RV-Droid: Runtime Verification and Enforcement for Android Applications

Yliès Falcone, Sebastian Currea, Mohamad Jaber

Laboratoire d’Informatique de Grenoble - VASCO Team
University of Grenoble, Université Joseph Fourier

3rd International Conference on Runtime Verification
Istanbul, Turkey
26 September, 2012
Andy Rubin: “900 K activations per day”
1. Android’s market shares:

![Pie chart showing mobile phone market share, 1Q 2012.](image)

2. Moore’s and Koomey’s laws for mobile devices

   comScore Reports. U.S. mobile subscriber market share, April 2012
Reliability of Android applications... can be improved

The Rise of Android’s malwares...

- Android malware increased by 472%
- no upfront validation process
- Android will remain the 1st target of mobile malware writers

Developing Android applications is difficult:

- complex communication scheme between activities
- complex life-cycle
- partial compatibility with JRE and some existing libraries

⇒ several bugged and unprotected applications
⇒ resource consumption and user-experience can be negative
Proposed solution: monitoring of Android applications

Using dynamic validation techniques to enhance confidence in the behavior of (untrustworthy) applications

Two processes over the runtime behavior

- checking some properties
- enforcing some properties

(up to the observability & controllability provided by the instrumentation technique)

Modify applications to incorporate monitors

- observe calls to the unified Android API and analyse parameters
- decide whether the properties are satisfied/violated (runtime oracle)
- correct the behavior by disabling some actions or modifying their return value
Dynamic Validation Techniques needs Instrumentation

Most successful one is \textit{Aspect-Oriented Programming}
- identify special locations in the execution of the system
- execute code when those locations are reached

Unfortunately current AOP is not (fully) compatible with Android
- constraints seriously hinders mobility
- possible only for self-developed applications (plugged to the computer or within the emulator)
- incompatibility between .\texttt{apk} files and aspect compilers
Outline

1. Overview of RV-Droid
2. A Tour of Examples
3. Related Work and Discussion
4. Summary and Perspectives
Overview of RV-Droid

RV-Droid: platform

User selects application & property to upload & download. RV-Droid processes these requests and delivers the result back to the user.

- **RV-Droid**: Platform for managing application and property repositories.
- **Java-MOP** and **RuleR**: Tools for monitoring and integrating software properties.
- **Property Repository**: Stores property information.
- **Application Repository**: Stores application information.
- **RV Cloud**: Centralizes monitoring and integration processes.
- **Monitor Request**: Initiates the monitoring process for selected application & property.
- **Result**: Delivers the monitored application and property.
Overview of RV-Droid

RV-Droid: screenshot
Overview of RV-Droid

RV-Droid: screenshot (ctd)
RV-Droid: features

Features:
- stand-alone application
- does not require any modification to the Android OS
- applications are retrieved *off-the-shelf*

For flexibility, all the involved processes can be done:
- embedded on the device
- in the cloud (web service with a configurable IP)
Outline

1. Overview of RV-Droid

2. A Tour of Examples

3. Related Work and Discussion

4. Summary and Perspectives
Outline

1. Overview of RV-Droid
2. A Tour of Examples
3. Related Work and Discussion
4. Summary and Perspectives

Android development good practices
Android Development Good Practices

P1 Before transmitting any data, it must be ensured that the device is connected to internet. And, it should be checked again each time the device is moved.

P2 All methods involved in the activity lifecycle should be overridden.

P3 The device rotation facility should not be disabled.

P4 Only one dialogue window should be popped-up.

P5 In the restricted-memory mode, an application should start at most one service and end it, and not let the Dalvik virtual machine kill it.
**P1** Before transmitting any data, it must be ensured that the device is connected to internet. (And, it should be checked again each time the device is moved.)
Lifecyle Methods should be Overridden

P2 *All methods involved in the activity lifecycle should be overridden.*

1. Observes the execution of these methods in the (implemented) application

2. Tracking the (simplified) application lifecycle:
Outline

1. Overview of RV-Droid
2. A Tour of Examples
3. Related Work and Discussion
4. Summary and Perspectives

Security Properties
We disabled the security findings found in *A study of android application security* published in SEC11 (Enck et al.)

Applications should not:
- access to the phone data, i.e., IMEI (device identifier), IMSI (subscriber identifier), and ICCID (SIM serial number)
- send SMS to premium-rate numbers
- call premium-rate numbers
- record audio or video without the user knowing it
- request the list of installed applications
- log events
- create unprotected Intents
- register unprotected broadcast receiver
Outline

1. Overview of RV-Droid
2. A Tour of Examples
3. Related Work and Discussion
4. Summary and Perspectives

Additional Examples
Blocking Advertisements

- Game displaying advertisements through banners
- Banner content is retrieved from the Web
Blocking Advertisements

- Ad display go through the methods in the package `com.google.ads`

```java
aspect BannerAspect {
  // Pointcut to block Google banners
  Object around() : execution(* com.google.ads..*(..))
    && !within(BannerAspect) {
      return null;
    }
}
```
Modifying the Device Location

- Applications display information according to the current location of the device (GPS coordinates)
Modifying the Device Location

```java
aspect ChangeLocationAspect{
    // Pointcut to Android location method.
    pointcut location(String provider) : call(* android.location.
    LocationManager.getLastKnownLocation(..)) && args(
    provider) && !within(ChangeLocationAspect);
    // Advice to change the device location
    Location around(String provider) : location(provider) {
      ...
      Location location = new Location(provider);
      // New latitude and longitude values in Greenland
      location.setLatitude(79.13826);
      location.setLongitude(-46.40625);
      return location;
      ...
    }
}
```

Applications:
- Testing
- Privacy
Collecting Statistics on Applications

Data sent by applications:
- how much
- where?

<table>
<thead>
<tr>
<th></th>
<th>G1</th>
<th>G2</th>
<th>G3</th>
<th>G4</th>
<th>G5</th>
<th>G6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sent</td>
<td>3559</td>
<td>15453</td>
<td>16223</td>
<td>2426</td>
<td>16197</td>
<td>14557</td>
</tr>
<tr>
<td>Received</td>
<td>1205</td>
<td>2364</td>
<td>2466</td>
<td>1123</td>
<td>3194</td>
<td>2283</td>
</tr>
</tbody>
</table>

```java
aspect UrlBytesAspect {
  // Pointcut to the method that loads an URL
  pointcut pageName(String page) : (execution(* android.webkit.
    WebView.loadUrl(..))
    || execution(* android.webkit.WebView.
      loadDataWithBaseURL(..)) )
  && args(page,..) && !within(UrlBytesAspect);
  ...
  startRX = android.net.TrafficStats.getTotalRxBytes();
  // variable to count received bytes
  startTX = android.net.TrafficStats.getTotalTxBytes();
  // variable to count transferred (sent) bytes
  ...
  // Advice that stores the Bytes consumed and the Url in a file
  after(String page): pageName(page) {
    ...
  }
}
```
Outline

1. Overview of RV-Droid
2. A Tour of Examples
3. Related Work and Discussion
4. Summary and Perspectives
Related Work and Discussion

Static Validation Techniques

Mostly relying on analyzing the permissions

Stonaway
- Checks the principle of least privileges
- Compares:
  - the *requested* permission (requested at installation time)
  - vs the *needed* permissions (used at runtime)

ComDroid
- Analysis of inter-application communication
- Prevent disclosure of information through intents (i.e., messages)
Dynamic Validation Techniques

TaintDroid (2010)
- Information-flow monitoring
- Tainting information in logs

Bauer et al.: closest approach
- Monitoring LTL formula over permissions by progression
- 2 variants:
  - modify two files on the device (observation of high-level events)
  - propose to add a kernel module (observation of high and low level events)
Comparison and Discussion

RV-Droid in comparison

- General and generic method (limited by the API)
- Devices are taken off-the-shelf (no modification needed)
- Applications are taken off-the-shelf
- Based on state-of-the-art RV tools:
  - expressiveness
  - efficiency
- Allows Runtime Enforcement
- Opens several academic and industrial perspectives

Current limitations

- RV of heavy applications has prohibitive overhead
- Observation using aspects has some limitations
Summary and Perspectives

Summary

- Modification of Android applications using aspect-oriented technology
- Applications are taken *off-the-shelf* (simple yet powerful)
- Devices are taken *off-the-shelf* (no void guarantee)
- Generic aspects: work with any Android device/application

Perspectives

- Dynamic validation techniques for reliability and security
- Trust and reliability (manufacturers, service providers, Google)
- Better integration with existing RV tools