

-  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

Enabling file access for WebViews is security-sensitive

Analyze your code

Security Hotspot Major cwe owasp android

WebViews can be used to display web content as part of a mobile application. A browser engine is used to render and display the content. Like a web application a mobile application that uses WebViews can be vulnerable to Cross-Site Scripting if untrusted code is rendered.

If malicious JavaScript code in a WebView is executed this can leak the contents of sensitive files when access to local files is enabled.

Ask Yourself Whether

- No local files have to be accessed by the Webview.
- The WebView contains untrusted data that could cause harm when rendered.

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

Recommended Secure Coding Practices

It's recommended to disable access to local files for WebViews unless it is necessary. In the case of a successful attack through a Cross-Site Scripting vulnerability the attackers attack surface decreases drastically if no files can be read out.

Sensitive Code Example

```
import android.webkit.WebView

val webView: WebView = findViewById(R.id.webview)
webView.getSettings().setAllowContentAccess(true) // Sensitive
webView.getSettings().setAllowFileAccess(true) // Sensitive
```

Compliant Solution

```
import android.webkit.WebView

val webView: WebView = findViewById(R.id.webview)
webView.getSettings().setAllowContentAccess(false)
webView.getSettings().setAllowFileAccess(false)
```

See

- [OWASP Top 10 2021 Category A3](#) - Injection
- [OWASP Top 10 2017 Category A6](#) - Security Misconfiguration
- [OWASP Top 10 2017 Category A7](#) - Cross-Site Scripting (XSS)
- [MITRE, CWE-79](#) - Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

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>