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

Equals method should be overridden in data classes containing array fields

Analyze your code

Bug Major

In data classes, the default behavior of the `equals()` method is to check the equality by field values. This works well for primitive fields or fields, whose type overrides `equals()`, but this behavior doesn't work as expected for array fields.

By default, array fields are compared by their reference, so overriding `equals()` is highly recommended to ensure a deep equality check. The same applies to the `hashcode()` method.

This rule reports an issue if a record class has an array field and is not overriding `equals()` or `hashcode()` methods.

Noncompliant Code Example

```
data class Person(val names: Array<String>, val age: Int
```

Compliant Solution

```
data class Person(val names: Array<String>, val age: Int
    override fun equals(other: Any?): Boolean {
        if (this === other) return true
        if (javaClass != other?.javaClass) return false

        other as Person

        if (!names.contentEquals(other.names)) return fa
        if (age != other.age) return false

        return true
    }

    override fun hashCode(): Int {
        var result = names.contentHashCode()
        result = 31 * result + age
        return result
    }
}
```

Available In:

sonarlint sonarcloud sonarqube

"ScheduledThreadPoolExecutor" should not have 0 core threads

Bug

Jump statements should not occur in "finally" blocks

Bug

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

Code Smell