**DRD10-X. Do not release apps that are debuggable**

This rule was developed in part by Stephanie Colton and Aashirya Kaushik at the October 20-22, 2017 OurCS Workshop (http://www.cs.cmu.edu/ourcs/register.html). For more information about this statement, see the About the OurCS Workshop page.

Android allows the attribute `android:debuggable` to be set to `true` in the manifest, so that the app can be debugged. By default this attribute is disabled, i.e., it is set to `false`, but it may be set to `true` to help with debugging during development of the app. However, an app should never be released with this attribute set to `true` as it enables users to gain access to details of the app that should be kept secure. With the attribute set to `true`, users can debug the app even without access to its source code.

**Noncompliant Code Example**

This noncompliant code example shows an app that has the `android:debuggable` attribute set to `true` being accessed to reveal sensitive data.

```
$ adb shell
shell@android:/ $ run-as com.example.someapp sh
shell@android:/data/data/com.example.someapp $ id
uid=10060(app_60) gid=10060(app_60)
shell@android:/data/data/com.example.someapp $ ls files/
secret_data.txt
shell@android:/data/data/com.example.someapp $ ls files/
secret_data.txt
password=GoogolPlex
account_number=31974286
```

Clearly, with the `android:debuggable` attribute set to `true`, sensitive date related to the app can be revealed to any user.

**Compliant Solution**

Ensure that the `android:debuggable` attribute is set to `false` before the app is released:

```
android:debuggable="false"
```

Note that some development environments (including Eclipse/ADT and Ant) automatically set `android:debuggable` to `true` for incremental or debugging builds but set it to `false` for release builds.

```
<configuration>
<compilation debug="true"/>
</configuration>
```

**Risk Assessment**

Releasing an app with its `android:debuggable` attribute set to `true` can leak sensitive information. In addition, the app is vulnerable to decompilation, resulting in alteration to source code. Attacker can leverage the additional information they gain from debugging output to mount attacks targeted on the framework, database, or other resources used by the application.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Severity</th>
<th>Likelihood</th>
<th>Remediation Cost</th>
<th>Priority</th>
<th>Level</th>
</tr>
</thead>
<tbody>
<tr>
<td>DRD10-X</td>
<td>High</td>
<td>Probable</td>
<td>Low</td>
<td>P18</td>
<td>L1</td>
</tr>
</tbody>
</table>

**Automated Detection**

Automatic detection of the setting of the `android:debuggable` attribute is straightforward. It is not feasible to automatically determine whether any data that might be revealed by debugging the app is sensitive.

**Related Vulnerabilities**

Search for vulnerabilities resulting from the violation of this rule on the CERT website.

**Related Guidelines**
<table>
<thead>
<tr>
<th>CWE</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>359</td>
<td>CWE 359: Exposure of Private Information</td>
</tr>
<tr>
<td>264</td>
<td>CWE 264: Permissions, Privileges, and Access Controls</td>
</tr>
</tbody>
</table>

**Bibliography**

<table>
<thead>
<tr>
<th>Title</th>
<th>URL</th>
</tr>
</thead>
</table>