Building a Trustworthy Computing Platform

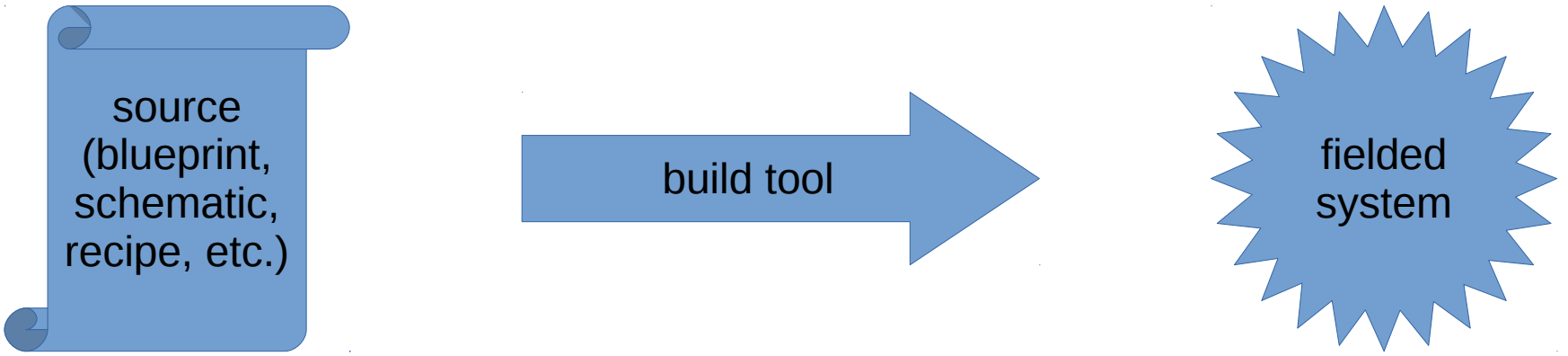
Gabriel L. Somlo, Ph.D.
<glsomlo@cert.org>

SEI, CERT Division
Carnegie Mellon University
Pittsburgh, PA 15213

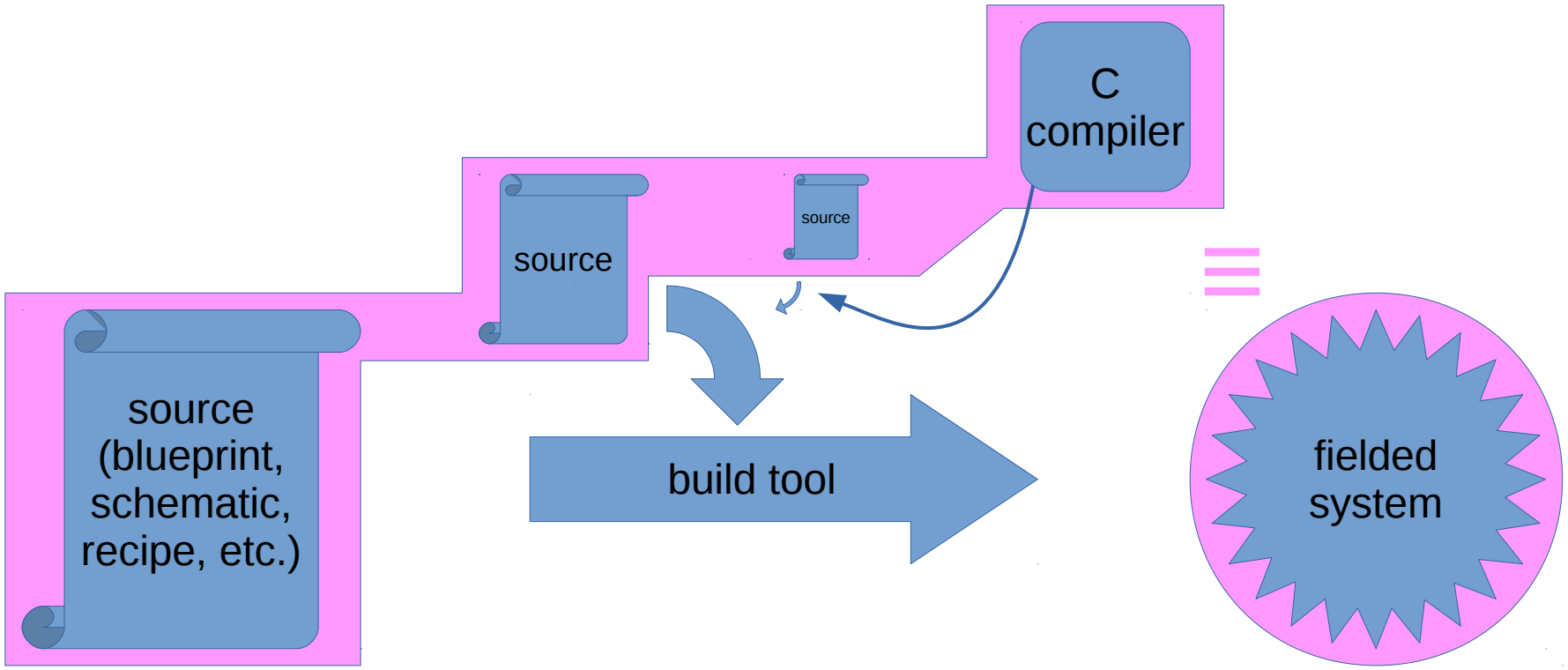
This material is based upon work funded and supported by the Department of Defense under Contract No. FA8702-15-D-0002 with Carnegie Mellon University for the operation of the Software Engineering Institute, a federally funded research and development center.

DM19-0566

Trust Anchors for Fielded Systems



Trust Anchors for Fielded Systems



Trusting Trust: Problem and Solution

- Self-propagating compiler hack (Ken Thompson)
  - Malicious C compiler inserts Trojan during *victim program* build
    - Clean source → malicious binary
      - Including *compiler's own* sources!
    - Compiler source hack *no longer needed* after 1st iteration!
- David A. Wheeler's defense: Diverse Double Compilation
  - Suspect compiler A: source S_A, binary B_A
  - Trusted compiler T: binary B_T

  $S_A \rightarrow B_A \rightarrow X$ $\qquad\qquad$ $S_A \rightarrow B_T \rightarrow Y$

    - X and Y are functionally identical, but different binaries

  $S_A \rightarrow X \rightarrow X_1$ $\qquad\qquad$ $S_A \rightarrow Y \rightarrow Y_1$

    - X_1 and Y_1 must be identical binaries (since X, Y functionally identical)

Recommendations

- Retain ability to *field strip* our cyber-weapons!
  - Require capability to rebuild system from *sources*
    - *Including* tool chain sources: HDL & software compilers!
  - Show of *good faith* from upstream supplier(s)
  - Built-in sustainment capability from *day one*
    - Solve “Trusting Trust” concerns
      - Available source code (to everything) acting as trust anchor

Bootstrapping a Trustworthy Platform

- Use DDC to obtain a clean C [cross-]compiler
- [Cross-]compile HDL compiler toolchain
- Cross-compile target OS (kernel, glibc, utilities)
- Build FPGA bitstream with HDL toolchain
- Boot target OS on FPGA
  - Self-hosting from this point forward
    - Any system component can be (re)built on the system itself!
  - Trust anchor: the cumulative set of source code
    - HDL, OS (kernel, glibc, utilities), and Compilers (C & HDL)

List of Ingredients

- FPGA development board
  - Lattice ECP5 Versa: LFE5UM5G-45F-VERSA
- Free/Open HDL (Hardware Description Language) toolchain
  - Verilog front-end: https://github.com/YosysHQ/yosys
  - ECP5 device db. & bitstream tools: https://github.com/SymbiFlow/prjtrellis
  - Place & Route back-end: https://github.com/YosysHQ/nextpnr
- Free/Open 64-bit CPU (RISC-V ISA)
  - RocketChip: https://github.com/freechipsproject/rocket-chip
- Free/Open System-on-Chip (SoC) environment (sys. bus & peripherals)
  - LiteX: https://github.com/enjoy-digital/litex
- Software stack (Linux, GCC, glibc)
  - Fedora: https://fedoraproject.org/wiki/Architectures/RISC-V

Simplified Computer Architecture