

Secrets

ABAP

Apex

C

C++

CloudFormation

COBOL

C#

CSS

Flex

Go

HTML

Java

JavaScript

Kotlin

Objective C

PHP

PL/I

PL/SQL

Python

RPG

Ruby

Scala

Swift

Terraform

Text

TypeScript

T-SQL

VB.NET

VB6

XML

Java

Java static code analysis

Unique rules to find Bugs, Vulnerabilities, Security Hotspots, and Code Smells in your JAVA code

All rules632

Vulnerability53

Bug154

Security Hotspot36

Code Smell389

Quick Fix42

Tags

Search by name...

Bug

Assertions should not compare an object to itself

Regex alternatives should not be redundant

Alternatives in regular expressions should be grouped when used with anchors

AssertJ methods setting the assertion context should come before an assertion

AssertJ configuration should be applied

JUnit5 test classes and methods should not be silently ignored

"ThreadLocal" variables should be cleaned up when no longer used

Strings and Boxed types should be compared using "equals()"

InputSteam.read() implementation should not return a signed byte

"compareTo" should not be overloaded

"iterator" should not return "this"

Broadcasting intents is security-sensitive

Analyze your code

Security Hotspot

Critical

cwe android owasp

In Android applications, broadcasting intents is security-sensitive. For example, it has led in the past to the following vulnerability:

- CVE-2018-9489

By default, broadcasted intents are visible to every application, exposing all sensitive information they contain.

This rule raises an issue when an intent is broadcasted without specifying any "receiver permission".

Ask Yourself Whether

- The intent contains sensitive information.
- Intent reception 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





```
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.os.UserHandle;
import android.support.annotation.RequiresApi;

public class MyIntentBroadcast {
    @RequiresApi(api = Build.VERSION_CODES.JELLY_BEAN_MR1)
    public void broadcast(Intent intent, Context context, BroadcastReceiver resultReceiver, String initialData, Bundle initialString broadcastPermission) {
        context.sendBroadcast(intent); // Sensitive
        context.sendBroadcastAsUser(intent, user); // Sensitive

        // Broadcasting intent with "null" for receiverPermission
        context.sendBroadcast(intent, null); // Sensitive
        context.sendBroadcastAsUser(intent, user, null); // Sensitive
        context.sendOrderedBroadcast(intent, null); // Sensitive
        context.sendOrderedBroadcastAsUser(intent, user, null, scheduler, initialCode, initialData, initialString);
    }
}
```

https://rules.sonarsource.com/java/RSPEC-5320

1/2

Map values should not be replaced unconditionally  Bug
Week Year ("YYYY") should not be used for date formatting  Bug
Exceptions should not be created without being thrown  Bug
Collection sizes and array length comparisons should make sense  Bug

Compliant Solution

```
import android.content.BroadcastReceiver;
import android.content.Context;
import android.content.Intent;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.os.UserHandle;
import android.support.annotation.RequiresApi;

public class MyIntentBroadcast {
    @RequiresApi(api = Build.VERSION_CODES.JELLY_BEAN_MR1)
    public void broadcast(Intent intent, Context context, UserHandle user,
        BroadcastReceiver resultReceiver,
        String initialData, Bundle initialData, String broadcastPermission) {

        context.sendBroadcast(intent, broadcastPermission);
        context.sendBroadcastAsUser(intent, user, broadcastPermission, initialData, initialData, broadcastPermission);
        context.sendOrderedBroadcast(intent, broadcastPermission, resultReceiver, initialData, initialData, broadcastPermission, user, broadcastPermission);
        context.sendOrderedBroadcastAsUser(intent, user, broadcastPermission, resultReceiver, initialData, initialData, broadcastPermission, user, broadcastPermission);
    }
}
```

See

- [OWASP Top 10 2021 Category A4](#) - Insecure Design
- [Mobile AppSec Verification Standard](#) - Platform Interaction Requirements
- [OWASP Mobile Top 10 2016 Category M1](#) - Improper Platform Usage
- [MITRE, CWE-927](#) - Use of Implicit Intent for Sensitive Communication
- [Android documentation](#) - Broadcast Overview - Security considerations and best practices

Available In:
 | 