Information Flow Systems

Promise: The information flow guarantees of security-typed languages provide a practical avenue to security by producing proofs of compliance with a specified policy.

Problem: Such languages are simply a tool for restricting information flow through source-code annotations: there is little present theory or practice to indicate how such annotations can be used to implement security in real systems.

Challenge: how to integrate information flow features with existing systems interfaces?

Approach

Implementing High-Level Policy

Extended Runtime Information Flow

Instrumenting Legacy Systems

(1) Trusted Declassification

(2) Automated Instrumentation

(3) Policy Equivalence

(4) Bridging the Semantic Gap

(5) Extended Labeling

(6) Universal Principals

(7) Dynamic Policy Store

(8) Downgrade Control Flow

(9) Guided Inference

(10) Optimal Declassification

(11) Label Refactoring

Impact: Applications can build upon guarantees provided by the compiler to ensure end-to-end security. This can certify a system’s compliance with local, organizational, or regulatory security policies.

Technical approach: this work fills the gaps between practice and theory by identifying the compile- and run-time tools and services needed to enforce the system and environment policy (see figure).

Result Highlights:

• Channels - integrating application IF labels with system policy
• Trusted declassification - tightening downgrade control flow
• Compliance - testing application and system policy equivalence
• JifClipse - an flow integrated development environment
• JPMail - a working large scale IF software system