Add Context to Inferring Risks from Mobile Application

Diebold Carter, Patrick Darin, Kinser Jessica,
MacCauley Sean,
Debin Liu, and Xiaoyong Zhou

Background

- Android (Google)
  - a widely anticipated open source operating system for mobile devices
  - it provides base operation system, application middleware layer, Java software development kit and a collection of system applications
- It uses a simple permission label assignment model to restrict access to resources and other applications.

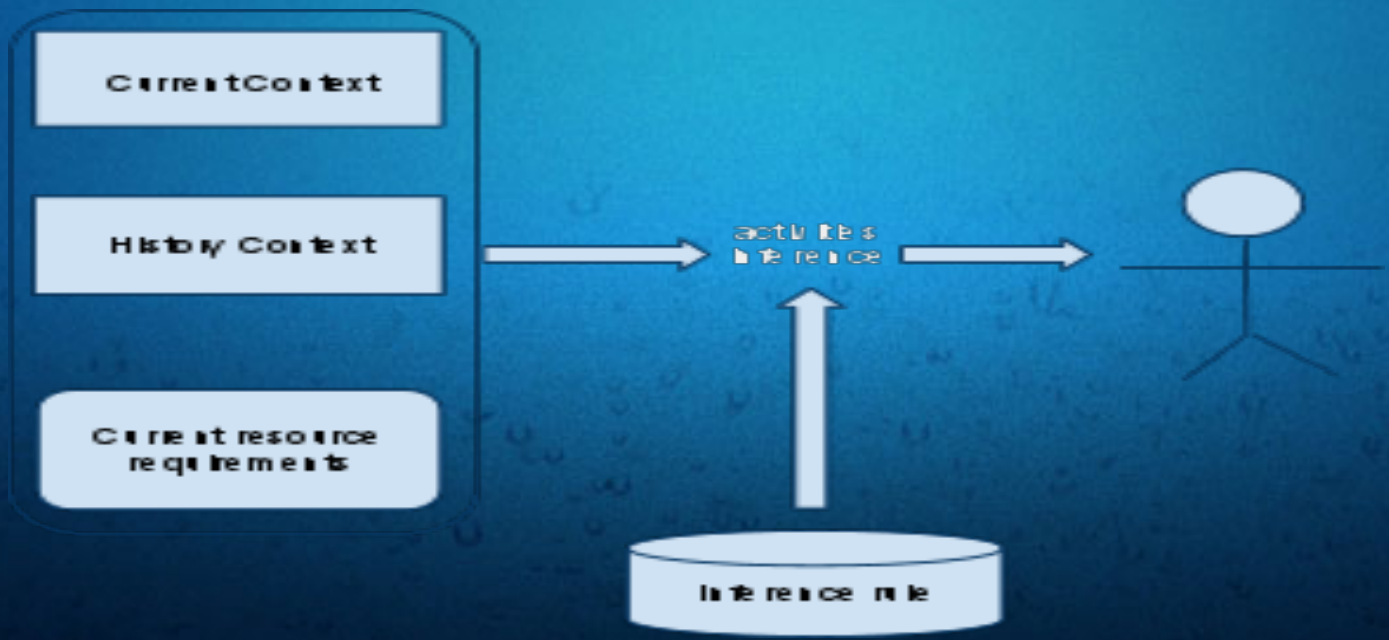
Problem Statement and Adversary Model

- Is software certifications and reference model enough to prevent malicious applications?
  - Certifications: iPhone OS...
    - Apple checks the program.
    - Binary program analysis is difficult and undecidable.
    - Developer can not give source code to Apple.
  - Reference Model: Android, Symbian,...
    - Sand-boxing, run time monitoring, integrity verification, ...
    - Listing all privileges a program has.
    - Sensor sniffing attack.
- Privileges list can not describe software clearly and user can not understand the associated risk.

Research Goal

- Context information retrieval
  - Find what context information is important to describe the behavior of a software and how to get such information.
- Context-aware program behavior inference.
  - Use context information to infer possible risk activities a software might have.
- Help user to understand the risk of a running software.
  - What interface is better to present the risk to user?

Framework


CurrentContext
History Context
Current resource requirements
activities inference
Inference rule

Approach and Methodology

- Capture current context information
  - use more attributes to identify applications
- Learn past context information from running history
  - recording past context information
- Estimate risk using context information
  - Combined with context information and current resource requesting and a set of inference rule, infer possible risk activities
- Design risk communication to convey appropriate quantified risk
  - What interface? What frequency?
- Build prototype
  - Build a prototype system on Android platform
- Test efficiency

Schedule

- Week 5 (2/23/10-1/3/10):
  - List all possible context information we can get.
  - Discussion of inference rule.
  - Distribute the work.
- Week 6 - 7 (2/3/10-3/15/10):
  - Discuss the design of a prototype.
  - Start implementing the prototype.