A Machine-learning Approach for Classifying and Categorizing Android Sources and Sinks

Siegfried Rasthofer, Steven Arzt, Eric Bodden
Popular Android Apps Leaking Credible Data - Report Finds

Skype for Android leaks sensitive data

WhatsApp is leaking and messaging developers may

Angry Birds and other Mobile Gaming apps leaking your private information to NSA

Spying Birds

Secure Software Engineering Group
Dynamic Approaches:
TaintDroid [OSDI’10], Aurasium [USENIX’12], “Dr. Android and Mr. Hide“ [SPSM’12], etc.

Static Approaches:
ScanDroid [TR 09], DeD [SEC’11], CHEX [CCS’12], LeakMiner [WCSE’12], ScanDal [Most’12], AndroidLeaks [TRUST’12], SAAF [SAC’13], FlowDroid [PLDI’14], etc.
...but wait
sources

sinks

code analysis

report potential privacy leaks
<table>
<thead>
<tr>
<th>Method</th>
<th>TaintDroid</th>
<th>SCanDroid</th>
<th>DeD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Location.getLongitude()</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Location.getLatitude()</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Browser.getAllBookmarks()</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>SmsManager.sendTextMessage</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Log.d()</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>URL.openConnection()</td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>
Extracting Sources/Sinks

SuSi

List of Sources

List of Sinks

Android API

GoogleGlass API

Chromecast API

...
Machine-Learning Approach
Feature-Database: Classification

returns a value

specific return-type

"getter"

modifier

dataflow to return

```java
public static final String[] getVisitedHistory(ContentResolver cr) {
    Cursor c = null;
    String[] str = null;
    try {
        String[] projection = new String[] {
            History.URL,
        };
        c = cr.query(History.CONTENT_URI, projection, History.VISITS + " > 0", null, null);
        if (c == null) return new String[0];
        str = new String[c.getCount()];
        int i = 0;
        while (c.moveToNext()) {
            str[i] = c.getString(0);
            i++;
        }
    } catch (IllegalStateException e) {
        Log.e(LOGTAG, "getVisitedHistory", e);
        str = new String[0];
    } finally {
        if (c != null) c.close();
    }
    return str;
}
```
Feature-Database: Classification

Feature-Categories:

- Method name
- Method has parameters
- Method’s return type
- Parameter type
- Method modifiers
- Modifiers of declaring class
- Name of declaring class

- Dataflow to return value
- Dataflow from parameter to (abstract) sink
Feature-Database: Categorization
Evaluation

Ten-fold cross validation:

\[
\text{Precision} = \frac{TP}{TP + FP}
\]

\[
\text{Recall} = \frac{TP}{TP + FN}
\]
Evaluation

- Google Glass API:
  Precision: 98% and Recall: 100%

- Google Chromecast API:
  Precision and Recall: 100%

Manual validation:
Evaluation

![Graph showing the number of source methods against Android version]

- Bluetooth
- Location
- NFC
## Top Source/Sink Methods in Android-Malware

<table>
<thead>
<tr>
<th>Method</th>
<th>TaintDroid</th>
<th>SCanDroid</th>
<th>DeD</th>
</tr>
</thead>
<tbody>
<tr>
<td>BluetoothAdapter.getAddress()</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>WifiInfo.getMacAddress()</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Locale.getCountry()</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>WifiInfo.getSSID()</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>GsmCellLocation.getCid()</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>GsmCellLocation.getLac()</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>Location.getLongitude()</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Location.getLatitude()</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Browser.getAllBookmarks()</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SmsManager.sendTextMessage</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Log.d()</td>
<td>x</td>
<td>x</td>
<td></td>
</tr>
<tr>
<td>URL.openConnection()</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Open-Source on GitHub: https://github.com/secure-software-engineering/SuSi

Siegfried Rasthofer
Secure Software Engineering Group (EC-SPRIDE)
Email: siegfried.rasthofer@cased.de
Blog: http://sse-blog.ec-spride.de