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

<XML/>

# XML static code analysis

Unique rules to find Bugs and Code Smells in your XML code

All rules 36 | Vulnerability 6 | Bug 5 | Security Hotspot 9 | Code Smell 16

Tags

Search by name...

Dependencies should not have "system" scope
Bug

XML files containing a prolog header should start with "<?xml" characters
Bug

Using clear-text protocols is security-sensitive
Security Hotspot

Receiving intents is security-sensitive
Security Hotspot

Restrict access to exported components with appropriate permissions
Vulnerability

"DefaultMessageListenerContainer" instances should not drop messages during restarts
Bug

"SingleConnectionFactory" instances should be set to "reconnectOnException"
Bug

Defining a single permission for read and write access of Content Providers is security-sensitive
Security Hotspot

Allowing application backup is security-sensitive
Security Hotspot

Requesting dangerous Android permissions is security-sensitive
Security Hotspot

Sections of code should not be commented out
Code Smell

Track uses of "FIXME" tags
Code Smell

## Dependencies should not have "system" scope

Analyze your code

Bug | Critical | maven lock-in

`system` dependencies are sought at a specific, specified path. This drastically reduces portability because if you deploy your artifact in an environment that's not configured just like yours is, your code won't work.

### Noncompliant Code Example

```
<dependency>
  <groupId>javax.sql</groupId>
  <artifactId>jdbc-stdext</artifactId>
  <version>2.0</version>
  <scope>system</scope>  <!-- Noncompliant -->
  <systemPath>/usr/bin/lib/rt.jar</systemPath>  <!-- remove this -->
</dependency>
```

Available In:

sonarlint | sonarcloud | sonarqube