Quire: Lightweight Provenance for Smart Phone Operating Systems

Michael Dietz, etc.
Presented by Ben Niu
Outline

- Problem(s)
- Solutions
- Application example
- Evaluation
- Critique
The popularity of Android phones are increasing.

Android applications have almost no vetting process before entering the application market.

The security of applications greatly rely on the system-level protection mechanisms.
Problem(s) (cont.)

- Deputy attacks.
  - Trusted applications may be abused by malicious applications via IPCs.
  - This issue is talked during last sos seminar.

- Mutual distrust.
  - Applications do not and should not trust each other.
  - The remote server does not and should not trust the devices.
  - When the communication between applications and servers and devices is involved, the trust issues arises.
  - Example: nested advertisement.
Problem(s) (cont.)

- The problem Quire desires to solve could be presented as “how to ensure all the IPCs between applications, RPCs between remote servers and local devices are legal and trusted.”
- “Legal”: a principal has the privilege calling the functions exported by another principal, even indirectly.
- “Trusted”: the callee principal receives exactly what the caller principal means without any forges.
Solutions (to deputy attack)

- Deputy attack prevention.
  - Quire introduces stack inspection (to be specific a call chain inspection) to ensure all the call nodes on the stack has the privilege of doing “something”.
  - When the flag “enablePrivilege” turned on, the call chain inspection would be shut down.
  - The call chain could be viewed as statements
    - A calls B calls C: B says A says OK.
Solutions (to deputy attack)
Solutions (to mutual trust)

- Verifiable statements
  - Call chain inspection could not solve the problem of mutual trust.
  - A call should be verified or signed.
  - Quire introduces a central authentication manager which do the verification.
    - A signs the call and sends to B, B sends A’s statement to the central authentication manager. After the authentication manager verifies the statement, B will continue running.
Solutions (to mutual trust)

- RPC attestation
  - Quire forbids the applications to communicate to the remote server directly, instead all the messages are sent via a “Network Provider” process using X.509 certificate (used in HTTPS protocols.)
Application example

- PayBuddy
  - A standalone Android application which exposes an activity to other applications on the device to allow those applications to request payments.
  - ExampleApp -> PayBuddy -> PayBuddy.com (remote server)
Application example (cont.)

Diagram:

- **Userspace**
  - MAC Key: $k_A$
  - PurchaseOrder $po$:
    - Cost $c$
    - Payee $p$
  - $MAC_{k_A}(po)$
  - ExampleApp

- **Operating System**
  - “ExampleApp says …”
  - “PayBuddy says …”

- **Auth Manager**
  - $k_B$ → “PayBuddy”

- **PayBuddy.com**

- **PayBuddy**
  - MAC Key: $k_{PB}$
  - RPC_{PayBuddy.com}(…)

- **Net Provider**
  - $k_A$ → “ExampleApp”

- **PayBuddy.com**

- **PayBuddy.com**
Evaluation

- Quire prototype was based on around 2,000 Java and C code
- Statement Verification and Creation
Critique

- This paper is a practical paper which solves a big issue in current Android systems, and we hope it will help Android designers to review the security problems.
- This paper is based on existing applied cryptography techniques, so the technology is sound.
Critique (cont.)

- uPro provides IPC (Inter-Protection Communication) via CDCs and current CDCs are short of such inspection.
- uPro does not provide CDC verification and this paper gives us a good demonstration.
- The performance of statement creation and verification on PCs or servers will beat the performance on the phone.