**XML External Entity (XXE) Injection Development Mitigation SOP**

An XML External Entity attack is a type of attack against an application that parses XML input. This attack occurs when XML input containing a reference to an external entity is processed by a weakly configured XML parser. This attack may lead to the disclosure of confidential data, denial of service, server side request forgery, port scanning from the perspective of the machine where the parser is located, and other system impacts.

Attacks can include disclosing local files, which may contain sensitive data such as passwords or private user data, using file: schemes or relative paths in the system identifier. Since the attack occurs relative to the application processing the XML document, an attacker may use this trusted application to pivot to other internal systems, possibly disclosing other internal content via http(s) requests or launching a CSRF attack to any unprotected internal services. Other attacks can access local resources that may not stop returning data, possibly impacting application availability if too many threads or processes are not released.

**Classic Example**

//need to add example here

The likelihood of this occurring is very low due to the complex nature of the system, so denial of service is the more likely attack, however, there already exists a mitigation strategy in place and applied to our system.

**Example**

Document document;

try {

DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

dbf = DocumentBuilderFactory.newInstance().newDocumentBuilder().newDocument();

} catch(Exception e) {

Throw new RuntimeException(e);

}

The document was initialized without configuring any proper security features.

**Defense Against XXE Injection**

Both DocumentBuilderFactory and SAXParserFactory XML Parsers can be configured using the same techniques to protect them against XML External Entity Injection.  The JAXP DocumentBuilderFactory setFeature method allows a developer to control which implementation-specific XML processor features are enabled or disabled. The features can either be set on the factory or the underlying XMLReader setFeature method.

There are additional configurations that may be necessary when protecting against other parsers and the OWASP XXE Injection Prevention Cheat Sheet, listed in the resources below.

For example, the best defense is to disallow all doctype declarations using the disallow-doctype-decl feature. If doctypes cannot be completely disables, other features can be set to disallow external general entities or external doctype declarations.

When protecting against a TransformerFactory, it is recommended to set the XMLConstants, access\_external\_dtd and access\_external\_stylesheet. Other parsers OWASP shows how to protect include XMLInputFactory, Validator, SchemaFactory, SAXTransformerFactory, XMLReader, and SAXReader, and more.

**Defense Example *(fixing Classic example above)***

//need to add example here

**Specific Explanation**

The instance of DocumentBuilderFactory, in the example below, disables the features that make the application vulnerable to this type of attack.

**Recommendation**

Document document;

try {

DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();

dbf.setFeature(EXTERNAL\_GENERAL\_ENTITIES\_FEATURE, false);

dbf.setFeature(EXTERNAL\_PARAMETER\_ENTITIES\_FEATURE, false);

document = dbf.newDocumentBuilder().newDocument();

} catch(Exception e) {

Throw new RuntimeException(e);

}

**References**

1. [OWASP – XML External Entity Injection](about:blank)