Component-based Decoupling of Mobile Applications using RemoteBinder

Yutao Liu  Yubin Xia  Haibo Chen
{ytiu.cc, xiyubin, haibochen}@sjtu.edu.cn

Motivation:
Enhance mobile security utilizing mobile cloud computing
- **Key issues:** make cloud and mobile cooperate with each other
  - Approach 1: running **replica** of mobile image in cloud[1].
  - Too strict consistency, consume high network bandwidth
  - Not transparent for application, or require tight coupling

Observation:
Android framework for application development
- Adopt a **component-based** approach
  - Each application consists of multiple loosely coupled components
  - Services, Activity, Broadcast Receiver, Content Provider
  - Each component communicate with others using Android binder mechanism

Mobile malware characteristics
- Malicious code usually runs in background Service
  - Most malware is repackaged, Services loosely coupled with others
  - Service is the common component to run maliciously without users noticing

Solution:
A transparent Service component-based decoupling approach
- **An integrated mobile-cloud protection framework**
  - Define the whole protection process in mobile and cloud
- **Cross-reference** tool for static analysis
- Find the closure of a Service and all of its outward API invoking points
- **Behavior-based suspicious Service detector**
  - Detect Service as suspicious based on its behavior
- **Proxy-based mechanism for bidirectional RPC**
  - Communicate the decoupled Service in cloud with other components in mobile
- **Asymmetric protection policies in mobile and cloud**
  - Lightweight protection in mobile and stronger protection in cloud for suspicious Services

Our solution requires Android framework modification: including AMS, apk installer, permission system, etc.

Working Prototype

Prototype evaluation
- Environment Setup
  - Samsung Galaxy Nexus with Android 4.2
  - Cloud emulator in Debian 6.0
- Prevent 7 types of malicious behavior
  - AMS, apk installer, permission system, etc.
- Performance evaluation
  - Remote method call delay is 4ms
  - Extra network data is less than 100 bytes/RPC

Related work
- **Replica-based mobile cloud computing for security**
- **Application partition approaches for mobile cloud computing**