rules.sonarsource.com

C static code analysis: XML parsers should not be vulnerable to XXE attacks

4-5 minutes

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

Xerces XercesDOMParser library:

```
#include "xercesc/parsers/XercesDOMParser.hpp"

XercesDOMParser *DOMparser = new XercesDOMParser();
// no entity reference node will be created so the entities will be expanded
DOMparser->setCreateEntityReferenceNodes(false); // Noncompliant
DOMparser->setDisableDefaultEntityResolution(false); // Noncompliant

DOMparser->parse(xmlFile);
```

Xerces SAX2XMLReader library:

```
#include "xercesc/sax2/SAX2XMLReader.hpp"
SAX2XMLReader* reader = XMLReaderFactory::createXMLReader(); // Noncompliant: by default entities resolution is enabled so SAX2XMLReader is not safe
reader->setFeature(XMLUni::fgXercesDisableDefaultEntityResolution, false); // Noncompliant: enable resolution of entities explicitly

reader->parse(xmlFile);
```

Xerces SAXParser library:

```
#include "xercesc/parsers/SAXParser.hpp"

SAXParser* SAXparser = new SAXParser(); // Noncompliant: by default entities resolution is enabled so SAXParser is not safe
SAXparser->setDisableDefaultEntityResolution(false); //
```

Noncompliant: enable resolution of entities explicitly

SAXparser->parse(xmlFile);

LibXML2 library:

#include "libxml/parser.h"

xmlDocPtr doc = xmlReadFile(xmlFile, nullptr, XML_PARSE_DTDLOAD | XML_PARSE_NOENT); // Noncompliant

Compliant Solution

Xerces XercesDOMParser library:

#include "xercesc/parsers/XercesDOMParser.hpp"

XercesDOMParser *DOMparser = new XercesDOMParser(); // by default XercesDOMParser is safe
DOMparser->setCreateEntityReferenceNodes(true); // Compliant: explicitly make the parser safe to XXE vulnerability
DOMparser->setDisableDefaultEntityResolution(true); // Compliant

DOMparser->parse(xmlFile);

Xerces SAX2XMLReader library:

#include "xercesc/sax2/SAX2XMLReader.hpp"

SAX2XMLReader* reader = XMLReaderFactory::createXMLReader();
reader->setFeature(XMLUni::fgXercesDisableDefaultEntityResolution, true); // Compliant

```
reader->parse(xmlFile);
```

Xerces SAXParser library:

```
#include "xercesc/parsers/SAXParser.hpp"

SAXParser* SAXparser = new SAXParser();
SAXparser->setDisableDefaultEntityResolution(true); // Compliant

SAXparser->parse(xmlFile);
```

LibXML2 library:

```
#include "libxml/parser.h"

xmlDocPtr doc = xmlReadFile(xmlFile, nullptr, 0); // Compliant: safe by default since version 2.9
```

See

- OWASP Top 10 2021 Category A5 - Security Misconfiguration
- OWASP Top 10 2017 Category A4 - XML External Entities (XXE)
- OWASP XXE Prevention Cheat Sheet for Xerces
- OWASP XXE Prevention Cheat Sheet for LibXML2
- MITRE, CWE-611 - Information Exposure Through XML External Entity Reference
- MITRE, CWE-827 - Improper Control of Document Type Definition