

-  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

"if ... else if" constructs should end with "else" clauses

Analyze your code

 Code Smell  Critical 

This rule applies whenever an `if` statement is followed by one or more `else if` statements; the final `else if` should be followed by an `else` statement.

The requirement for a final `else` statement is defensive programming.

The `else` statement should either take appropriate action or contain a suitable comment as to why no action is taken. This is consistent with the requirement to have a final `default` clause in a `switch` statement.

Noncompliant Code Example

```
if (x == 0) {
    doSomething()
} else if (x == 1) {
    doSomethingElse()
}
```

Compliant Solution

```
if (x == 0) {
    doSomething()
} else if (x == 1) {
    doSomethingElse()
} else {
    throw IllegalStateException()
}
```

Exceptions

When all branches of an `if-else if` end with `return`, `break` or `throw`, the code that comes after the `if` implicitly behaves as if it was in an `else` clause. This rule will therefore ignore that case.

Available In:
 |  | 

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