

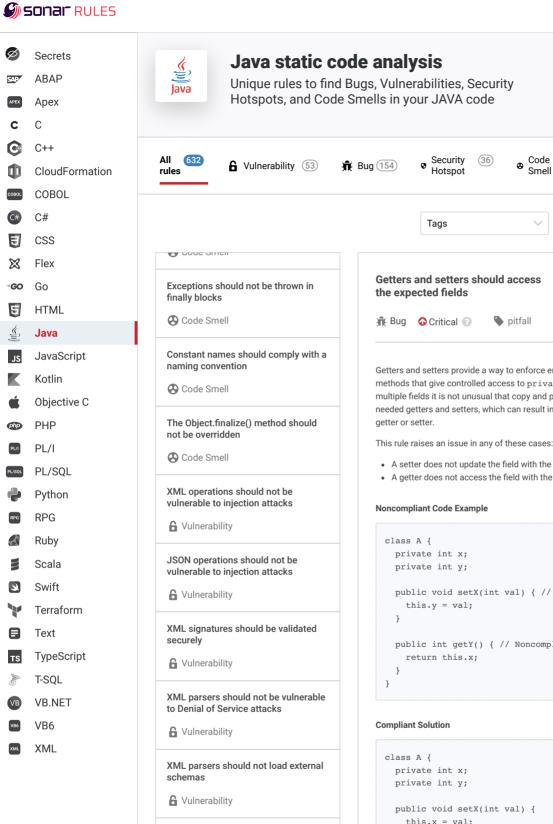
Products >

O Quick 42 Fix

Q

(389)

Search by name.



Mobile database encryption keys should not be disclosed

Reflection should not be vulnerable to

Authorizations should be based on

Vulnerability

injection attacks Vulnerability

strong decisions Vulnerability

```
Analyze your code
Getters and setters provide a way to enforce encapsulation by providing public
methods that give controlled access to private fields. However in classes with
multiple fields it is not unusual that copy and paste is used to quickly create the
needed getters and setters, which can result in the wrong field being accessed by a
This rule raises an issue in any of these cases:
 • A setter does not update the field with the corresponding name.
 • A getter does not access the field with the corresponding name.
    public void setX(int val) { // Noncompliant: field 'x' is
    public int getY() { // Noncompliant: field 'y' is not used
      this.x = val;
    public int getY() {
       return this.y;
  }
 Available In:
 sonarlint ⊕ | sonarcloud ↔ | sonarqube
```

© 2008-2022 SonarSource S.A., Switzerland. All content is copyright protected. SONAR,

Java static code anal
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

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