Secrets
ABAP
Apex
AzureResourceManager
C
C#
C++
CloudFormation
COBOL
CSS
Dart
Docker
Flex
Go
HTML
Java
JavaScript
JCL
Kotlin
Kubernetes
Objective C
PHP
PL/I
PL/SQL
Python
RPG
Ruby
Scala
Swift
Terraform
Text
TypeScript

Java static code analysis

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

All rules 704 | Vulnerability 58 | Bug 173 | Security Hotspot 38 | Code Smell 435 | Quick Fix 63

Tags | Impact | Clean code attribute | Search by name...

Credentials should not be hard-coded
Vulnerability

Components should not be vulnerable to intent redirection
Vulnerability

Extracting archives should not lead to zip slip vulnerabilities
Vulnerability

Server-side templates should not be vulnerable to injection attacks
Vulnerability

Dynamic code execution should not be vulnerable to injection attacks
Vulnerability

NoSQL operations should not be vulnerable to injection attacks
Vulnerability

HTTP request redirections should not be open to forging attacks
Vulnerability

Deserialization should not be vulnerable to injection attacks

XML parsers should not be vulnerable to XXE attacks

Analyze your code

Intentionality - Complete | Security

Vulnerability | Blocker | cwe symbolic-execution

This vulnerability allows the usage of external entities in XML.

Why is this an issue? | How can I fix it? | More Info

External Entity Processing allows for XML parsing with the involvement of external entities. However, when this functionality is enabled without proper precautions, it can lead to a vulnerability known as XML External Entity (XXE) attack.

What is the potential impact?

Exposing sensitive data

One significant danger of XXE vulnerabilities is the potential for sensitive data exposure. By crafting malicious XML payloads, attackers can reference external entities that contain sensitive information, such as system files, database credentials, or configuration files. When these entities are processed during XML parsing, the attacker can extract the contents and gain unauthorized access to sensitive data. This poses a severe threat to the confidentiality of critical information.

Exhausting system resources

T-SQL
VB.NET
VB6
XML

Vulnerability

Endpoints should not be vulnerable to reflected cross-site scripting (XSS) attacks

Vulnerability

Database queries should not be vulnerable to injection attacks

Vulnerability

A secure password should be used when connecting to a database

Vulnerability

XPath expressions should not be vulnerable to injection attacks

Vulnerability

I/O function calls should not be vulnerable to path injection attacks

Vulnerability

LDAP queries should not be vulnerable to injection attacks

Vulnerability

OS commands should not be vulnerable to command injection attacks

Vulnerability

"@SpringBootApplication" and "@ComponentScan" should not be used in the default package

Bug

Another consequence of XXE vulnerabilities is the potential for denial-of-service attacks. By exploiting the ability to include external entities, attackers can construct XML payloads that cause resource exhaustion. This can overwhelm the system's memory, CPU, or other critical resources, leading to system unresponsiveness or crashes. A successful DoS attack can disrupt the availability of services and negatively impact the user experience.

Forging requests

XXE vulnerabilities can also enable Server-Side Request Forgery (SSRF) attacks. By leveraging the ability to include external entities, an attacker can make the vulnerable application send arbitrary requests to other internal or external systems. This can result in unintended actions, such as retrieving data from internal resources, scanning internal networks, or attacking other systems. SSRF attacks can lead to severe consequences, including unauthorized data access, system compromise, or even further exploitation within the network infrastructure.

Available In:
sonarlint | sonarcloud | sonarqube

© 2008-2024 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.

Sonar helps developers write Clean Code.
Privacy Policy | Cookie Policy