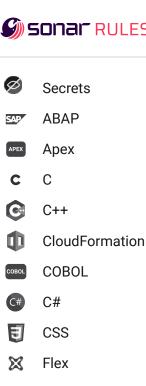
Search by name...





Go

HTML 5

Java

**JavaScript** 

Kotlin

Kubernetes

Objective C

PHP

PL/I

PL/SQL

Python

**RPG** 

Ruby

Scala

Swift

Terraform

Text

**TypeScript** 

T-SQL

**VB.NET** 

VB6

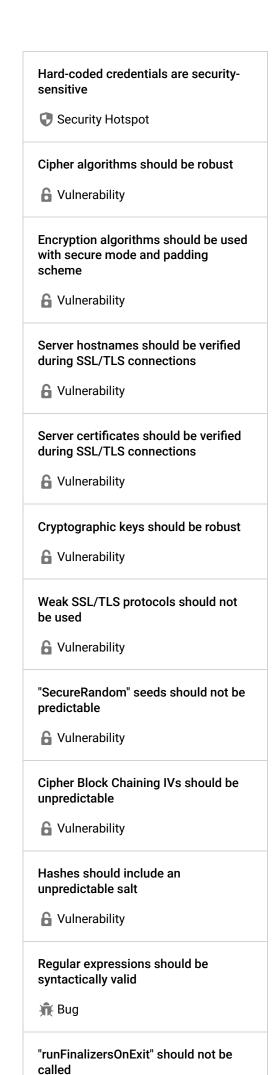
**XML** 



## Kotlin static code analysis

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

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



📆 Bug

🖷 Bug

"ScheduledThreadPoolExecutor"

Jump statements should not occur in

should not have 0 core threads

```
Unnecessary imports should be
                                              Analyze your code
removed
☼ Code Smell ♥ Minor ②
                               unused
The imports part of a file should be handled by the Integrated Development
Environment (IDE), not manually by the developer.
Unused and useless imports should not occur if that is the case.
Leaving them in reduces the code's readability, since their presence can be
confusing.
Noncompliant Code Example
  package my.company
  import kotlin.String
                                      // Noncompliant; "kotlin" cla
                                      // Noncompliant; same-package
  import my.company.SomeClass
  import java.io.File
                                      // Noncompliant; "File" is no
  import my.company2.SomeType
  class ExampleClass {
    val someString = ""
    val something = SomeType()
  }
Compliant Solution
  package my.company
  import java.io.File
  import my.company2.SomeType
  class ExampleClass {
    val someString = ""
    val something = SomeType()
    lateinit var fileUsage: File
Exceptions
Imports for types mentioned in KDoc comments are ignored.
```

© 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** 

sonarlint ⊕ | sonarcloud ↔ | sonarqube

Available In:

"finally" blocks
74 13
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
Cognitive Complexity of functions should not be too high
Code Smell
String literals should not be duplicated
☼ Code Smell