Pwned at Birth

Issues of a Poisoned Tool-Chain

Randy Bush <randy@psg.com>

The Tool Chain

When my laptop's fan goes on, I think it is the NSA, GCHQ, Israelis, Chinese, … fighting to see who will own me today

Our Laptops

That companies with security as part of their mission have all their engineers walking around with unassured laptops scares the heck out of me

The Tool-Chain

When constructing assurance-critical tools, we need to maximize assurance in the tools used to build them

We have NO ASSURANCE of our tool set, from CPU to Compiler to Libraries to Kernel to …

Some of the Fears

- ToolChain Poisoning
- Device Poisoning
- Side-Channel Attacks
- How can you tell if your vendor actually implemented CrypTech, and correctly?

The Compiler

- Ken Thompson's 1984 Turing Award paper *Reflections on Trusting Trust*

- A self-reproducing trap in the C compiler which "would match code in the UNIX "login" command. The replacement code would miscompile the login command so that it would accept either the intended encrypted password or a particular known password." You have been owned!

Double-Diverse Compilation

- In his 2009 PhD dissertation, David Wheeler explained how to counter the "trusting trust" attack by using the "Diverse Double-Compiling" (DDC) technique
- We can use this on GCC and clang to get somewhat assured compilers
- But you still have to inspect the source!

Critical Tool-Chain

- C compilers audited and built using DCC
- Audited kernel, libc, …
- Audited whole darn UNIX or Linux
- Audited Verilog compiler
- Audited FPGA download tools
- Audited test tools
- Trojan prevention & detection

First Stage Build

Make Safe Compilers through Double Diverse Compilation

Build Behind an Air Gap

In a Clean Room (no USBs, no WiFi, …)

Don't Trust the CPU?

Find ten year old 486 boxes and Double Diverse cross-compile

But you Must Audit the Source

Auditing

- It's not just the compiler
- Or the base system components
- It's also your Applications
- How you package and distribute
- …
- Auditing is likely to be as big a job as producing the code

HDL / Verilog

- But FPGAs/ASICs are programmed in a Hardware Definition Language, Verilog
- It is very hard to get an open Verilog compiler
- Verilog can not compile itself, so DDC is not applicable here, just a DCC C compiler
- We are working on methods of gaining trust in the FPGA tool chain

What Level Are We Prepared to Trust?

Funding From

A Few Private Donations