sonar RULES

Products

Secrets
ABAP
Apex
C
C++
CloudFormation
COBOL
C#
CSS
Flex
Go
HTML
Java
JavaScript
Kotlin
Objective C
PHP
PL/I
PL/SQL
Python
RPG
Ruby
Scala
Swift
Terraform
Text
TypeScript
T-SQL
VB.NET
VB6
XML

Java static code analysis

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

All rules 632 | Vulnerability 53 | Bug 154 | Security Hotspot 36 | Code Smell 389 | Quick Fix 42

Tags

Search by name...

JSON operations should not be vulnerable to injection attacks
Vulnerability

XML signatures should be validated securely
Vulnerability

XML parsers should not be vulnerable to Denial of Service attacks
Vulnerability

XML parsers should not load external schemas
Vulnerability

Mobile database encryption keys should not be disclosed
Vulnerability

Reflection should not be vulnerable to injection attacks
Vulnerability

Authorizations should be based on strong decisions
Vulnerability

OpenSAML2 should be configured to prevent authentication bypass
Vulnerability

Server-side requests should not be vulnerable to forging attacks
Vulnerability

Collections should not be modified while they are iterated
Bug

Equals method should be overridden in records containing array fields
Bug

Reflection should not be used to increase accessibility of records' fields

Locks should be released

Analyze your code

Bug Critical cwe multi-threading

If a lock is acquired and released within a method, then it must be released along all execution paths of that method.

Failing to do so will expose the conditional locking logic to the method's callers and hence be deadlock-prone.

Noncompliant Code Example

```
public class MyClass {
  public void doSomething() {
    Lock lock = new Lock();
    lock.lock(); // Noncompliant
    if (isInitialized()) {
      // ...
      lock.unlock();
    }
  }
}
```

Compliant Solution

```
public class MyClass {
  public void doSomething() {
    Lock lock = new Lock();
    if (isInitialized()) {
      lock.lock();
      // ...
      lock.unlock();
    }
  }
}
```

See

- MITRE, CWE-459 - Incomplete Cleanup

Available In:

sonarlint sonarcloud sonarqube

Bug

AssertJ assertions with "Consumer" arguments should contain assertion inside consumers

Bug

The regex escape sequence \cX should only be used with characters in the @-_ range

Bug

Regular expressions should not overflow the stack

Bug