




 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... 🔍

@Transactional methods must call "setComplete" on their "SessionStatus" objects

Bug

"wait" should not be called when multiple locks are held

Bug

"PreparedStatement" and "ResultSet" methods should be called with valid indices

Bug

Files opened in append mode should not be used with ObjectOutputStream

Bug

"wait(...)" should be used instead of "Thread.sleep(...)" when a lock is held

Bug

Printf-style format strings should not lead to unexpected behavior at runtime

Bug

Methods "wait(...)", "notify()" and "notifyAll()" should not be called on Thread instances

Bug

Methods should not call same-class methods with incompatible "@Transactional" values

Bug

Recursion should not be infinite

Bug

Loops should not be infinite

Bug

Double-checked locking should not be used

Bug

### XML parsers should not be vulnerable to XXE attacks

Analyze your code

Vulnerability

Blocker

cwe owasp

XML standard allows the use of entities, declared in the DOCTYPE of the document, which can be [internal](#) or [external](#).

When parsing the XML file, the content of the external entities is retrieved from an external storage such as the file system or network, which may lead, if no restrictions are put in place, to arbitrary file disclosures or [server-side request forgery \(SSRF\)](#) vulnerabilities.

It's recommended to limit resolution of external entities by using one of these solutions:

- If DOCTYPE is not necessary, completely disable all DOCTYPE declarations.
- If external entities are not necessary, completely disable their declarations.
- If external entities are necessary then:
  - Use XML processor features, if available, to authorize only required protocols (eg: https).
  - And use an entity resolver (and optionally an XML Catalog) to resolve only trusted entities. == Noncompliant Code Example

For [DocumentBuilder](#), [SAXParser](#), [XMLInput](#), [Transformer](#) and [Schema](#) JAPX factories:

```
DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
SAXParserFactory factory = SAXParserFactory.newInstance(); //
XMLInputFactory factory = XMLInputFactory.newInstance(); //
TransformerFactory factory = javax.xml.transform.TransformerFactory.newInstance();
SchemaFactory factory = SchemaFactory.newInstance(XMLConstants.W3C_XML_SCHEMA_NS_URI);
```

For [Dom4j](#) library:

```
SAXReader xmlReader = new SAXReader(); // Noncompliant
```

For [Jdom2](#) library:

```
SAXBuilder builder = new SAXBuilder(); // Noncompliant
```





#### Compliant Solution

For [DocumentBuilder](#), [SAXParser](#), [XMLInput](#), [Transformer](#) and [Schema](#) JAPX factories:

```
DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
// to be compliant, completely disable DOCTYPE declaration:
factory.setFeature("http://apache.org/xml/features/disallow-doctype-decl", true);
// or completely disable external entities declarations:
factory.setFeature("http://xml.org/sax/features/external-general-entities", false);
```

https://rules.sonarsource.com/java/RSPEC-2755

1/2

resources should be closed
 Bug
Hard-coded credentials are security-sensitive
 Security Hotspot
Methods returns should not be invariant
 Code Smell
"ThreadGroup" should not be used
 Code Smell
"clone" should not be overridden

```
factory.setFeature("http://xml.org/sax/features/external-par
// or prohibit the use of all protocols by external entities
factory.setAttribute(XMLConstants.ACCESS_EXTERNAL_DTD, "");
factory.setAttribute(XMLConstants.ACCESS_EXTERNAL_SCHEMA, ""
// or disable entity expansion but keep in mind that this do
// and this solution is not correct for OpenJDK < 13 due to
factory.setExpandEntityReferences(false);

SAXParserFactory factory = SAXParserFactory.newInstance();
// to be compliant, completely disable DOCTYPE declaration:
factory.setFeature("http://apache.org/xml/features/disallow-
// or completely disable external entities declarations:
factory.setFeature("http://xml.org/sax/features/external-gen
factory.setFeature("http://xml.org/sax/features/external-par
// or prohibit the use of all protocols by external entities
SAXParser parser = factory.newSAXParser(); // Noncompliant
parser.setProperty(XMLConstants.ACCESS_EXTERNAL_DTD, "");
parser.setProperty(XMLConstants.ACCESS_EXTERNAL_SCHEMA, "");

XMLInputFactory factory = XMLInputFactory.newInstance();
// to be compliant, completely disable DOCTYPE declaration:
factory.setProperty(XMLInputFactory.SUPPORT_DTD, false);
// or completely disable external entities declarations:
factory.setProperty(XMLInputFactory.IS_SUPPORTING_EXTERNAL_E
// or prohibit the use of all protocols by external entities
factory.setProperty(XMLConstants.ACCESS_EXTERNAL_DTD, "");
factory.setProperty(XMLConstants.ACCESS_EXTERNAL_SCHEMA, "")

TransformerFactory factory = javax.xml.transform.Transformer
// to be compliant, prohibit the use of all protocols by ext
factory.setAttribute(XMLConstants.ACCESS_EXTERNAL_DTD, "");
factory.setAttribute(XMLConstants.ACCESS_EXTERNAL_STYLESHEET

SchemaFactory factory = SchemaFactory.newInstance(XMLConstan
// to be compliant, completely disable DOCTYPE declaration:
factory.setFeature("http://apache.org/xml/features/disallow-
// or prohibit the use of all protocols by external entities
factory.setProperty(XMLConstants.ACCESS_EXTERNAL_DTD, "");
factory.setProperty(XMLConstants.ACCESS_EXTERNAL_SCHEMA, "")
```

For [Dom4j](#) library:

```
SAXReader xmlReader = new SAXReader();
xmlReader.setFeature("http://apache.org/xml/features/disallo
```

For [Jdom2](#) library:

```
SAXBuilder builder = new SAXBuilder();
builder.setProperty(XMLConstants.ACCESS_EXTERNAL_DTD, "");
builder.setProperty(XMLConstants.ACCESS_EXTERNAL_SCHEMA, "")
```

See

- [OWASP Top 10 2021 Category A5](#) - Security Misconfiguration
- [Oracle Java Documentation](#) - XML External Entity Injection Attack
- [OWASP Top 10 2017 Category A4](#) - XML External Entities (XXE)
- [OWASP XXE Prevention Cheat Sheet](#)
- [MITRE, CWE-611](#) - Information Exposure Through XML External Entity Reference
- [MITRE, CWE-827](#) - Improper Control of Document Type Definition

Available In:

