






























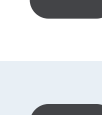


-  Secrets
-  ABAP
-  Apex
-  C
-  C++
-  CloudFormation
-  COBOL
-  C#
-  CSS
-  Flex
-  Go
-  HTML
-  Java
-  JavaScript
-  Kotlin
-  Kubernetes
-  Objective C
-  PHP
-  PL/I
-  PL/SQL
-  Python
-  RPG
-  Ruby
-  Scala
-  Swift
-  Terraform
-  Text
-  TypeScript
-  T-SQL
-  VB.NET
-  VB6
-  XML
- XML

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...



Struts validation forms should have unique names

 Vulnerability

Default EJB interceptors should be declared in "ejb-jar.xml"

 Vulnerability

Hard-coded credentials are security-sensitive

 Security Hotspot

Defined filters should be used

 Vulnerability

Basic authentication should not be used

 Vulnerability

Hibernate should not update database schemas

 Bug

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"

Struts validation forms should have unique names

Analyze your code

 Vulnerability

 Blocker



 cwe struts

According to the Common Weakness Enumeration,

If two validation forms have the same name, the Struts Validator arbitrarily chooses one of the forms to use for input validation and discards the other. This decision might not correspond to the programmer's expectations...

In such a case, it is likely that the two forms should be combined. At the very least, one should be removed.

Noncompliant Code Example

```
<form-validation>
  <formset>
    <form name="BookForm"> ... </form>
    <form name="BookForm"> ... </form>  <!-- Noncompliant -->
  </formset>
</form-validation>
```

Compliant Solution

```
<form-validation>
  <formset>
    <form name="BookForm"> ... </form>
  </formset>
</form-validation>
```

See

- [MITRE, CWE-102](#) - Struts: Duplicate Validation Forms
- [OWASP, Improper Data Validation](#) - Struts: Duplicate Validation Forms

Available In:

sonarlint



sonarcloud



sonarqube

