CWE-91 : XML Injection

This guide explains XML injection in more detail.

Contents:

XML Injection (CWE-91)

CWE-91 describes **XML Injection** as follows:

> "The software does not properly neutralize special elements that are used in XML, allowing attackers to modify the syntax, content, or commands of the XML before it is processed by an end system."

XML injection (CWE-91) attacks can be successful if the app does not properly neutralize special elements that are used in XML, allowing attackers to modify the syntax, content, or commands of the XML before it is processed by an end system.

By using special *metacharacters*, an attacker might be able to discover information about the XML structure, and then it will be able to try to inject XML data and tags (Tag injection).

If the software allows untrusted inputs to control part or all of an XSLT stylesheet, an attacker may change the structure and content of the resulting XML. If the resulting XML ends in a browser, the attacker may choose contents to launch cross-site scripting attacks or execute operations at the server with a victim's identity allowed by the browser's same-origin policy (a variant of the cross-site request forgery attack). The attacker may also use this flaw to launch attacks targeted at the server, like fetching content from arbitrary files, running arbitrary Java code, or executing OS commands, when certain XSLT functions are not disabled.

Another case is when the application deserializes XML documents from untrusted sources (e.g. in a REST framework), and if an attacker can provide the XML document to be deserialized, he/she may be able to execute arbitrary code on the server, including opening a reverse shell to launch commands.

XML Injection (CWE-91) coverage by Kiuwan

In Kiuwan, you can search rules covering XML-Injection (CWE-90) filtering by

- Vulnerability Type = **Injection**, and/or
- CWE tag = **CWE:91**

Kiuwan incorporates next rules for XML-Injection (CWE-91) for the following languages.

To obtain detailed information on functionality, coverage, parameterization, remediation, example codes, etc., follow the same steps as described in SQL Injection.

Language	Rule code	CWE
C#	**OPT.CSHARP.JSONInjection**	91
	OPT.CSHARP.XMLInjection	91
Java	**OPT.JAVA.SEC_JAVA.XsltInjection**	91
Objective-C	**OPT.OBJECTIVEC.JSONInjection**	91,345
PHP	**OPT.PHP.SEC.XsltInjection**	91
Python	**OPT.PYTHON.SECURITY.XmlInjection**	91
Swift	**OPT.SWIFT.SECURITY.XMLInjection**	91