

COBOL

C# **CSS**

Flex

Go

5 **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

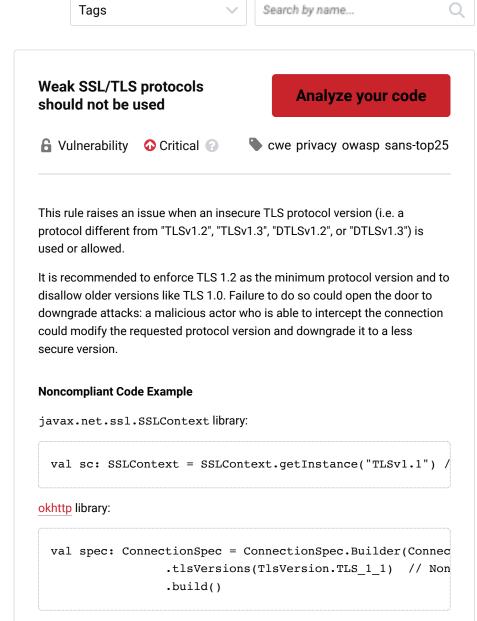


Kotlin static code analysis

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

Code Smell (56) All rules 98 6 Vulnerability (10) **R** Bug (17) Security Hotspot (15)

Hard-coded credentials are securitysensitive 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 Hashes should include an unpredictable salt Vulnerability Regular expressions should be syntactically valid Rug Bug "runFinalizersOnExit" should not be called 👬 Bug



Compliant Solution

javax.net.ssl.SSLContext library:

val sc: SSLContext = SSLContext.getInstance("TLSv1.2") /

okhttp library:

val spec: ConnectionSpec = ConnectionSpec.Builder(ConnectionSpe .tlsVersions(TlsVersion.TLS 1 2) // Com .build()

See

- OWASP Top 10 2021 Category A2 Cryptographic Failures
- OWASP Top 10 2021 Category A7 Identification and Authentication **Failures**
- OWASP Top 10 2017 Category A3 Sensitive Data Exposure
- OWASP Top 10 2017 Category A6 Security Misconfiguration
- Mobile AppSec Verification Standard Network Communication Requirements
- OWASP Mobile Top 10 2016 Category M3 Insecure Communication
- MITRE, CWE-327 Inadequate Encryption Strength
- MITRE, CWE-326 Use of a Broken or Risky Cryptographic Algorithm
- SANS Top 25 Porous Defenses
- SSL and TLS Deployment Best Practices Use secure protocols

Available In:

sonarlint ⊕ | sonarcloud 👌 | sonarqube |

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

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR, SONARSOURCE, SONARLINT, SONARQUBE and SONARCLOUD are trademarks of SonarSource S.A. All other trademarks and copyrights are the property of their respective owners. All rights are expressly reserved.

Privacy Policy