Security Guide of WeChat Cross-Border Pay:

Dear WeChat Cross-Border Pay merchants:
Please confirm whether you have specific settings in your system to prevent XML External Entity Injection vulnerability when parsing xml information of WeChat Pay notification. If not, please fix the vulnerability by the following guide as soon as possible.
Note: SDK for In-APP payment is not affected.

Fix Suggestion
If the SDK provided on WeChat website is integrated in your system, please update it to the newest version: https://pay.weixin.qq.com/wiki/doc/api/jsapi.php?chapter=11_1. If not, please check your own parsing logic by the following guide.
To prevent XXE vulnerability, there are specific settings required in your code, please refer to the following samples for different development environments:
【PHP】

```
libxml_disable_entity_loader(true);
```

【JAVA】

```
import javax.xml.parsers.DocumentBuilderFactory;
import javax.xml.parsers.ParserConfigurationException; // catching unsupported features
...
DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
String FEATURE = null;
try {
// This is the PRIMARY defense. If DTDs (doctypes) are disallowed, almost all XML entity attacks are prevented
// Xerces 2 only - http://xerces.apache.org/xerces2-j/features.html#disallow-doctype-decl
FEATURE = "http://apache.org/xml/features/disallow-doctype-decl";
dbf.setFeature(FEATURE, true);

// If you can't completely disable DTDs, then at least do the following:
// Xerces 1 - http://xerces.apache.org/xerces-j/features.html#external-general-entities
// Xerces 2 - http://xerces.apache.org/xerces2-j/features.html#external-general-entities
// JDK7+ - http://xml.org/sax/features/external-general-entities
FEATURE = "http://xml.org/sax/features/external-general-entities";
dbf.setFeature(FEATURE, false);

// Xerces 1 - http://xerces.apache.org/xerces-j/features.html#external-parameter-entities
// Xerces 2 - http://xerces.apache.org/xerces2-j/features.html#external-parameter-entities
// JDK7+ - http://xml.org/sax/features/external-parameter-entities
FEATURE = "http://xml.org/sax/features/external-parameter-entities";
dbf.setFeature(FEATURE, false);

// Disable external DTDs as well
```

```
FEATURE = "http://apache.org/xml/features/nonvalidating/load-external-dtd";
dbf.setFeature(FEATURE, false);

// and these as well, per Timothy Morgan's 2014 paper: "XML Schema, DTD, and Entity Attacks"
dbf.setXIncludeAware(false);
dbf.setExpandEntityReferences(false);

// And, per Timothy Morgan: "If for some reason support for inline DOCTYPEs are a requirement, then
// ensure the entity settings are disabled (as shown above) and beware that SSRF attacks
// (http://cwe.mitre.org/data/definitions/918.html) and denial
// of service attacks (such as billion laughs or decompression bombs via "jar:") are a risk."

// remaining parser logic
...
} catch (ParserConfigurationException e) {
// This should catch a failed setFeature feature
logger.info("ParserConfigurationException was thrown. The feature '" +
FEATURE + "' is probably not supported by your XML processor.");
...
}
catch (SAXException e) {
// On Apache, this should be thrown when disallowing DOCTYPE
logger.warning("A DOCTYPE was passed into the XML document");
...
}
catch (IOException e) {
// XXE that points to a file that doesn't exist
logger.error("IOException occurred, XXE may still possible: " + e.getMessage());
...
}
DocumentBuilder safebuilder = dbf.newDocumentBuilder();
```

【.Net】

```
XmlResolver = null;
```

【Python】

```
from lxml import etree
xmlData = etree.parse(xmlSource,etree.XMLParser(resolve_entities=False))
```

【c/c++(The common library is libxml2 libxerces-c)】

【libxml2】:

Please confirm that the following configuration item is closed:

XML_PARSE_NOENT 和 XML_PARSE_DTDLOAD
xxe vulnerability is resolved on versions higher than 2.9

【libxerces-c】:
If XercesDOMParser is used:
XercesDOMParser *parser = new XercesDOMParser;
parser->setCreateEntityReferenceNodes(false);

If SAXParser is used:
SAXParser* parser = new SAXParser;
parser->setDisableDefaultEntityResolution(true);

If SAX2XMLReader is used:
SAX2XMLReader* reader = XMLReaderFactory::createXMLReader();
parser->setFeature(XMLUni::fgXercesDisableDefaultEntityResolution, true);

For more vulnerability fix suggestions:
https://www.owasp.org/index.php/XML_External_Entity_(XXE)_Prevention_Cheat_Sheet#C.2FC.2B.2B