




 Secrets


 ABAP


 Apex


 C


 C++


 CloudFormation


 COBOL


 C#


 CSS


 Flex


 Go

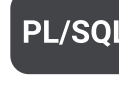
 HTML


 Java


 JavaScript


 Kotlin


 Kubernetes


 Objective C


 PHP


 PL/I


 PL/SQL


 Python


 RPG


 Ruby


 Scala


 Swift


 Terraform


 Text

 TypeScript

 T-SQL

 VB.NET

 VB6

 XML

XML



XML static code analysis

Unique rules to find Bugs and Code Smells in your XML code

- All rules 36
- Vulnerability 6
- Bug 5
- Security Hotspot 9
- Code Smell 16

Tags ▾

Search by name... 🔍

Receiving intents is security-sensitive	Security Hotspot
Restrict access to exported components with appropriate permissions	Vulnerability
"DefaultMessageListenerContainer" instances should not drop messages during restarts	Bug
"SingleConnectionFactory" instances should be set to "reconnectOnException"	Bug
Defining a single permission for read and write access of Content Providers is security-sensitive	Security Hotspot
Allowing application backup is security-sensitive	Security Hotspot
Requesting dangerous Android permissions is security-sensitive	Security Hotspot
Sections of code should not be commented out	Code Smell
Track uses of "FIXME" tags	Code Smell
Custom permissions should not be defined in the 'android.permission' namespace	Vulnerability
Having a permissive Cross-Origin Resource Sharing policy is security-sensitive	Security Hotspot
Delivering code in production with debug features activated is security-sensitive	Security Hotspot

Receiving intents is security-sensitive

Analyze your code

Security Hotspot

Critical

cwe owasp sans-top25 android

Android applications can receive broadcasts from the system or other applications. Receiving intents is security-sensitive. For example, it has led in the past to the following vulnerabilities:

- CVE-2019-1677
- CVE-2015-1275

Receivers can be declared in the manifest or in the code to make them context specific. If the receiver is declared in the manifest Android will start the application if it is not already running once a matching broadcast is received. The receiver is an entry point into the application.

Other applications can send potentially malicious broadcasts, so it is important to consider broadcasts as untrusted and to limit the applications that can send broadcasts to the receiver.

Permissions can be specified to restrict broadcasts to authorized applications. Restrictions can be enforced by both the sender and receiver of a broadcast. If permissions are specified when registering a broadcast receiver, then only broadcasters who were granted this permission can send a message to the receiver.

This rule raises an issue when a receiver is registered without specifying any "broadcast permission".

Ask Yourself Whether

- The data extracted from intents is not sanitized.
- Intents broadcast is not restricted.

There is a risk if you answered yes to any of those questions.

Recommended Secure Coding Practices

Restrict the access to broadcasted intents. See [Android documentation](#) for more information.

Sensitive Code Example

```
<receiver android:name=".MyBroadcastReceiver" android:exported="true">  <!-- Sensitive -->  <intent-filter>    <action android:name="android.intent.action.AIRPLANE_MODE" />  </intent-filter></receiver>
```

Compliant Solution

Enforce permissions:

```
<receiver android:name=".MyBroadcastReceiver"  android:permission="android.permission.SEND_SMS"  android:exported="true">  <intent-filter>    <action android:name="android.intent.action.AIRPLANE_MODE" />  </intent-filter></receiver>
```

Do not export the receiver and only receive system intents:

```
<receiver android:name=".MyBroadcastReceiver" android:exported="false">  <intent-filter>    <action android:name="android.intent.action.AIRPLANE_MODE" />  </intent-filter></receiver>
```

See

- Mobile AppSec Verification Standard - Platform Interaction Requirements
- OWASP Mobile Top 10 2016 Category M1 - Improper Platform Usage
- MITRE, CWE-925 - Improper Verification of Intent by Broadcast Receiver
- MITRE, CWE-926 - Improper Export of Android Application Components
- SANS Top 25 - Insecure Interaction Between Components
- Android documentation - Broadcast Overview - Security considerations and best practices

Available In:

