TypeScript static code analysis

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

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

libxmljs module:

```
const libxmljs = require("libxmljs");
var fs = require('fs');

var xml = fs.readFileSync('xxe.xml', 'utf8');

var xmlDoc = libxmljs.parseXmlString(xml, { noblanks: true,
```

Compliant Solution

libxmljs module:

```
const libxmljs = require("libxmljs");
var fs = require('fs');

var xml = fs.readFileSync('xxe.xml', 'utf8');

var xmlDoc = libxmljs.parseXmlString(xml); // Compliant: noe
```

See

- OWASP Top 10 2021 Category A5 - Security Misconfiguration
- 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:

cross-origin communications

🔓 Vulnerability

Regular expressions should not be vulnerable to Denial of Service attacks

🔓 Vulnerability

File uploads should be restricted

🔓 Vulnerability

Regular expressions should be syntactically valid

Bug

Types without members, 'any' and 'never' should not be used in type

sonarlint | sonarcloud | sonarqube