RiskMon: Continuous and Automated Risk Assessment of Mobile Applications

Yiming Jing, Gail-Joon Ahn, Ziming Zhao and Hongxin Hu
Current Practices for Users to Understand Risks of Applications

- **Install-time warnings**
  - Services that cost you money: directly call phone numbers
  - Your location: coarse (network-based) location, fine (GPS) location
  - Network communication: full Internet access
  - Your accounts: Google Maps, manage the accounts list, use the authentication credentials of an account
  - Storage: modify/delete USB storage contents

- **Runtime consent dialogs**
  - "OpenTable" Would Like to Use Your Current Location
    - Don’t Allow
    - OK
Loopholes in Current Practices

Google+

Chase

Gonein60
Root Causes of Loopholes

• App’s runtime behaviors are not assessed
  – Requesting resources does not incur risks.
  – How resources are used is unknown.

• End-users cannot apprehend current practices
  – Users lack the capabilities to digest technical information and estimate risks.
Design Goals

• Continuous monitoring
  – Collect intelligences about runtime behaviors of mobile apps during their lifetime.

• Automated risk assessment
  – Generate a risk assessment baseline to reduce the required sophistication to perform effective risk assessment.

• Intuitive risk presentation
  – Assist general users to understand the risks.
RiskMon Architecture

Android Applications

GooglePlay  Device  User

Meta Information  API Traces  Security Requirements

Application Intelligence Aggregator  Baseline Learner  Risk Meter

RiskMon
Application Intelligence Aggregator
for Continuous Monitoring

Android Applications

GooglePlay | Device | User

Meta Information | API Traces | Security Requirements

Application Intelligence Aggregator

RiskMon

Baseline Learner | Risk Meter
Baseline Learner for Automated Risk Assessment
Risk Meter
for Intuitive Risk Presentation

Android Applications

GooglePlay
Device
User

Meta Information
API Traces
Security Requirements

Application Intelligence Aggregator
Baseline Learner

RiskMon

Risk Meter

Application Intelligence Aggregator

Google Play
Device
User
Collecting API Traces

• Tap into API call/callbacks that are protected by permission groups

- 24 permission groups, 145 permissions
- 1,003 types of IPC transactions
Extracting Features from API Traces

- Type of IPC transaction
- Call/callback
- Length of received message
- Length of sent message
Application Intelligence Aggregator

Collecting Meta Information

- Parse webpages of app markets
Application Intelligence Aggregator

Extracting Features from Meta Information

- Category
- Rating score
- # of reviews
- # of installs
Application Intelligence Aggregator

Feature Vector

Type of IPC transaction

Call/Callback

Recv Length

Send Length

<LocationManager-1, 0, 0.01, 0.08, -0.1, -0.2, 0.88, 0.55, 0.8>

Category X

Category Y

Rating Score

# of Reviews

# of Installs
Baseline Learner

Risk Assessment Baseline: General Idea

• A user-chosen trusted app implicitly indicates the user’s expectation of similar apps’ runtime behaviors

• Learn a baseline from a set of user-chosen trusted apps
Users have disparate expectations

**User-chosen relevancy levels** capture a user’s expectation for each permission group
Baseline Learner

Labeling Feature Vectors

• Rank feature vectors with relevancy levels

\[
\begin{align*}
\vec{x}_1 &= \langle \text{PhoneSubinfo-5, ..., Relevancy Level=1} \rangle \\
\vec{x}_2 &= \langle \text{RecognitionService-1, ..., Relevancy Level=3} \rangle \\
\vec{x}_3 &= \langle \text{LocationManager-14, ..., Relevancy Level=5} \rangle
\end{align*}
\]

Skype
Generating an Assessment Baseline

• Learn a baseline from the ranked vectors
  – Ranking SVM learns a weight vector \( \vec{w}^* \).
  – Weight vector \( \vec{w}^* \) determines a hyperplane.
  – For any vector \( \vec{x} \), its distance to the hyperplane indicates the risk.

– The hyperplane is the risk assessment baseline.
Risk Meter

Presenting Risks Intuitively

App risk rankings

Risk composition
Evaluation: Experimental Setup

- **Dataset**
  - 33 participants
  - 14 applications
  - 33,368,458 Binder IPC transactions

User-chosen Trusted Apps

Target Apps
Evaluation: App Rankings

- Tool for acquiring user-chosen relevancy levels

![Amazon Mobile](image)

Please rate the following permission groups by their relevancy to this app.

<table>
<thead>
<tr>
<th>Permission Group</th>
<th>Relevant</th>
<th>Probably Relevant</th>
<th>Neutral</th>
<th>Probably Irrelevant</th>
<th>Irrelevant</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCOUNTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AFFECTS_BATTERY</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CAMERA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LOCATION</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NETWORK</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>PHONE_CALLS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1 2 3 4 5

Low risk

High risk
Evaluation: App Risk Rankings

- User-chosen relevancy levels of 33 participants
Evaluation: App Risk Rankings

- App rankings by average cumulative risks measured from 33 baselines:
Evaluation: Risk Composition

• Risk composition of SogouInput and PPS.TV

- ConnectivityManager-getActiveNetworkInfo()
- ConnectivityManager-getAllNetworkInfo()

- PhoneSubinfo-getDeviceID()
- PhoneSubinfo-getSubscriberID()
- TelephonyRegistry-listen()
Evaluation: System Usability

• Metrics explained
  – Likeability: basic opinion towards the idea
  – Simplicity: burden due to intervention
  – Perception: perceived increased risk awareness

<table>
<thead>
<tr>
<th>Metric</th>
<th>Average</th>
<th>Lower bound on 95% confidence interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>Likeability</td>
<td>0.811</td>
<td>0.797</td>
</tr>
<tr>
<td>Simplicity</td>
<td>0.674</td>
<td>0.645</td>
</tr>
<tr>
<td>Risk perception</td>
<td>0.758</td>
<td>0.751</td>
</tr>
</tbody>
</table>
Evaluation: System Overhead

• Overheads introduced by each module of RiskMon:

<table>
<thead>
<tr>
<th>Micro Benchmark</th>
<th>Average (sec.)</th>
<th>Std. (sec.)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Intelligence Aggregator</td>
<td>8.27</td>
<td>0.07</td>
</tr>
<tr>
<td>Baseline Learner</td>
<td>289.56</td>
<td>235.88</td>
</tr>
<tr>
<td>Risk Meter</td>
<td>0.55</td>
<td>0.17</td>
</tr>
</tbody>
</table>
Summary

• We have articulated the previously unidentified factors that are necessary for effective risk assessment of mobile applications
• We have proposed RiskMon, a continuous and automated risk assessment framework
• We have provided system evaluation and usability study of our methodology
• We would further extend RiskMon to support different risk mitigation strategies such as automated permission revocation
This work was supported in part by the NSF grant CNS-0916688.