

-  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



Kotlin static code analysis

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

All rules 98 Vulnerability 10 Bug 17 Security Hotspot 15 Code Smell 56

Tags ▾

Search by name... 🔍

Hard-coded credentials are security-sensitive	Security Hotspot
Cipher algorithms should be robust	Vulnerability
Encryption algorithms should be used with secure mode and padding scheme	Vulnerability
Server hostnames should be verified during SSL/TLS connections	Vulnerability
Server certificates should be verified during SSL/TLS connections	Vulnerability
Cryptographic keys should be robust	Vulnerability
Weak SSL/TLS protocols should not be used	Vulnerability
"SecureRandom" seeds should not be predictable	Vulnerability
Cipher Block Chaining IVs should be unpredictable	Vulnerability
Hashes should include an unpredictable salt	Vulnerability
Regular expressions should be syntactically valid	Bug
"runFinalizersOnExit" should not be called	Bug

Delivering code in production with debug features activated is security-sensitive

Analyze your code

Security Hotspot Minor cwe error-handling debug user-experience owasp

Delivering code in production with debug features activated is security-sensitive. It has led in the past to the following vulnerabilities:

- [CVE-2018-1999007](#)
- [CVE-2015-5306](#)
- [CVE-2013-2006](#)

An application’s debug features enable developers to find bugs more easily and thus facilitate also the work of attackers. It often gives access to detailed information on both the system running the application and users.

Ask Yourself Whether

- the code or configuration enabling the application debug features is deployed on production servers or distributed to end users.
- the application runs by default with debug features activated.

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

Recommended Secure Coding Practices

Do not enable debug features on production servers or applications distributed to end users.

Sensitive Code Example

`WebView.setWebContentsDebuggingEnabled(true)` for Android enables debugging support:

```
import android.webkit.WebView

WebView.setWebContentsDebuggingEnabled(true) // Sensitive
```

Compliant Solution

`WebView.setWebContentsDebuggingEnabled(false)` for Android disables debugging support:

```
import android.webkit.WebView











WebView.setWebContentsDebuggingEnabled(false)
```

See

- [OWASP Top 10 2021 Category A5](#) - Security Misconfiguration
- [OWASP Top 10 2017 Category A3](#) - Sensitive Data Exposure
- [MITRE, CWE-489](#) - Active Debug Code
- [MITRE, CWE-215](#) - Information Exposure Through Debug Information

Available In:

sonarcloud | sonarqube

<div><div>"ScheduledThreadPoolExecutor" should not have 0 core threads</div><div> Bug</div></div>
<div><div>Jump statements should not occur in "finally" blocks</div><div> Bug</div></div>
<div><div>Using clear-text protocols is security-sensitive</div><div> Security Hotspot</div></div>
<div><div>Accessing Android external storage is security-sensitive</div><div> Security Hotspot</div></div>
<div><div>Receiving intents is security-sensitive</div><div> Security Hotspot</div></div>
<div><div>Broadcasting intents is security-sensitive</div><div> Security Hotspot</div></div>
<div><div>Using weak hashing algorithms is security-sensitive</div><div> Security Hotspot</div></div>
<div><div>Using pseudorandom number generators (PRNGs) is security-sensitive</div><div> Security Hotspot</div></div>
<div><div>Empty lines should not be tested with regex MULTILINE flag</div><div> Code Smell</div></div>
<div><div>Cognitive Complexity of functions should not be too high</div><div> Code Smell</div></div>