Security

Hotspot

Tags

int y = f(1);

int y = f(1);

@Test

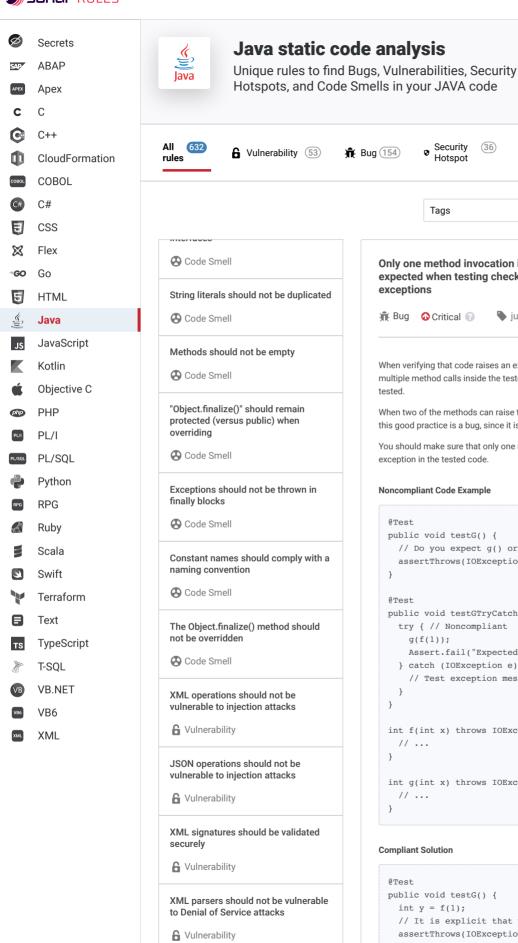
try {

(36)



Products >

O Quick 42 Fix



XML parsers should not load external

schemas

Vulnerability

```
Only one method invocation is
expected when testing checked
                                              Analyze your code
exceptions
Rug Oritical 🕝
                          junit tests
When verifying that code raises an exception, a good practice is to avoid having
multiple method calls inside the tested code, to be explicit about what is exactly
When two of the methods can raise the same checked exception, not respecting
this good practice is a bug, since it is not possible to know what is really tested.
You should make sure that only one method can raise the expected checked
exception in the tested code.
Noncompliant Code Example
 public void testG() {
    // Do you expect g() or f() throwing the exception?
    assertThrows(IOException.class, () -> g(f(1)) ); // Noncom
 @Test
 public void testGTryCatchIdiom() {
    try { // Noncompliant
      g(f(1));
      Assert.fail("Expected an IOException to be thrown");
    } catch (IOException e) {
       // Test exception message...
 int f(int x) throws IOException {
    // ...
 int g(int x) throws IOException {
    // ...
Compliant Solution
  @Test
 public void testG() {
```

// It is explicit that we expect an exception from g() and

assertThrows(IOException.class, () -> g(y));

public void testGTryCatchIdiom() {

(389)

Search by name.

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

```
g(y);
Assert.fail("Expected an IOException to be thrown");
} catch (IOException e) {
// Test exception message...
}

Available In:
sonarlint ⊕ | sonarcloud ♠ | sonarqube
```

© 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