Sec: Untrusted Code, Security and Proofs

On-going Work in the PLS Group

Nicolas Magaud
School of Computer Science and Engineering
The University of New South Wales

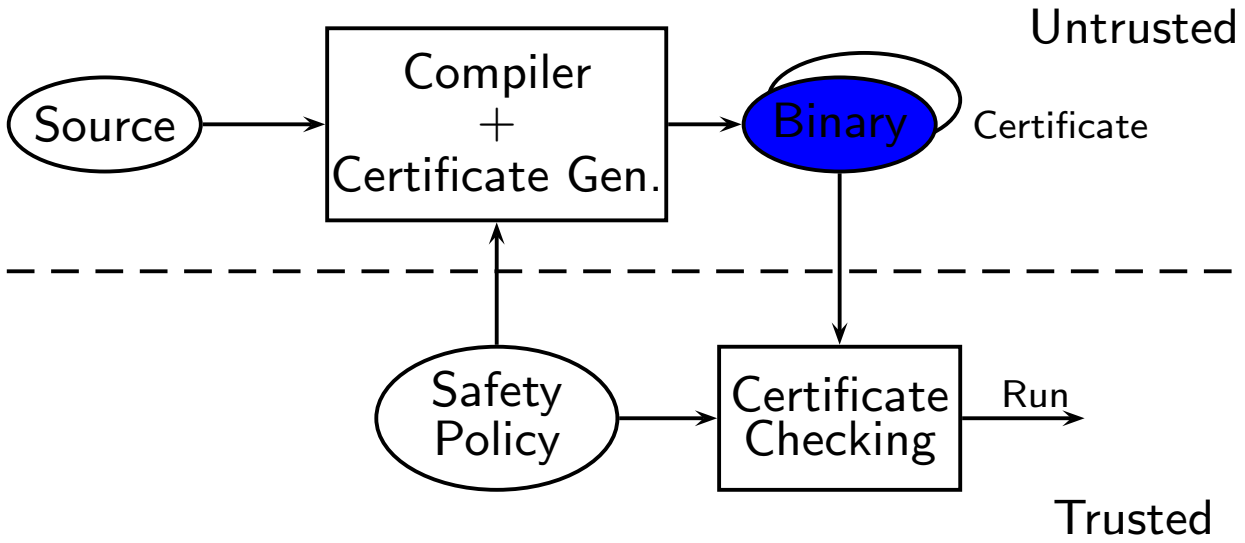
March 30th, 2004
ERTOS seminar

Security and Untrusted Code

- Unknown Programs are not Safe to Run !
  - They can be either malicious or just buggy.
  - How to Trust a Program ?
- Solutions
  - Certificate-Bearing Code (a.k.a. Proof-Carring Code)
  - Execution Monitors (and OS-related techniques)

Proof-Carrying Code

- Infrastructure



- Certificates (and Proofs) about what the program does not do...
- Type Safety, Memory Safety, etc.

The Sec Approach

- A Multi-language Secure Framework
- A (Functional) Intermediate Language with Dependent Types
  - Architecture Neutral, Low Level, Certified Code
  - based on the ANF intermediate language
  - and the logic of LF (with dependent types to encode proofs)
- How to ensure Reliability ?
  - The smaller the TCB, the safer the system.
  - Certifying Compilers:
    Compiling High-level Languages and the attached Proofs
  - OS Protection Mechanisms: Enforcing time and space bounds

On-going Research (I)

- Description of the syntax, semantics and type system for *Seccode*
  - On-paper Description
  - Implementing Examples
  - A tool to transform *Seccode* proof annotations into LF (*secpf*)
- Formal Description in LF:
  - types and terms
  - small-step semantics (transition relation)
  - type system
  - properties: preservation and progress
- How about our LF-like Framework for Proofs ? Is it Sound ?

On-going Research (II)

- *Seccode* to Real-World Intel (i386) Machine Code
  - Part of the TCB
  - Code generation (Register Allocation, Optimizations)
  - Pushing optimisations back into $Seccode$
  - Removing it from the TCB
- A Formal Machine Model with Multiple Regions
  - Protection Domains and Shared Regions
  - Machine Model with Access Rights in Twelf
  - Formalizing Operations Systems Related Features

Plan for the Future

- Implementing a Real-World Practicable Framework
- Investigating Further Language-Based Techniques
- Reconciling OS and Language Based Techniques
- Making Concrete Experiments to Assess the Framework
- Any Other Suggestions ?