

Software Security for Mobile Devices

Eric Bodden

HEINZ NIXDORF INSTITUT
UNIVERSITÄT PADERBORN

Fraunhofer
IEM

contributors:

Prof. Dr. Heiko Mantel, Prof. Dr. Markus Müller-Olm, Prof. Dr. Alexander Pretschner, Prof. Dr. Wolfgang Reif, Prof. Dr. Gregor Snelting, Dr. Alexandre Bartel, Kuzman Katkalov, Dr. Enrico Lovat, Martin Mohr, Benedikt Nordhoff, Matthias Perner, David Schneider, Dr. Artem Starostin, Dr. Kurt Stenzel, …

Android App Security

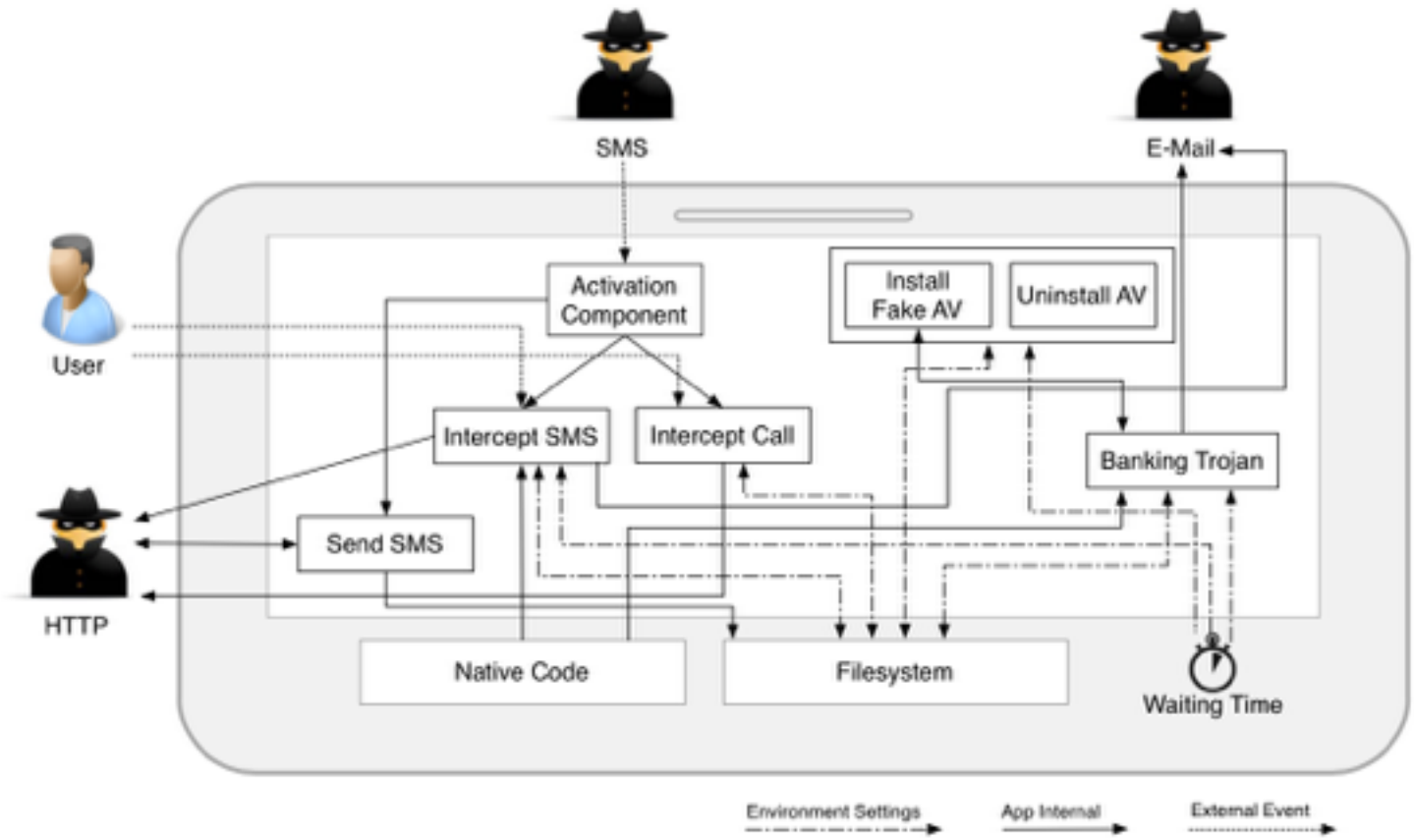
Android is the most widely used mobile operating system

- 85% market share at the end of 2015
  http://www.gartner.com/newsroom/id/3169417

Security problems of Android apps are frequent

- **"Another Popular Android Application, Another Leak"**
  https://www.fireeye.com/blog/threat-research/2015/08/another_popular_andr.html
- **"Mobile Threat Monday: Evil Android App Steals Text Messages"**
  http://www.pcmag.com/article2/0,2817,2478552,00.asp
- **"Mobile Threat Monday: Android Malware Looks Safe, Steals Your Photos and Messages"**
  http://securitywatch.pcmag.com/mobile-security/331088-mobile-threat-monday-android-malware-looks-safe-steals-your-photos-and-messages
- **"Top Mobile Apps Overwhelmingly Leak Private Data: Study"**
  http://www.eweek.com/mobile/top-mobile-apps-overwhelmingly-leak-private-data-study

Android Malware



BadAccent malware, 2014/2015

Android App Security

REUTERS EDITION: U.S. SIGN IN | REGISTER Search Reuters

HOME BUSINESS MARKETS WORLD POLITICS TECH OPINION BREAKINGVIEWS MONEY LIFE PICTURES VIDEO

WATCH LIVE : Security operation underway in Brussels for Paris attacks suspect »

ADVERTISEMENT

Markets | Wed Jun 17, 2015 3:35am EDT

Related: STOCKS, MARKETS, CYCLICAL CONSUMER GOODS, TECHNOLOGY

'Billions' of records at risk from mobile app data flaw

* Flaw leaves data stored by apps vulnerable -researchers
* Almost every category of app considered vulnerable
* Passwords, addresses, photos, medical data all at risk
* Records affected "will likely be in the billions"

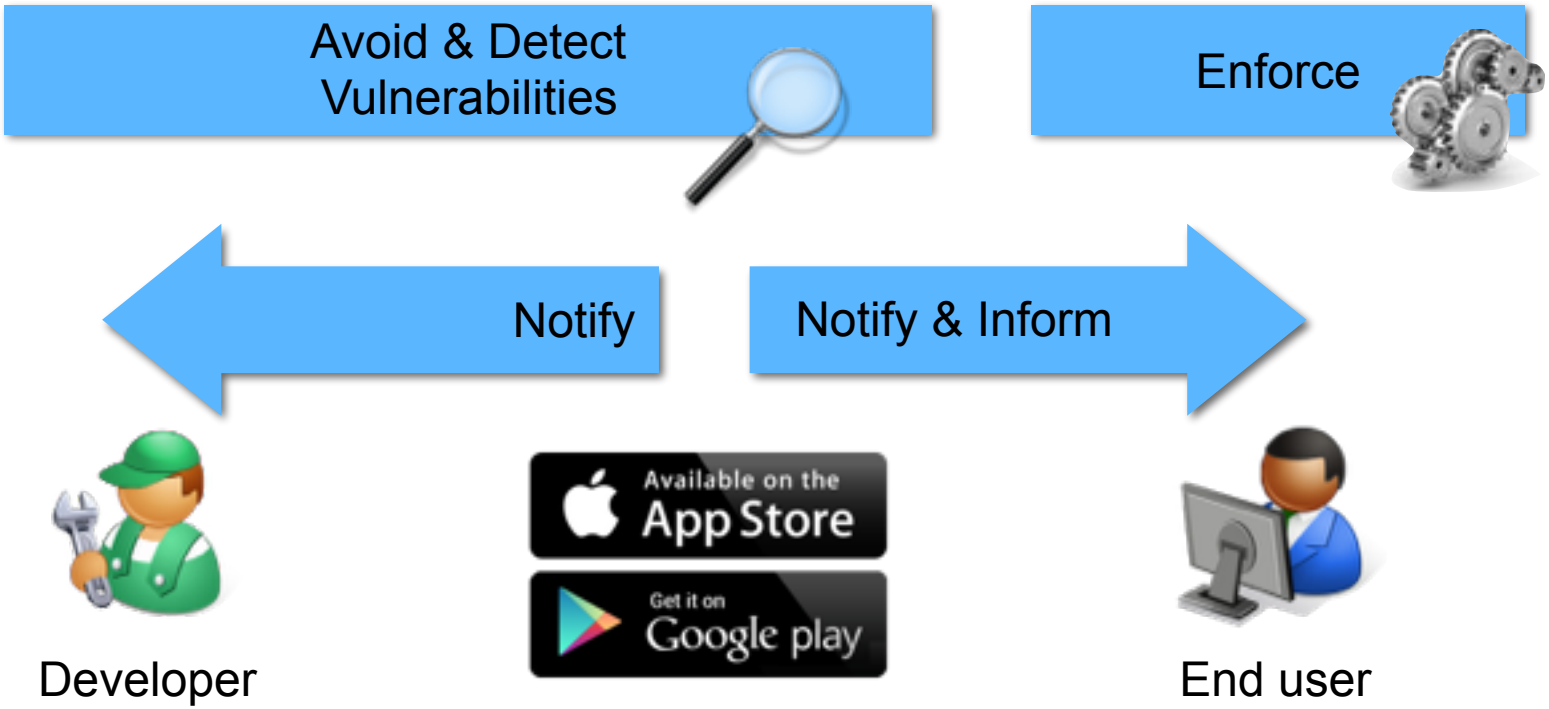
By Jeremy Wagstaff

EDITOR'S CHOICE

Vietnam's Gaga is running for office

Must aid developers in protecting their applications

Ways to enhance security



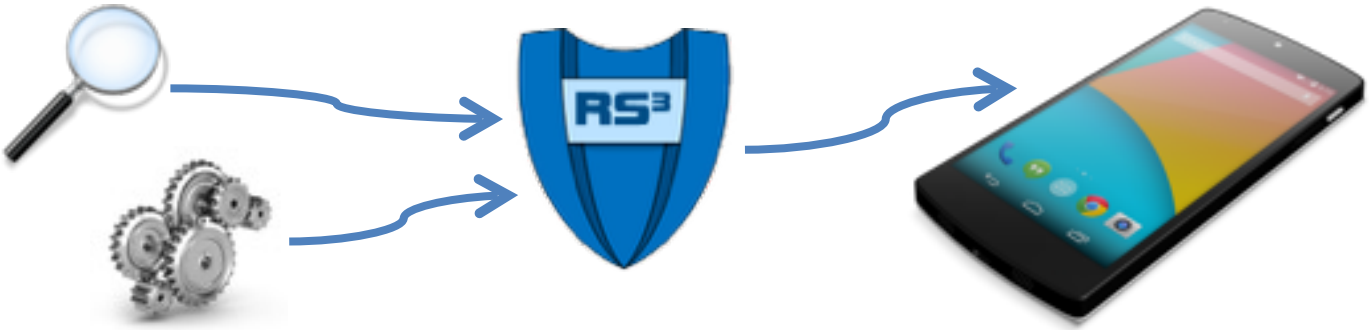
Our Goals and Approach

Goal: enable users to specify their personal security requirements

- confidentiality of sensitive data (i.e., information-flow control)
- limited uses of resources (i.e., usage control)
- protection shall complement Android's built-in protection mechanisms

Approach:

- develop interfaces for users to specify their security requirements
- develop solutions for establishing these security requirements
- display the security guarantees or potential problems to users



Our Prototype: The RS³ Certifying App Store

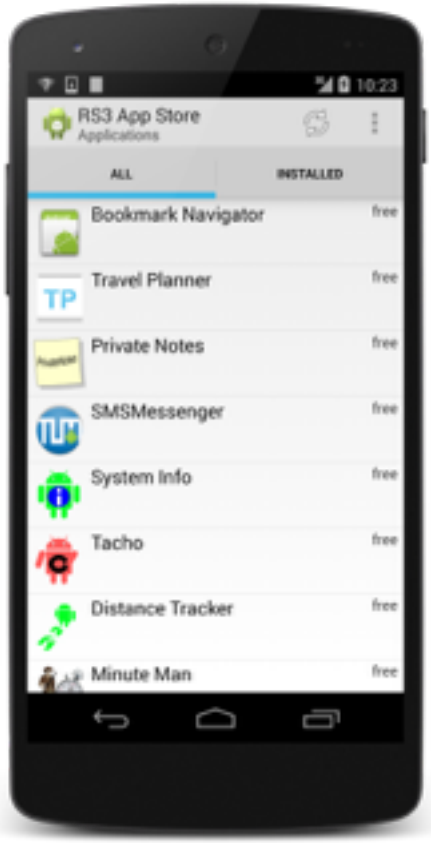
The RS³ Certifying App Store

Provides typical functionality of an app store

- browsing apps and displaying details about them, e.g., a description
- intention: familiar to users, hence easy to use

Important: many RS³ technologies integrated

- two static information-flow analyses
- run-time monitoring and enforcement
- security certificates from security-by-design



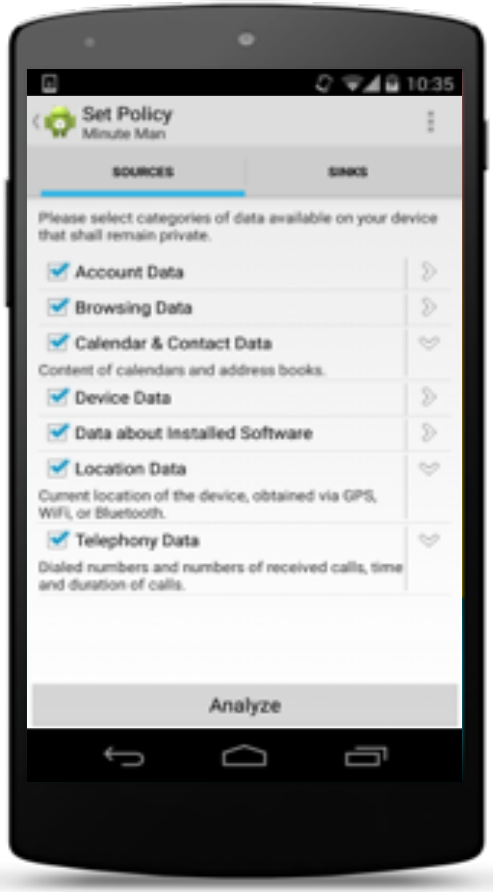
Support of User-Defined Policies

Users may have different security concerns

- enable users to specify their individual security requirements

Accessible to non-expert users

- user interface abstracts from technical details
- users select categories of data they want to keep secret
- a short description is given for each category to help the user decide
- if users are unsure, they can just check-mark category



Further exploration of alternative interfaces is planned.

Integration of Multiple Analyses

Static security analysis

- … is performed at design- or at compile-time
- … provides guarantees about all possible executions
- if an app passes the analysis for some property, then the property is satisfied by every execution of the app
- static analysis can be used for information-flow control

Dynamic security enforcement

- … is performed on-the-fly during execution of an app
- … provides guarantees for an actual execution of an app
- if an execution of an app causes an action that would violate the property of interest then this action is prevented at run-time
- dynamic enforcement can be used for usage control

We use two approaches for two orthogonal security concerns.

The Type-based Security Analysis (from Cassandra)

Analysis specified by a security type system

- security types are used to specify whether data is confidential or not
- security types are used to specify whether an information sink is trusted
- security type systems are sets of rules

$$\frac{M[p] = \texttt{binop}\ x_a, x_b, x_c, bop \quad \text{rda}' = \text{rda}[x_a \mapsto \text{rda}(x_b) \sqcup \text{rda}(x_c) \sqcup se(p)]}{M, region_{p,M}, \text{fda}, \text{mda}, ret, se \vdash p : \text{rda} \to \text{rda}'}$$

$$\frac{M[p] = \texttt{new-instance}\ x_a, c \quad \text{rda}' = \text{rda}[x_a \mapsto se(p)]}{M, region_{p,M}, \text{fda}, \text{mda}, ret, se \vdash p : \text{rda} \to \text{rda}'}$$

$$\frac{M[p] = \texttt{iget}\ x_a, x_b, f \quad \text{rda}' = \text{rda}[x_a \mapsto \text{rda}(x_b) \sqcup \text{fda}(f) \sqcup se(p)]}{M, region_{p,M}, \text{fda}, \text{mda}, ret, se \vdash p : \text{rda} \to \text{rda}'}$$

Type system proven to be sound

- each application the analysis validates has secure information flow

Formal foundation of the analysis

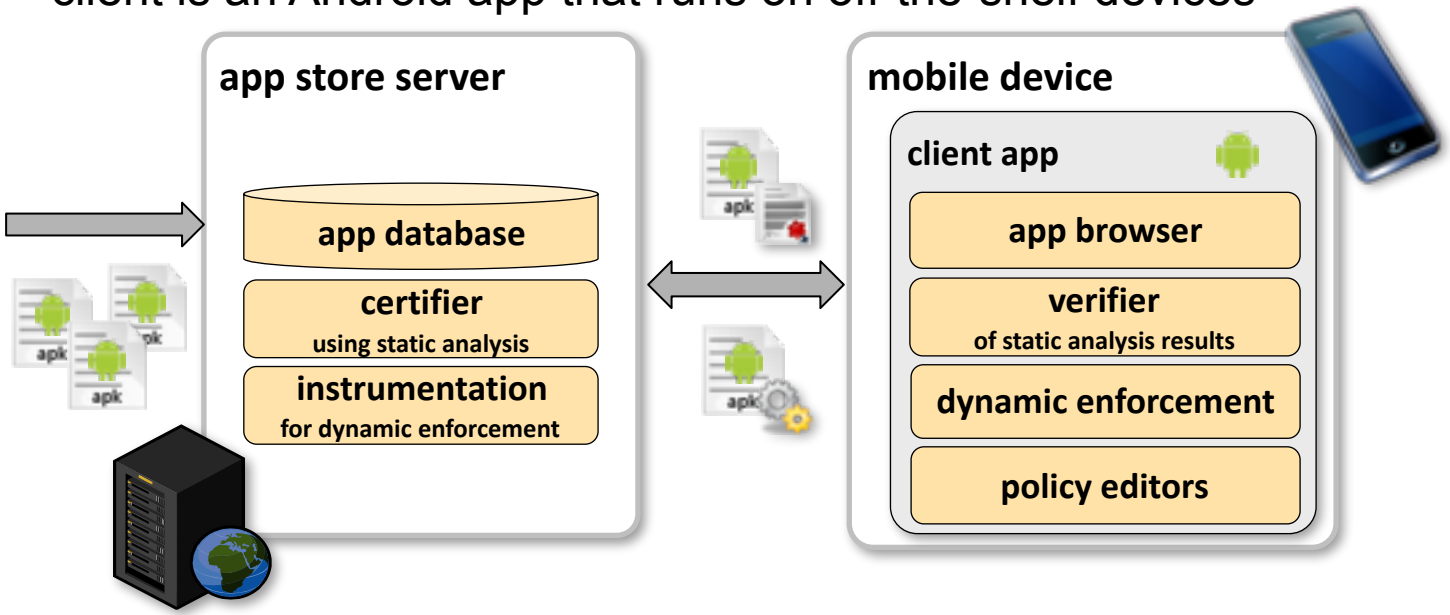
- formal model of Dalvik bytecode operational semantics
- formal definition of security based on the semantics

First such analysis for Android with soundness proof.

Architecture of the App Store

Client-server architecture

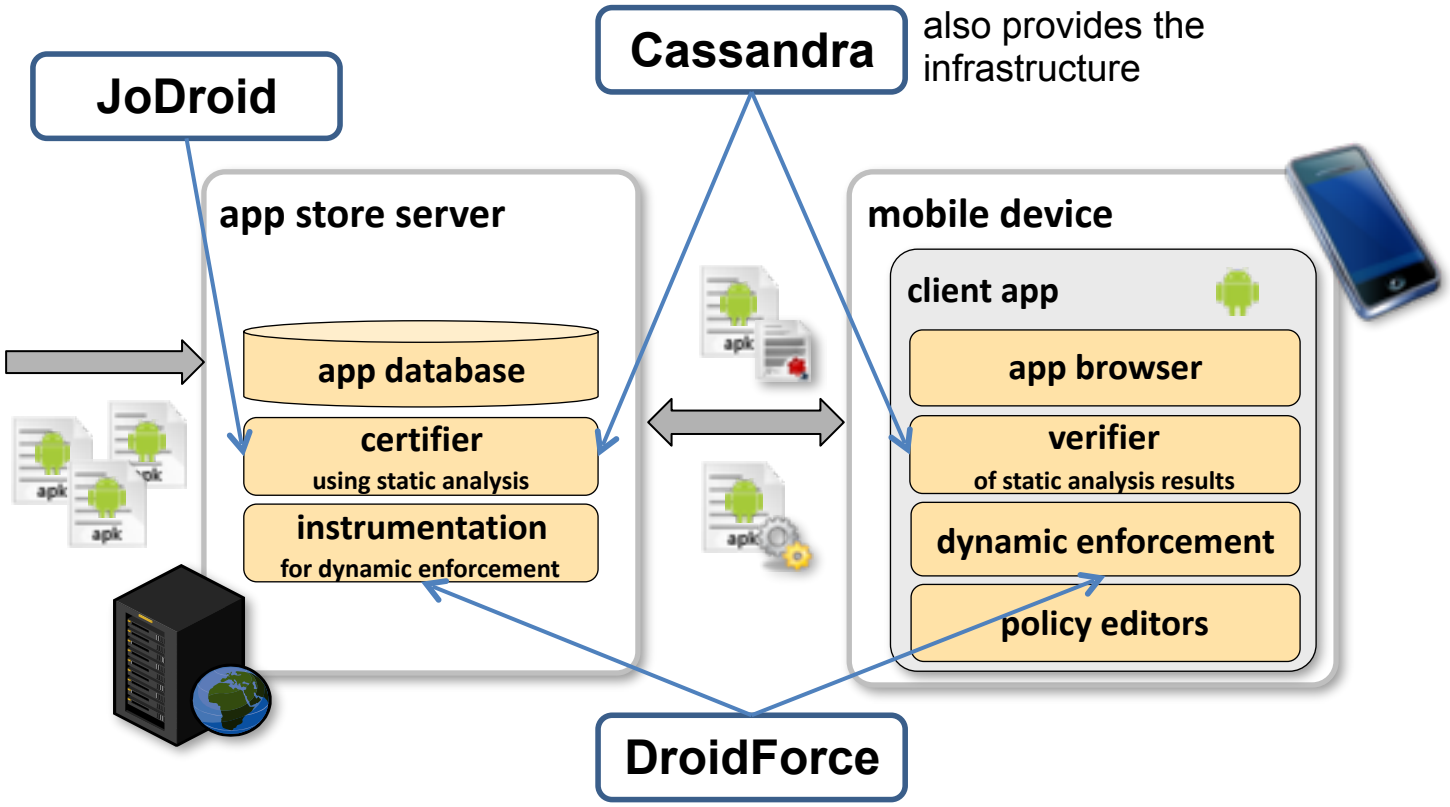
- server stores and distributes apps
- client is an Android app that runs on off-the-shelf devices



Security technology is integrated in both client and server

- resource-intensive operations are performed on the server
- the user can specify his security requirements in the client app

Integration of Analysis Tools



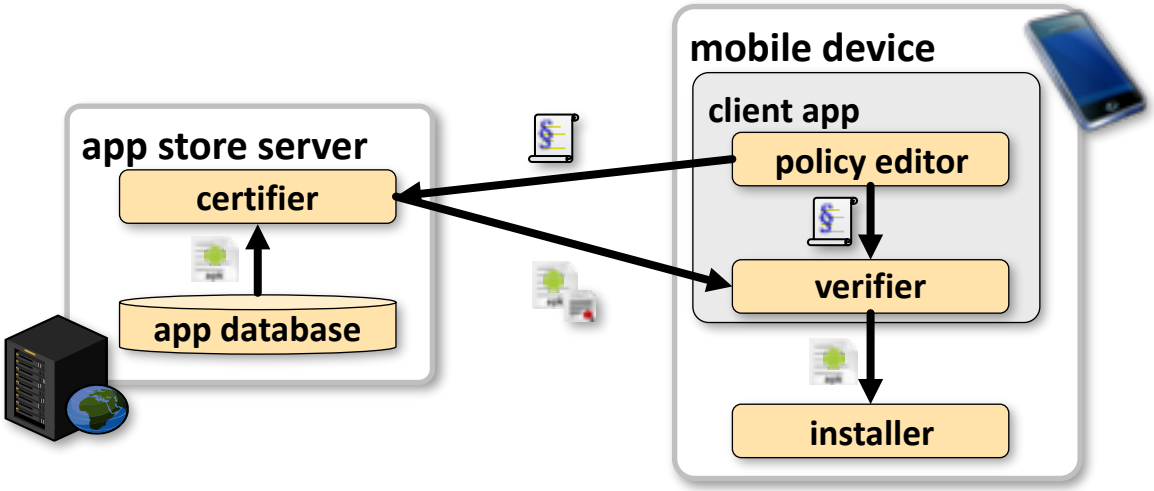
Proof-Carrying Code
(from Cassandra)

General idea (Necula, 1997):

- provide program code together with proof of a property
- essential property: proof can easily be verified

In our case:

- server receives user policy and analyzes app (type inference)
- client obtains app and analysis result and verifies result (type check)
  - type check is fast



Presentation of Results
(from Cassandra)

Results of static analyses are displayed to the user visually

- shows all flows of potentially sensitive information to an output channel

Users can quickly make informed decisions whether to install an app.

Overview of Integrated Tools

Cassandra: static information-flow analysis

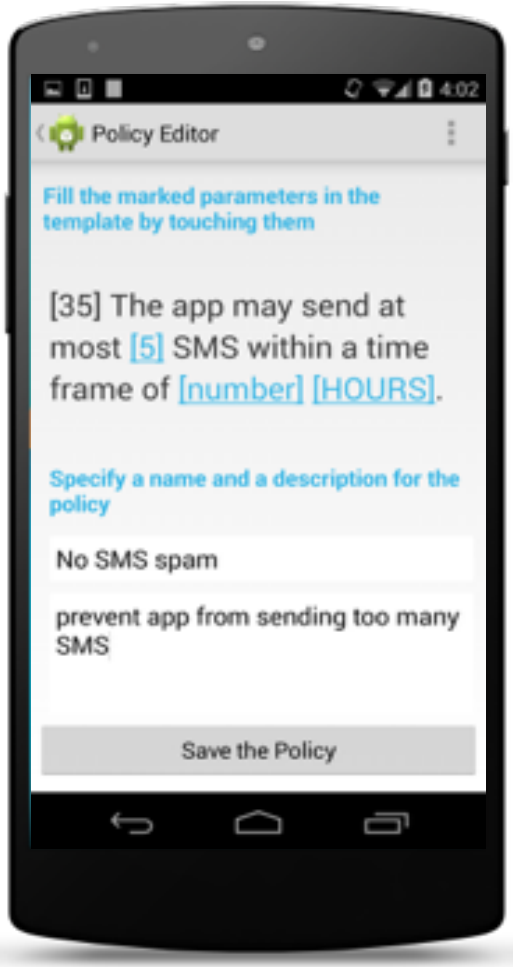
- based on a security type system
- policy editor for selecting categories of data to be kept secret

JoDroid: static information-flow analysis

- based on program-dependence graphs
- uses the same policy editor as Cassandra

DroidForce: dynamic usage control

- prohibits unwanted functionality of apps
  - e.g., “not more than 5 SMS are sent per hour”
- editor allows users to instantiate predefined policy templates
- uses static pre-analysis of **FlowDroid**





Investigated Apps so far

Initially: Self-developed case studies

- simplified apps with the core functionality of existing apps
- 8 case studies in total

App	Functionality	Security Problem
Bookmark Navigator	Display browser bookmarks.	Browsing history is leaked.
Minute Man	Break off calls to limit costs.	Called telephone numbers are leaked.
Distance Tracker	Record jogging distance.	Location of the device is leaked.

Recently: Third-party apps

- so far: analyzed 48 apps from the F-Droid app store
- static analysis: found problematic flow of information in some apps
- dynamic enforcement: successful instrumentation of all apps
- model-driven development: found some apps promising complex informal security guarantees

FlowDroid soon in widespread productive use

FlowDroid
currently being adopted
for use by one of the world's
largest app-store providers

Will go live in fall

More on FlowDroid at:
https://github.com/secure-software-engineering/soot-infoflow

Conclusion

Goals:

- ensure confidentiality of sensitive data
- limited uses of resources
- enable users to enforce their individual security requirements

Approach:

- develop an app store based on RS³ tools
- provide interfaces for specifying requirements

Prototype: The RS³ Certifying App Store

Outlook:

- provide security guarantees for and detect security problems in third-party apps

Selected Relevant Publications

- Steffen Lortz, Heiko Mantel, Artem Starostin, Timo Bähr, David Schneider, and Alexandra Weber:
  Cassandra: Towards a Certifying App Store for Android.
  In Proceedings of the 4th ACM CCS Workshop on Security and Privacy in Smartphones and Mobile Devices, pages 93-104, 2014.

- Martin Mohr, Jürgen Graf, and Martin Hecker:
  JoDroid: Adding Android Support to a Static Information Flow Control Tool.
  In Proceedings of the 8th Working Conference on Programming Languages, pages 140-145, 2015.

- Siegfried Rasthofer, Steven Arzt, Enrico Lovat, and Eric Bodden:
  DroidForce: Enforcing Complex, Data-Centric, System-Wide Policies in Android.
  In Proceedings of the 9th International Conference of Availability, Reliability and Security, pages 40-49, 2014.

- Steven Arzt, Siegfried Rasthofer, Christian Fritz, Eric Bodden, Alexandre Bartel, Jacques Klein, Yves Le Traon, Damien Octeau and Patrick McDaniel:
  FlowDroid: Precise Context, Flow, Field, Object-sensitive and Lifecycle-aware Taint Analysis for Android Apps, In Proceedings of the 35th ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI '14), pages 259–269, ACM, 2014.