A Policy Enforcement Framework for Android

Kaustubh Keskar

M.Tech. CSE,

Indian Institute of Technology Bombay

Outline

- Introduction
- Literature Survey
- Proposal
- Conclusion

Basics – Android Architecture

Components of an Android app [3]:

- **Activity**: User interface (UI) of an application
- **Service**: Background process invisible to the user
- **Content Provider**: An interface to the database
- **Broadcast Receiver**: An asynchronous event mailbox for Intent messages

Basics – Android Security

- Application Sandboxing
  - Unique UID for every app
  - Every app is run into separate VM
- Application Signing
  - Self signed certificate is sufficient

Basics – Android Security

- Permission Model
  - Android protects device and OS features (services), using permission labels
  - All-or-nothing approach at the time of installation
  - Cannot revoke granted permissions [unless uninstalled]
  - App gets unrestricted access to the resource

A Policy Enforcement Framework

- Deals with users' security and privacy concerns, by allowing them to define policy rules

- Goals
  - To restrict the usage of resources
  - To prevent privilege escalation attack
  - In general, to provide fine-grained access control

A Policy Enforcement Framework

- Users of the system: End-user, or trusted third party, or both

- Context-aware policies
  - Based on environmental or system attributes like time, location, CPU speed, battery, etc.

Motivation

- A policy enforcement framework for Aakash tablet
  - No apps during quiz/exam time
  - Limited set of apps during school-time
  - Different set of apps for different subjects/courses
  - Parental control (at home)

Motivation

- Context Attribute (for context-aware policies)
  - Battery virtualization: Battery consumption information per process

- Remote Access Mechanisms (to update or enforce policies)
  - Existing: SMS, Bluetooth, WiFi
  - Not suitable, if the number of users, is large

Outline

- Introduction
- Literature Survey
- Proposal
- Conclusion

Saint [2]

- Framework to protect apps from other apps
- **Install-time enforcement**: Controls permission assignment
- **Runtime enforcement**: Governs communication access between components

Saint [2]

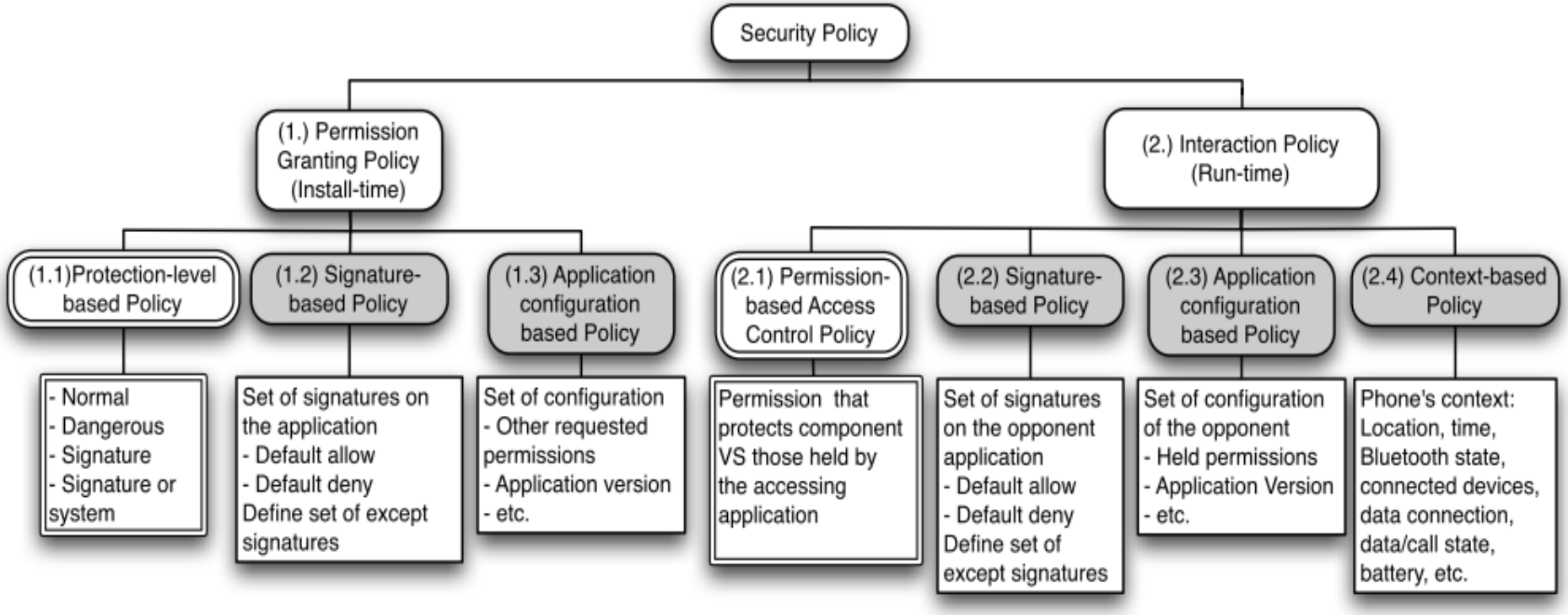

Figure: Saint [2]

Outline

- Introduction
- Literature Survey
- Proposal
- Conclusion

Proposal

- A policy enforcement framework for Aakash tablet
  - No apps during quiz/exam time
  - Limited set of apps during school-time
  - Different set of apps for different subjects/courses
  - Parental control (at home)

Proposal

Proposed Architecture:

- Users: Schools/Colleges, Teachers, Parents (i.e. trusted third parties and end-users)
  - Requires priority handling: Teachers > Schools
  - We don't want to allow students to enforce policies, but parents should be allowed - provide authentication mechanism
- Context-aware policies: Time, Location, Battery

Proposal

- Context Attribute (for context-aware policies)
  - Battery virtualization: Battery consumption information per process

Proposed Architecture:

- Android has private API for app-level battery consumption information in PowerUsageSummary.java [5]

Proposal

- Remote Access Mechanisms (to update or enforce policies)
  - Existing: SMS, Bluetooth, WiFi
  - Not suitable, if the number of users is large

Proposal

Proposed Architecture:

- SMS: paid service

  Bluetooth: limited range, cannot handle more users

  WiFi: requires polling
- Google Cloud Messaging (GCM) [4]
- Free service – Push mechanism instead of polling
- Stores messages on GCM servers if the device is online

Outline

- Introduction
- Literature Survey
- Proposal
- Conclusion

Conclusion

- Compared existing policy enforcement frameworks

- Proposed solution for use-cases

References

- Mauro Conti, Vu Thien Nga Nguyen, and Bruno Crispo.
  CRePE: context-related policy enforcement for android.
  In Proceedings of the 13th international conference on Information security, ISC'10, pages 331–345, Berlin, Heidelberg, 2011. Springer-Verlag.

- Machigar Ongtang, Stephen McLaughlin, William Enck, and Patrick McDaniel.
  Semantically Rich Application-Centric Security in Android.
  In Proceedings of the 2009 Annual Computer Security Applications Conference, ACSAC '09, pages 340–349, Washington, DC, USA, 2009. IEEE Computer Society.

References

- Android Developers Team.
  Application Fundamentals.
  http://developer.android.com/guide/components/fundamentals.html, 2013. [Online; accessed on 12-Feb-2013].

- Android Developers Team.
  GCM Architectural Overview.
  http://developer.android.com/guide/google/gcm/gcm.html, 2012. [Online; accessed on 5-Oct-2012].

- Android Developers Team.
  PowerUsageSummary.java.
  http://androidxref.com/4.0.4/xref/packages/apps/Settings/src/com/android/settings/fuelgauge/PowerUsageSummary.java, 2012. [Online; accessed on 5-Oct-2012].