.rs` | sh
- download file, verify signature ... run code
- download source, verify signature, compile from source
- emerge –emptytree @world
- rewrite everything in assembly

Trust

Different levels of trust:

- curl `http://example.net/hackme` | sudo sh
- curl –proto '=https' –tlsv1.2 -sSf `https://sh.rustup.rs` | sh
- download file, verify signature ... run code
- download source, verify signature, compile from source
- emerge –emptytree @world
- rewrite everything in assembly
- build it up from transitors

Trust

Different levels of trust:

- curl `http://example.net/hackme` | sudo sh
- curl –proto '=https' –tlsv1.2 -sSf `https://sh.rustup.rs` | sh
- download file, verify signature ... run code
- download source, verify signature, compile from source
- emerge –emptytree @world
- rewrite everything in assembly
- build it up from transitors
- I have a beach, some wood, abundant sunshine, and a lot of time

Trusting Trust

Ken Thompson
Reflections on trusting trust, 1984
https://archive.org/details/reflections-on-trusting-trust

The Moral of Trusting Trust

"You can't trust code that you did not totally create yourself.
(Especially code from companies that employ people like me.)
No amount of source-level verification or scrutiny will protect you
from using untrusted code." - Ken Thompson

Did I say 1984, I meant 1974

Karger, 1974
"... insert a trap door into the... compiler...
the trap door can maintain itself,
even when the compiler is recompiled"

Decades of Trust

Since 1974

- 1984: Reflections on trusting trust

Decades of Trust

Since 1974

- 1984: Reflections on trusting trust
- 1980s: some papers about compiling multiple times

Decades of Trust

Since 1974

- 1984: Reflections on trusting trust
- 1980s: some papers about compiling multiple times
- 1990s ... usenet post mumbling about multiple compilers

Decades of Trust

Since 1974

- 1984: Reflections on trusting trust
- 1980s: some papers about compiling multiple times
- 1990s ... usenet post mumbling about multiple compilers
- 2000s: some more papers about compiling multiple times

Decades of Trust

Since 1974

- 1984: Reflections on trusting trust
- 1980s: some papers about compiling multiple times
- 1990s ... usenet post mumbling about multiple compilers
- 2000s: some more papers about compiling multiple times
- 2005: Countering Trusting Trust through Diverse Double-Compiling

Decades of Trust

Since 1974

- 1984: Reflections on trusting trust
- 1980s: some papers about compiling multiple times
- 1990s ... usenet post mumbling about multiple compilers
- 2000s: some more papers about compiling multiple times
- 2005: Countering Trusting Trust through Diverse Double-Compiling
- 2009: Fully Countering Trusting Trust through Diverse Double-Compiling

Decades of Trust

Since 1974

- 1984: Reflections on trusting trust
- 1980s: some papers about compiling multiple times
- 1990s ... usenet post mumbling about multiple compilers
- 2000s: some more papers about compiling multiple times
- 2005: Countering Trusting Trust through Diverse Double-Compiling
- 2009: Fully Countering Trusting Trust through Diverse Double-Compiling
- ... and some high profile compromises!

XcodeGhost or should we say Strawhorse?

XcodeGhost, 2015

- Modified version of Apple's Xcode

XcodeGhost or should we say Strawhorse?

XcodeGhost, 2015

- Modified version of Apple's Xcode
- Over 4000 compromised apps

SolarWhat?

SolarWinds, 2020

- Compromised build server...

SolarWhat?

SolarWinds, 2020

- Compromised build server...
- ...via weak and/or leaked passphrases

SolarWhat?

SolarWinds, 2020

- Compromised build server...
- ...via weak and/or leaked passphrases
- signing certificates compromised

SolarWhat?

SolarWinds, 2020

- Compromised build server...
- ...via weak and/or leaked passphrases
- signing certificates compromised
- possibly 18000 affected installations

The price of Trust

What is the Price...
Of Trusting Trust?

Reproducible Builds

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

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

Building on a solid foundation of turtles

`https://bootstrappable.org`
Compiling your C compiler with a C compiler
And a C compiler to compile the other C compiler
...Ad infinitum
Or any other language (rust, java, haskell, etc.)

Java bootstrap

Java bootstrapping

- openjdk17 needs...

Java bootstrap

Java bootstrapping

- openjdk17 needs...
- openjdk16 which needs...

Java bootstrap

Java bootstrapping

- openjdk17 needs...
- openjdk16 which needs...
- ...

Java bootstrap

Java bootstrapping

- openjdk17 needs...
- openjdk16 which needs...
- ...
- openjdk9 ... etc.

Rust bootstrap

Rust bootstrapping

- rust 1.64 needs...

Rust bootstrap

Rust bootstrapping

- rust 1.64 needs...
- rust 1.63 which needs...

Rust bootstrap

Rust bootstrapping

- rust 1.64 needs...
- rust 1.63 which needs...
- ...

Rust bootstrap

Rust bootstrapping

- rust 1.64 needs...
- rust 1.63 which needs...
- ...
- rust 1.54 can be built with mrustc

Rust bootstrap

Rust bootstrapping

- rust 1.64 needs...
- rust 1.63 which needs...
- ...
- rust 1.54 can be built with mrustc
- mrustc is written in C++

Diverse Double Compiling

David A. Wheeler
Fully Countering Trusting Trust through Diverse Double-Compiling, 2009
`https://dwheeler.com/trusting-trust/dissertation/html/wheeler-trusting-trust-ddc.html`

A beautiful Mes

GNU Mes is a Scheme interpreter and C compiler for bootstrapping the GNU System.
`https://www.gnu.org/software/mes/`

We made the same Mes

Bit-for-bit identical Mes built on three different distributions
https://reproducible-builds.org/news/2019/12/21/reproducible-bootstrap-of-mes-c-compiler/

Beginning with a Mes

GNU Guix: The Reduced Binary Seed Bootstrap
`https://guix.gnu.org/en/manual/devel/en/guix.html#Reduced-Binary-Seed-Bootstrap`

- ...

Beginning with a Mes

GNU Guix: The Reduced Binary Seed Bootstrap
`https://guix.gnu.org/en/manual/devel/en/guix.html#Reduced-Binary-Seed-Bootstrap`

- ...
- Reduced to 145MB of bootstrap binaries (from 250MB)

Beginning with a Mes

GNU Guix: The Reduced Binary Seed Bootstrap
`https://guix.gnu.org/en/manual/devel/en/guix.html#Reduced-Binary-Seed-Bootstrap`

- ...
- Reduced to 145MB of bootstrap binaries (from 250MB)
- Using Mes and guile...

Beginning with a Mes

GNU Guix: The Reduced Binary Seed Bootstrap
`https://guix.gnu.org/en/manual/devel/en/guix.html#Reduced-Binary-Seed-Bootstrap`

- ...
- Reduced to 145MB of bootstrap binaries (from 250MB)
- Using Mes and guile...
- Builds from source GCC, binutils, glibc, etc.

Beginning with a Mes

GNU Guix: The Reduced Binary Seed Bootstrap
`https://guix.gnu.org/en/manual/devel/en/guix.html#Reduced-Binary-Seed-Bootstrap`

- ...
- Reduced to 145MB of bootstrap binaries (from 250MB)
- Using Mes and guile...
- Builds from source GCC, binutils, glibc, etc.
- 145MB of binaries is still not really auditable...

Before The Mes and Beyond

GNU Guix: The Full-Source Bootstrap
Now available in the "core-updates" branch!

- hex0 (357-byte binary)

Before The Mes and Beyond

GNU Guix: The Full-Source Bootstrap
Now available in the "core-updates" branch!

- hex0 (357-byte binary)
- hex1

Before The Mes and Beyond

GNU Guix: The Full-Source Bootstrap
Now available in the "core-updates" branch!

- hex0 (357-byte binary)
- hex1
- M0

Before The Mes and Beyond

GNU Guix: The Full-Source Bootstrap
Now available in the "core-updates" branch!

- hex0 (357-byte binary)
- hex1
- M0
- hex2

Before The Mes and Beyond

GNU Guix: The Full-Source Bootstrap
Now available in the "core-updates" branch!

- hex0 (357-byte binary)
- hex1
- M0
- hex2
- M1

Before The Mes and Beyond

GNU Guix: The Full-Source Bootstrap
Now available in the "core-updates" branch!

- hex0 (357-byte binary)
- hex1
- M0
- hex2
- M1
- mescc-tools

Before The Mes and Beyond

GNU Guix: The Full-Source Bootstrap
Now available in the "core-updates" branch!

- hex0 (357-byte binary)
- hex1
- M0
- hex2
- M1
- mescc-tools
- M2-Planet

Before The Mes and Beyond

GNU Guix: The Full-Source Bootstrap
Now available in the "core-updates" branch!

- hex0 (357-byte binary)
- hex1
- M0
- hex2
- M1
- mescc-tools
- M2-Planet
- Mes

Before The Mes and Beyond

GNU Guix: The Full-Source Bootstrap
Now available in the "core-updates" branch!

- hex0 (357-byte binary)
- hex1
- M0
- hex2
- M1
- mescc-tools
- M2-Planet
- Mes
- TinyCC (patched)

Before The Mes and Beyond

GNU Guix: The Full-Source Bootstrap
Now available in the "core-updates" branch!

- hex0 (357-byte binary)
- hex1
- M0
- hex2
- M1
- mescc-tools
- M2-Planet
- Mes
- TinyCC (patched)
- old versions of GCC, binutils, glibc, gzip, tar ...

Before The Mes and Beyond

GNU Guix: The Full-Source Bootstrap
Now available in the "core-updates" branch!

- hex0 (357-byte binary)
- hex1
- M0
- hex2
- M1
- mescc-tools
- M2-Planet
- Mes
- TinyCC (patched)
- old versions of GCC, binutils, glibc, gzip, tar ...
- modern GCC and everything

Make it live

`https://github.com/fosslinux/live-bootstrap`

- A live environment

Make it live

`https://github.com/fosslinux/live-bootstrap`

- A live environment
- From kernel and a bit of source code

Make it live

`https://github.com/fosslinux/live-bootstrap`

- A live environment
- From kernel and a bit of source code
- To a reproducibly bootstrapped toolchain

Make it live

`https://github.com/fosslinux/live-bootstrap`

- A live environment
- From kernel and a bit of source code
- To a reproducibly bootstrapped toolchain
- no pregenerated "source" code shortcuts

Make it live

`https://github.com/fosslinux/live-bootstrap`

- A live environment
- From kernel and a bit of source code
- To a reproducibly bootstrapped toolchain
- no pregenerated "source" code shortcuts
- work-in-progress, but a lot of progress!

UEFI based bootstrap

Work-in-progress UEFI bootstrap
`https://git.stikonas.eu/andrius/stage0-uefi`

Bare Metal Bootstrap

Stage0 on Bare Metal?
`https://git.savannah.nongnu.org/cgit/stage0.git/tree/`

Freedom in your bits and bytes

Free/Libre and Open Source Software
Allows arbitrary third-party verification

Forget Trust, Verify

No need to Trust, All you need is:

- Free/Libre and Open Source Software

Forget Trust, Verify

No need to Trust, All you need is:

- Free/Libre and Open Source Software
- Reproducible Builds

Forget Trust, Verify

No need to Trust, All you need is:

- Free/Libre and Open Source Software
- Reproducible Builds
- Bootstrapping

Forget Trust, Verify

No need to Trust, All you need is:

- Free/Libre and Open Source Software
- Reproducible Builds
- Bootstrapping
- Diverse compilation

Forget Trust, Verify

No need to Trust, All you need is:

- Free/Libre and Open Source Software
- Reproducible Builds
- Bootstrapping
- Diverse compilation
- ... and lots of compile cycles

Copyright and attributions