

Introduction to **Secure Coding Checker**

A diagnostic tool for identifying vulnerabilities in Android apps

Sony Digital Network Applications, Inc.

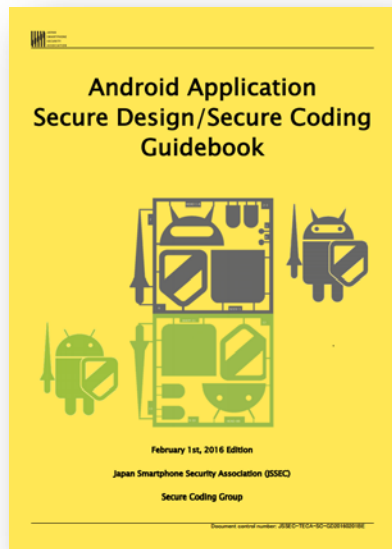
What is Secure Coding Checker?

- A web-based diagnostic tool that identifies vulnerabilities in Android apps, explains their causes, and proposes remedies (code revisions that address the vulnerability).
- The tool scans an .apk file, checks for the presence of vulnerabilities, and displays remedies—all in a few tens of seconds.

Key features of Secure Coding Checker

1. **Developers need only upload .apk files to get diagnostic results within a few tens of seconds**
2. **100% compliant with JSSEC secure coding guidelines**
3. **Provides comprehensive information, from vulnerability diagnoses to proposed fixes**
4. **Priced to allow easy access to developers (around 500,000 yen)**

Diagnostic criteria taken from the Android Secure Design / Secure Coding Guidebook



The *Android Secure Design / Secure Coding Guidebook*, issued by the Japan Smartphone Security Forum, is the canonical textbook of Android security and the basis of security recommendations issued by Japan's Ministry of Internal Affairs and Communications.



- Used by telecommunications carriers and many app vendors
- Used by app-development outsourcing firms to establish acceptance criteria

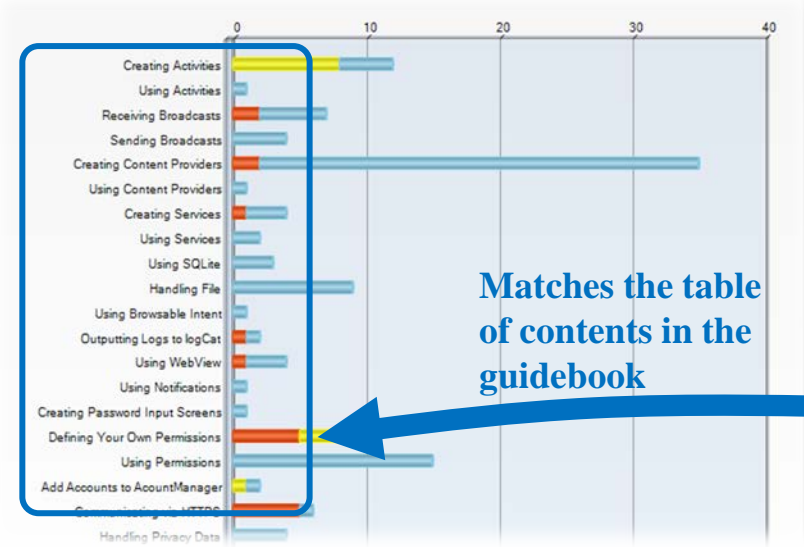
Updated one to two times annually

The PDF version may be downloaded free of charge.
<https://www.jssec.org/report/securecoding.html>



Visualizing the effects of security provisions

Apps are automatically analyzed, and their vulnerabilities are displayed in a graphical format.






Matches the table of contents in the guidebook

Guidebook

- 1. Introduction
- 2. Composition of the Guidebook
- 3. Basic Knowledge of Secure Design and Secure Coding
- 4. Using Technology in a Safe Way
 - 4.1. Creating/Using Activities
 - 4.2. Receiving/Sending Broadcasts
 - 4.3. Creating/Using Content Providers
 - 4.4. Creating/Using Services
 - 4.5. Using SQLite
 - 4.6. Handling Files
 - 4.7. UsingBrowsable Intent
 - 4.8. Outputting Log to LogCat
 - 4.9. Using WebView
 - 4.10. Using Notifications
- 5. How to use Security Functions
 - 5.1. Creating Password Input Screens
 - 5.2. Permission and Protection Level
 - 5.3. Add In-house Accounts to Account Manager
 - 5.4. Communicating via HTTPS
 - 5.5. Handling privacy data
 - 5.6. Using Cryptography
 - 5.7. Using fingerprint authentication features
- 6. Difficult Problems
 - 6.1. Risk of Information Leakage from Clipboard

Question-asking functionality

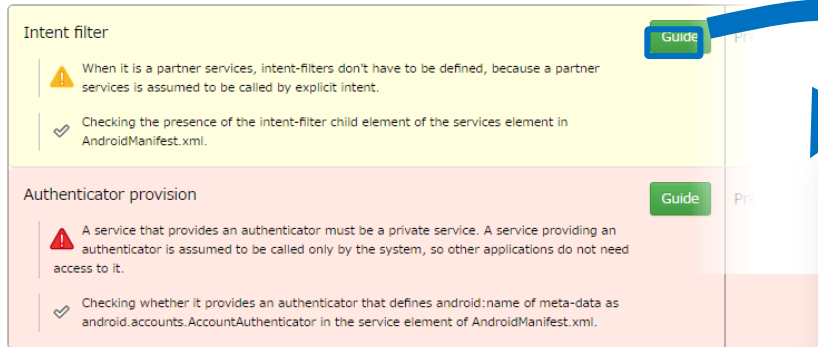
The tester asks developers questions to improve the accuracy of tests for vulnerabilities in specifications

<p>What is the activity type?</p> <p> Use of activities is determined by which applications use it. For example, private activities are used by only your application, public activities can be used by many other applications, and partner/in-house activities are used by a limited number of applications.</p>	<p>Guide</p> <ul style="list-style-type: none"><input type="radio"/> Unanswered<input type="radio"/> Private<input checked="" type="radio"/> Public<input type="radio"/> Partner<input type="radio"/> In-house
<p>Does it return sensitive information via the result intent?</p> <p> Sensitive information refers to information assets that should be protected in your application. Including sensitive information in call results means that the sensitive information is intentionally provided to other applications.</p>	<p>Guide</p> <ul style="list-style-type: none"><input checked="" type="radio"/> Unanswered<input type="radio"/> Yes<input type="radio"/> No
<p>Do you validate input values?</p> <p> Validating the safety of input data means ensuring that your application functions normally even if the input data is malicious or consists of any possible values. If the safety of input data is always guaranteed, select "No need to check".</p>	<p>Guide</p> <ul style="list-style-type: none"><input checked="" type="radio"/> Unanswered<input type="radio"/> Yes<input type="radio"/> No<input type="radio"/> No need to check

The information contained in the .apk file alone is insufficient to ensure accurate security assessments. Additional information regarding the specifications and design of the app is also required.

Full coverage from identification to correction of vulnerabilities

The tool suggests methods for fixing the vulnerabilities it detects



Intent filter

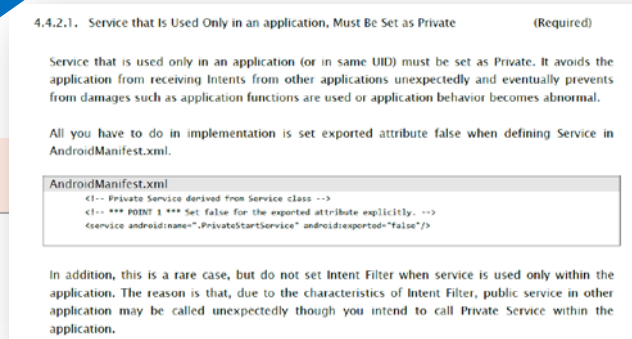
- ⚠ When it is a partner services, Intent-filters don't have to be defined, because a partner services is assumed to be called by explicit intent.
- ✓ Checking the presence of the intent-filter child element of the services element in AndroidManifest.xml.

Authenticator provision

- ⚠ A service that provides an authenticator must be a private service. A service providing an authenticator is assumed to be called only by the system, so other applications do not need access to it.
- ✓ Checking whether it provides an authenticator that defines android:name of meta-data as android.accounts.AccountAuthenticator in the service element of AndroidManifest.xml.

Jump directly to the section of the guidebook that discusses the security issue in question

Guidebook



4.4.2.1. Service that Is Used Only in an application, Must Be Set as Private (Required)

Service that is used only in an application (or in same UID) must be set as Private. It avoids the application from receiving Intents from other applications unexpectedly and eventually prevents from damages such as application functions are used or application behavior becomes abnormal.

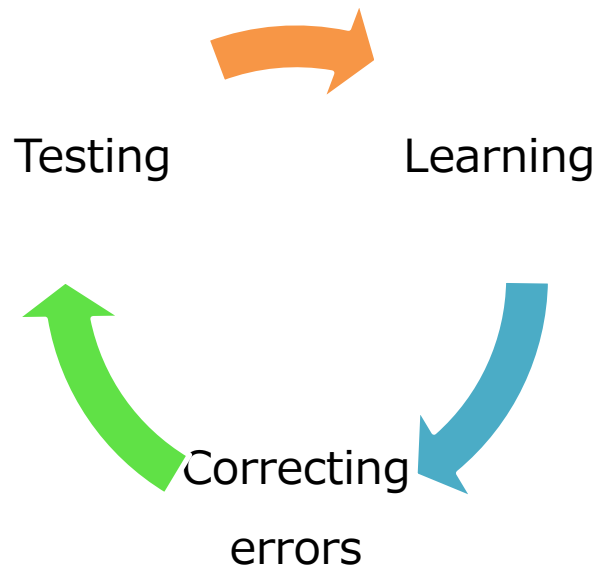
All you have to do in implementation is set exported attribute false when defining Service in AndroidManifest.xml.

```
AndroidManifest.xml
<!-- Private Service derived from Service class -->
<!-- *** PDHBT *** Set false for the exported attribute explicitly. -->
<service android:name=".PrivateStartService" android:reported="false"/>
```

In addition, this is a rare case, but do not set Intent Filter when service is used only within the application. The reason is that, due to the characteristics of Intent Filter, public service in other application may be called unexpectedly though you intend to call Private Service within the application.

The guidebook contains commercially-usable sample code that may be copied and pasted to fix vulnerabilities

Facilitating a virtuous cycle of learning



Advantage #1

Developers learn what is necessary—and *only* the minimal number of new concepts necessary—to complete their current task. This ensures that new concepts learned are immediately used for practical applications.

Advantage #2

Methods for fixing problems are always provided for all problems detected, ensuring no delays or lagtime in the code-revision process.

Advantage #3

Developers learn security concepts based on specific examples of vulnerabilities in apps they have designed themselves. This ensures that newly acquired knowledge becomes deeply embedded, making the learning process more effective.

How to contact us

Now offering free 2-week trials

Product website:

<http://www.sonydna.com/sdna/e/solution/scc.html>

Sony Digital Network Applications, Inc.

sdna-security-sales@jp.sony.com

Secure Coding Checker Department

FAQ

FAQ 1/3

Q: Are the items tested, and the content of the tests, ever updated?

A: Yes! The items tested and the content of the tests are updated whenever the JSSEC guidelines are updated.

Q: It is possible to run tests on obfuscated apps?

A: Yes, this is possible. We recommend that testing be performed before obfuscation.

Q: How does your billing system count the number of apps tested?

A: Apps are distinguished by their package names. Version upgrades of apps for which the package name remains unchanged are considered to be the same app as the original app.

Q: What languages are supported?

A: Both English and Japanese are supported. You may select your desired language from the login window.

FAQ 2/3

Q: What cloud service do you use?

A: We use Microsoft Azure.

Q: How are uploaded .apk files handled?

A: .apk files are deleted immediately after testing is complete. Only the test results are saved.

Q: Does the tester check program sections written in C or C++?

A: No, these sections lie outside the tester's purview. **Secure Coding Checker** only tests code sections written in Java.

FAQ 3/3

Q: How do you determine the number of apps that may be tested at any one time?

A: You may test a fixed number of apps within any given month. More specifically, suppose that in a given month you are working in a framework that allows for an upper limit of 3 apps, and suppose you have tested three apps named A, B, and C during that month. Then apps A, B, and C are the only apps you may test during that month. However, the apps included in the three-app limit may change every month; thus, next month you may delete apps A, B, and C and instead test three new apps D, E, and F. However, note that the test history for apps will be deleted together with those apps.

Q: Is there a limit on the number of times I may test an app?

A: No. You may test an app any number of times within the period of your contract.

Q: Is there a limit on the size of apps that may be tested?

A: Yes. The size limit on apps is 500 MB.



Sony Digital Network Applications, Inc.

21-28 Higashigotanda 2-chome, Shinagawa-ku, Tokyo, 141-0022 Japan
TEL : +81-50-3750-1897

Sony is a trademark of Sony corporation.

Other Sony product or service names are trademarks or registered trademarks of Sony Corporation or each Sony group company.

All other trademarks or registered trademarks are the property of their respective owners.