Fortify Security Report

Jun 17, 2015

dkwakkel

Executive Summary

Issues Overview

On Jun 2, 2015, a source code review was performed over the poi code base. 3,213 files, 178,513 LOC (Executable) were scanned and reviewed for defects that could lead to potential security vulnerabilities. A total of 3319 reviewed findings were uncovered during the analysis.

Issues by Fortify Priority Order

Refined by: category:"xml entity expansion injection" OR category:"xml external entity injection"

Low	22
High	10

Recommendations and Conclusions

The Issues Category section provides Fortify recommendations for addressing issues at a generic level. The recommendations for specific fixes can be extrapolated from those generic recommendations by the development group.

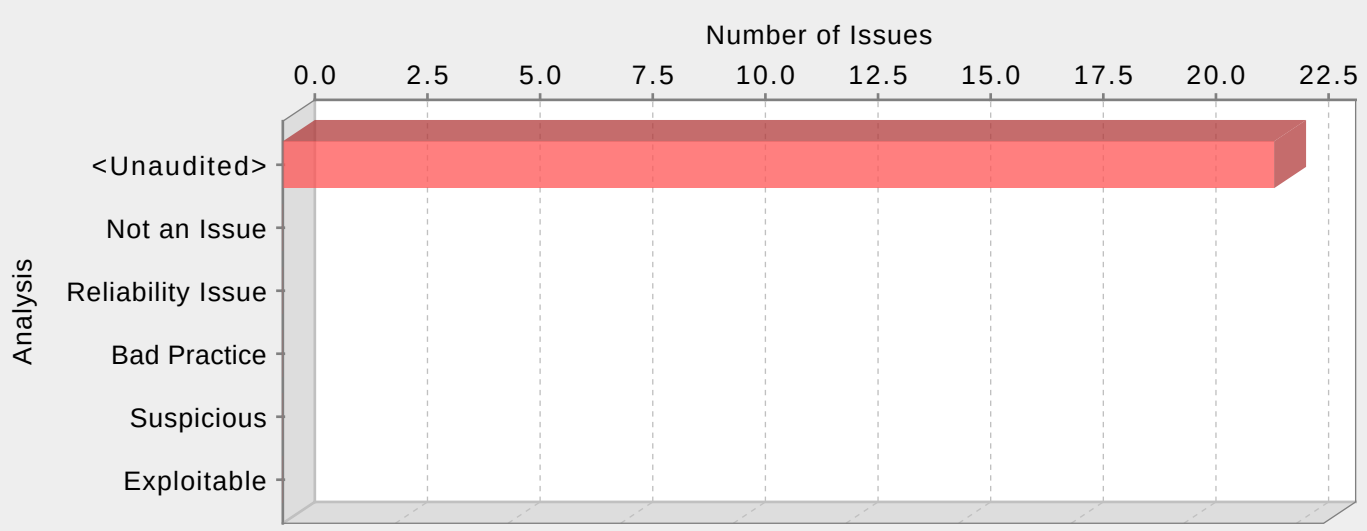
Results Outline

Overall number of results

The scan found 3319 issues.

Vulnerability Examples by Category

Category: XML Entity Expansion Injection (22 Issues)



Abstract:

Using XML parsers configured to not prevent nor limit Document Type Definition (DTD) entity resolution can expose the parser to an XML Entity Expansion injection

Explanation:

XML Entity Expansion injection also known as XML Bombs are DoS attacks that benefit from valid and well-formed XML blocks that expand exponentially until they exhaust the server allocated resources. XML allows to define custom entities which act as string substitution macros. By nesting recurrent entity resolutions, an attacker can easily crash the server resources.

The following XML document shows an example of an XML Bomb.

```
<?xml version="1.0"?>
<!DOCTYPE lolz [
<!ENTITY lol "lol">
<!ENTITY lol2 "&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;&lol;">
<!ENTITY lol3 "&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;&lol2;">
<!ENTITY lol4 "&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;&lol3;">
<!ENTITY lol5 "&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;&lol4;">
<!ENTITY lol6 "&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;&lol5;">
<!ENTITY lol7 "&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;&lol6;">
<!ENTITY lol8 "&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;&lol7;">
<!ENTITY lol9 "&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;&lol8;">
]>
<lolz>&lol9;</lolz>
```

This test could crash the server by expanding the small XML document into more than 3GB in memory.

Recommendations:

An XML parser should be configured securely so that it does not allow document type definition (DTD) custom entities as part of an incoming XML document.

To avoid XML Entity Expansion injection the "secure-processing" property should be set for an XML factory, parser or reader:

```
factory.setFeature("http://javax.xml.XMLConstants/feature/secure-processing", true);
```

In JAXP 1.3 and earlier versions, when the secure processing feature is on, default limitations are set for DOM and SAX parsers. These limits are:

entityExpansionLimit = 64,000
elementAttributeLimit = 10,000

Since JAXP 1.4, the secure processing feature is turned on by default. In addition to the above limits, a new maxOccur limit is added to the validating parser. The limit is:

maxOccur = 5,000

If inline DOCTYPE declaration is not needed, it can be completely disabled with the following property:

factory.setFeature("http://apache.org/xml/features/disallow-doctype-decl", true);

ExcelFileFormatDocFunctionExtractor.java, line 450 (XML Entity Expansion Injection)

Fortify Priority:	Low	**Folder**	Low
Kingdom:	Input Validation and Representation		
Abstract:	The XML parser configured in ExcelFileFormatDocFunctionExtractor.java:450 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.		
Sink:	ExcelFileFormatDocFunctionExtractor.java:450 xr.parse(...) : XML document parsed allowing external entity resolution()		

```
448
449                 try {
450                  xr.parse(inSrc);
451                  is.close();
452                 } catch (IOException e) {
```

PkiTestUtils.java, line 221 (XML Entity Expansion Injection)

Fortify Priority:	Low	**Folder**	Low
Kingdom:	Input Validation and Representation		
Abstract:	The XML parser configured in PkiTestUtils.java:221 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.		
Sink:	PkiTestUtils.java:221 documentBuilder.parse(...) : XML document parsed allowing external entity resolution()		

```
219                         DocumentBuilder documentBuilder = documentBuilderFactory
220                                 .newDocumentBuilder();
221                         Document document = documentBuilder.parse(inputSource);
222                         return document;
223                   }
```

TestXSSFExportToXML.java, line 496 (XML Entity Expansion Injection)

Fortify Priority:	Low	**Folder**	Low
Kingdom:	Input Validation and Representation		
Abstract:	The XML parser configured in TestXSSFExportToXML.java:496 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.		
Sink:	TestXSSFExportToXML.java:496 docBuilder.parse(...) : XML document parsed allowing external entity resolution()		

```
494                     docBuilder.setEntityResolver(new DummyEntityResolver());
495
496                     docBuilder.parse(new ByteArrayInputStream(xmlData.getBytes("UTF-8")));
497                 }
```

TestWordToConverterSuite.java, line 169 (XML Entity Expansion Injection)

Fortify Priority:	Low	**Folder**	Low
Kingdom:	Input Validation and Representation		
Abstract:	The XML parser configured in TestWordToConverterSuite.java:169 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.		
Sink:	TestWordToConverterSuite.java:169 newInstance().newTransformer() : XML document parsed allowing external entity resolution()		

```
167
168                     Transformer transformer = TransformerFactory.newInstance()
```

```
169                             .newTransformer();
170                     transformer.setOutputProperty( OutputKeys.ENCODING, "utf-8" );
171                     transformer.setOutputProperty( OutputKeys.INDENT, "yes" );
```

WordToFoConverter.java, line 95 (XML Entity Expansion Injection)

Fortify Priority:	Low	**Folder**	Low
Kingdom:	Input Validation and Representation		

Abstract: The XML parser configured in WordToFoConverter.java:95 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.

Sink: WordToFoConverter.java:95 tf.newTransformer() : XML document parsed allowing external entity resolution()

```
93                      StreamResult streamResult = new StreamResult( out );
94                      TransformerFactory tf = TransformerFactory.newInstance();
95                      Transformer serializer = tf.newTransformer();
96                      // TODO set encoding from a command argument
97                      serializer.setOutputProperty( OutputKeys.ENCODING, "UTF-8" );
```

ExcelToFoConverter.java, line 100 (XML Entity Expansion Injection)

Fortify Priority:	Low	**Folder**	Low
Kingdom:	Input Validation and Representation		

Abstract: The XML parser configured in ExcelToFoConverter.java:100 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.

Sink: ExcelToFoConverter.java:100 tf.newTransformer() : XML document parsed allowing external entity resolution()

```
98
99                      TransformerFactory tf = TransformerFactory.newInstance();
100                     Transformer serializer = tf.newTransformer();
101                     // TODO set encoding from a command argument
102                     serializer.setOutputProperty( OutputKeys.ENCODING, "UTF-8" );
```

ExcelToHtmlConverter.java, line 93 (XML Entity Expansion Injection)

Fortify Priority:	Low	**Folder**	Low
Kingdom:	Input Validation and Representation		

Abstract: The XML parser configured in ExcelToHtmlConverter.java:93 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.

Sink: ExcelToHtmlConverter.java:93 tf.newTransformer() : XML document parsed allowing external entity resolution()

```
91
92                      TransformerFactory tf = TransformerFactory.newInstance();
93                      Transformer serializer = tf.newTransformer();
94                      // TODO set encoding from a command argument
95                      serializer.setOutputProperty( OutputKeys.ENCODING, "UTF-8" );
```

TestWordToFoConverter.java, line 52 (XML Entity Expansion Injection)

Fortify Priority:	Low	**Folder**	Low
Kingdom:	Input Validation and Representation		

Abstract: The XML parser configured in TestWordToFoConverter.java:52 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.

Sink: TestWordToFoConverter.java:52 newInstance().newTransformer() : XML document parsed allowing external entity resolution()

```
50
51                      Transformer transformer = TransformerFactory.newInstance()
52                              .newTransformer();
53                      transformer.setOutputProperty( OutputKeys.INDENT, "yes" );
54                      transformer.transform(
```

TestWordToConverterSuite.java, line 139 (XML Entity Expansion Injection)			
Fortify Priority:	Low	Folder	Low
Kingdom:	Input Validation and Representation		
Abstract:	The XML parser configured in TestWordToConverterSuite.java:139 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.		
Sink:	TestWordToConverterSuite.java:139 newInstance().newTransformer() : XML document parsed allowing external entity resolution()		

```
137
138                     Transformer transformer = TransformerFactory.newInstance()
139                             .newTransformer();
140                     transformer.setOutputProperty( OutputKeys.ENCODING, "utf-8" );
141                     transformer.setOutputProperty( OutputKeys.INDENT, "false" );
```

XSSFExportToXml.java, line 232 (XML Entity Expansion Injection)			
Fortify Priority:	Low	Folder	Low
Kingdom:	Input Validation and Representation		
Abstract:	The XML parser configured in XSSFExportToXml.java:232 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.		
Sink:	XSSFExportToXml.java:232 transfac.newTransformer() : XML document parsed allowing external entity resolution()		

```
230                 //set up a transformer
231                 TransformerFactory transfac = TransformerFactory.newInstance();
232                 Transformer trans = transfac.newTransformer();
233                 trans.setOutputProperty(OutputKeys.OMIT_XML_DECLARATION, "yes");
234                 trans.setOutputProperty(OutputKeys.INDENT, "yes");
```

OOXMLPrettyPrint.java, line 126 (XML Entity Expansion Injection)			
Fortify Priority:	Low	Folder	Low
Kingdom:	Input Validation and Representation		
Abstract:	The XML parser configured in OOXMLPrettyPrint.java:126 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.		
Sink:	OOXMLPrettyPrint.java:126 transformerFactory.newTransformer() : XML document parsed allowing external entity resolution()		

```
124     private static void pretty(Document document, OutputStream outputStream, int
indent) throws TransformerException {
125         TransformerFactory transformerFactory = TransformerFactory.newInstance();
126         Transformer transformer = transformerFactory.newTransformer();
127         transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
128         if (indent > 0) {
```

TestWordToConverterSuite.java, line 110 (XML Entity Expansion Injection)			
Fortify Priority:	Low	Folder	Low
Kingdom:	Input Validation and Representation		
Abstract:	The XML parser configured in TestWordToConverterSuite.java:110 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.		
Sink:	TestWordToConverterSuite.java:110 newInstance().newTransformer() : XML document parsed allowing external entity resolution()		

```
108
109                     Transformer transformer = TransformerFactory.newInstance()
110                             .newTransformer();
111                     transformer.setOutputProperty( OutputKeys.ENCODING, "utf-8" );
112                     transformer.setOutputProperty( OutputKeys.INDENT, "false" );
```

WordToTextConverter.java, line 196 (XML Entity Expansion Injection)

Fortify Priority:	Low	Folder	Low
Kingdom:	Input Validation and Representation		
Abstract:	The XML parser configured in WordToTextConverter.java:196 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.		
Sink:	WordToTextConverter.java:196 tf.newTransformer() : XML document parsed allowing external entity resolution()		

```
194
195                 TransformerFactory tf = TransformerFactory.newInstance();
196                 Transformer serializer = tf.newTransformer();
197                 // TODO set encoding from a command argument
198                 serializer.setOutputProperty( OutputKeys.ENCODING, "UTF-8" );
```

WordToHtmlConverter.java, line 142 (XML Entity Expansion Injection)

Fortify Priority:	Low	Folder	Low
Kingdom:	Input Validation and Representation		
Abstract:	The XML parser configured in WordToHtmlConverter.java:142 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.		
Sink:	WordToHtmlConverter.java:142 tf.newTransformer() : XML document parsed allowing external entity resolution()		

```
140
141                     TransformerFactory tf = TransformerFactory.newInstance();
142                     Transformer serializer = tf.newTransformer();
143                     // TODO set encoding from a command argument
144                     serializer.setOutputProperty( OutputKeys.ENCODING, "UTF-8" );
```

TestExcelConverterSuite.java, line 105 (XML Entity Expansion Injection)

Fortify Priority:	Low	Folder	Low
Kingdom:	Input Validation and Representation		
Abstract:	The XML parser configured in TestExcelConverterSuite.java:105 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.		
Sink:	TestExcelConverterSuite.java:105 newInstance().newTransformer() : XML document parsed allowing external entity resolution()		

```
103
104                 Transformer transformer = TransformerFactory.newInstance()
105                         .newTransformer();
106                 transformer.setOutputProperty( OutputKeys.ENCODING, "utf-8" );
107                 transformer.setOutputProperty( OutputKeys.INDENT, "yes" );
```

TestWordToHtmlConverter.java, line 73 (XML Entity Expansion Injection)

Fortify Priority:	Low	Folder	Low
Kingdom:	Input Validation and Representation		
Abstract:	The XML parser configured in TestWordToHtmlConverter.java:73 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.		
Sink:	TestWordToHtmlConverter.java:73 newInstance().newTransformer() : XML document parsed allowing external entity resolution()		

```
71
72                  Transformer transformer = TransformerFactory.newInstance()
73                          .newTransformer();
74                  transformer.setOutputProperty( OutputKeys.INDENT, "yes");
75                  transformer.setOutputProperty( OutputKeys.ENCODING, "utf-8");
```

WordToTextConverter.java, line 114 (XML Entity Expansion Injection)

Fortify Priority:	Low	Folder	Low
Kingdom:	Input Validation and Representation		

Abstract:	The XML parser configured in WordToTextConverter.java:114 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.
Sink:	WordToTextConverter.java:114 tf.newTransformer() : XML document parsed allowing external entity resolution()

```
112
113                     TransformerFactory tf = TransformerFactory.newInstance();
114                     Transformer serializer = tf.newTransformer();
115                     // TODO set encoding from a command argument
116                     serializer.setOutputProperty( OutputKeys.ENCODING, "UTF-8" );
```

PkiTestUtils.java, line 231 (XML Entity Expansion Injection)

Fortify Priority:	Low	**Folder**	Low
Kingdom:	Input Validation and Representation		
Abstract:	The XML parser configured in PkiTestUtils.java:231 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.		
Sink:	PkiTestUtils.java:231 transformerFactory.newTransformer() : XML document parsed allowing external entity resolution()		

```
229             TransformerFactory transformerFactory = TransformerFactory
230                     .newInstance();
231             Transformer transformer = transformerFactory.newTransformer();
232             /*
233              * We have to omit the ?xml declaration if we want to embed the
```

XLSX2CSV.java, line 371 (XML Entity Expansion Injection)

Fortify Priority:	Low	**Folder**	Low
Kingdom:	Input Validation and Representation		
Abstract:	The XML parser configured in XLSX2CSV.java:371 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.		
Sink:	XLSX2CSV.java:371 sheetParser.parse(...) : XML document parsed allowing external entity resolution()		

```
369             ContentHandler handler = new MyXSSFSheetHandler(styles, strings,
this.minColumns, this.output);
370             sheetParser.setContentHandler(handler);
371             sheetParser.parse(sheetSource);
372         }
```

RecordGenerator.java, line 82 (XML Entity Expansion Injection)

Fortify Priority:	Low	**Folder**	Low
Kingdom:	Input Validation and Representation		
Abstract:	The XML parser configured in RecordGenerator.java:82 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.		
Sink:	RecordGenerator.java:82 builder.parse(...) : XML document parsed allowing external entity resolution()		

```
80                  DocumentBuilderFactory factory =
XMLHelper.getDocumentBuilderFactory();
81                  DocumentBuilder builder = factory.newDocumentBuilder();
82                  Document document = builder.parse(file);
83                  Element record = document.getDocumentElement();
84                  String extendstg =
record.getElementsByTagName("extends").item(0).getFirstChild().getNodeValue();
```

RecordGenerator.java, line 135 (XML Entity Expansion Injection)

Fortify Priority:	Low	**Folder**	Low
Kingdom:	Input Validation and Representation		

Abstract:	The XML parser configured in RecordGenerator.java:135 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.
Sink:	RecordGenerator.java:135 tf.newTransformer(...) : XML document parsed allowing external entity resolution()

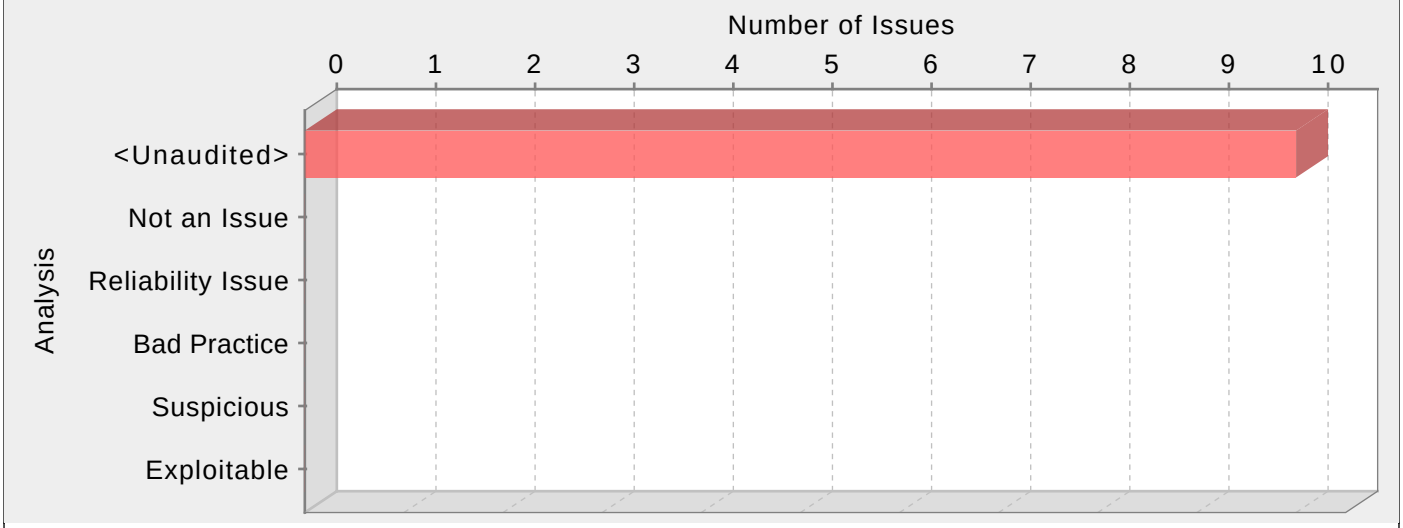
```
133                 try
134                 {
135                     t = tf.newTransformer(ss);
136                 }
137                 catch (TransformerException ex)
```

TestExcelConverterSuite.java, line 134 (XML Entity Expansion Injection)

Fortify Priority:	Low	**Folder**	Low
Kingdom:	Input Validation and Representation		
Abstract:	The XML parser configured in TestExcelConverterSuite.java:134 does not prevent nor limit Document Type Definition (DTD) entity resolution. This can expose the parser to an XML Entity Expansion injection.		
Sink:	TestExcelConverterSuite.java:134 newInstance().newTransformer() : XML document parsed allowing external entity resolution()		

```
132
133                 Transformer transformer = TransformerFactory.newInstance()
134                         .newTransformer();
135                 transformer.setOutputProperty( OutputKeys.ENCODING, "utf-8" );
136                 transformer.setOutputProperty( OutputKeys.INDENT, "no" );
```

Category: XML External Entity Injection (10 Issues)



Abstract:

Using XML parsers configured to not prevent nor limit external entities resolution can expose the parser to an XML External Entities attack

Explanation:

XML External Entities attacks benefit from an XML feature to build documents dynamically at the time of processing. An XML entity allows inclusion of data dynamically from a given resource. External entities allow an XML document to include data from an external URI. Unless configured to do otherwise, external entities force the XML parser to access the resource specified by the URI, e.g., a file on the local machine or on a remote system. This behavior exposes the application to XML External Entity (XXE) attacks, which can be used to perform denial of service of the local system, gain unauthorized access to files on the local machine, scan remote machines, and perform denial of service of remote systems.

The following XML document shows an example of an XXE attack.

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE foo [
<!ELEMENT foo ANY >
<!ENTITY xxe SYSTEM "file:///dev/random" >]><foo>&xxe;</foo>
```

This example could crash the server (on a UNIX system), if the XML parser attempts to substitute the entity with the contents of the /dev/random file.

Recommendations:

An XML parser should be configured securely so that it does not allow external entities as part of an incoming XML document.

To avoid XXE injections the following properties should be set for an XML factory, parser or reader:

```
factory.setFeature("http://xml.org/sax/features/external-general-entities", false);
factory.setFeature("http://xml.org/sax/features/external-parameter-entities", false);
```

If inline DOCTYPE declaration is not needed, it can be completely disabled with the following property:

```
factory.setFeature("http://apache.org/xml/features/disallow-doctype-decl", true);
```

RecordGenerator.java, line 82 (XML External Entity Injection)

Fortify Priority:	High	**Folder**	High
Kingdom:	Input Validation and Representation		
Abstract:	XML parser configured in RecordGenerator.java:82 does not prevent nor limit external entities resolution. This can expose the parser to an XML External Entities attack.		
Sink:	RecordGenerator.java:82 builder.parse(...) : XML document parsed allowing external entity resolution()		

```
80                  DocumentBuilderFactory factory =
XMLHelper.getDocumentBuilderFactory();
81                  DocumentBuilder builder = factory.newDocumentBuilder();
82                  Document document = builder.parse(file);
83                  Element record = document.getDocumentElement();
```

```
84                          String extendstg =
record.getElementsByTagName("extends").item(0).getFirstChild().getNodeValue();
```

OOXMLPrettyPrint.java, line 126 (XML External Entity Injection)

Fortify Priority:	High	**Folder**	High
Kingdom:	Input Validation and Representation		

Abstract: XML parser configured in OOXMLPrettyPrint.java:126 does not prevent nor limit external entities resolution. This can expose the parser to an XML External Entities attack.

Sink: OOXMLPrettyPrint.java:126 transformerFactory.newTransformer() : XML document parsed allowing external entity resolution()

```
124     private static void pretty(Document document, OutputStream outputStream, int
indent) throws TransformerException {
125         TransformerFactory transformerFactory = TransformerFactory.newInstance();
126         Transformer transformer = transformerFactory.newTransformer();
127         transformer.setOutputProperty(OutputKeys.ENCODING, "UTF-8");
128         if (indent > 0) {
```

WordToTextConverter.java, line 114 (XML External Entity Injection)

Fortify Priority:	High	**Folder**	High
Kingdom:	Input Validation and Representation		

Abstract: XML parser configured in WordToTextConverter.java:114 does not prevent nor limit external entities resolution. This can expose the parser to an XML External Entities attack.

Sink: WordToTextConverter.java:114 tf.newTransformer() : XML document parsed allowing external entity resolution()

```
112
113                 TransformerFactory tf = TransformerFactory.newInstance();
114                 Transformer serializer = tf.newTransformer();
115                 // TODO set encoding from a command argument
116                 serializer.setOutputProperty( OutputKeys.ENCODING, "UTF-8" );
```

WordToHtmlConverter.java, line 142 (XML External Entity Injection)

Fortify Priority:	High	**Folder**	High
Kingdom:	Input Validation and Representation		

Abstract: XML parser configured in WordToHtmlConverter.java:142 does not prevent nor limit external entities resolution. This can expose the parser to an XML External Entities attack.

Sink: WordToHtmlConverter.java:142 tf.newTransformer() : XML document parsed allowing external entity resolution()

```
140
141                 TransformerFactory tf = TransformerFactory.newInstance();
142                 Transformer serializer = tf.newTransformer();
143                 // TODO set encoding from a command argument
144                 serializer.setOutputProperty( OutputKeys.ENCODING, "UTF-8" );
```

WordToFoConverter.java, line 95 (XML External Entity Injection)

Fortify Priority:	High	**Folder**	High
Kingdom:	Input Validation and Representation		

Abstract: XML parser configured in WordToFoConverter.java:95 does not prevent nor limit external entities resolution. This can expose the parser to an XML External Entities attack.

Sink: WordToFoConverter.java:95 tf.newTransformer() : XML document parsed allowing external entity resolution()

```
93                  StreamResult streamResult = new StreamResult( out );
94                  TransformerFactory tf = TransformerFactory.newInstance();
95                  Transformer serializer = tf.newTransformer();
96                  // TODO set encoding from a command argument
97                  serializer.setOutputProperty( OutputKeys.ENCODING, "UTF-8" );
```

XSSFExportToXml.java, line 232 (XML External Entity Injection)			
Fortify Priority:	High	Folder	High
Kingdom:	Input Validation and Representation		
Abstract:	XML parser configured in XSSFExportToXml.java:232 does not prevent nor limit external entities resolution. This can expose the parser to an XML External Entities attack.		
Sink:	XSSFExportToXml.java:232 transfac.newTransformer() : XML document parsed allowing external entity resolution()		

```
230                 //set up a transformer
231                 TransformerFactory transfac = TransformerFactory.newInstance();
232                 Transformer trans = transfac.newTransformer();
233                 trans.setOutputProperty(OutputKeys.OMIT_XML_DECLARATION, "yes");
234                 trans.setOutputProperty(OutputKeys.INDENT, "yes");
```

ExcelToFoConverter.java, line 100 (XML External Entity Injection)			
Fortify Priority:	High	Folder	High
Kingdom:	Input Validation and Representation		
Abstract:	XML parser configured in ExcelToFoConverter.java:100 does not prevent nor limit external entities resolution. This can expose the parser to an XML External Entities attack.		
Sink:	ExcelToFoConverter.java:100 tf.newTransformer() : XML document parsed allowing external entity resolution()		

```
98
99                  TransformerFactory tf = TransformerFactory.newInstance();
100                 Transformer serializer = tf.newTransformer();
101                 // TODO set encoding from a command argument
102                 serializer.setOutputProperty( OutputKeys.ENCODING, "UTF-8" );
```

RecordGenerator.java, line 135 (XML External Entity Injection)			
Fortify Priority:	High	Folder	High
Kingdom:	Input Validation and Representation		
Abstract:	XML parser configured in RecordGenerator.java:135 does not prevent nor limit external entities resolution. This can expose the parser to an XML External Entities attack.		
Sink:	RecordGenerator.java:135 tf.newTransformer(...) : XML document parsed allowing external entity resolution()		

```
133             try
134             {
135                 t = tf.newTransformer(ss);
136             }
137             catch (TransformerException ex)
```

ExcelToHtmlConverter.java, line 93 (XML External Entity Injection)			
Fortify Priority:	High	Folder	High
Kingdom:	Input Validation and Representation		
Abstract:	XML parser configured in ExcelToHtmlConverter.java:93 does not prevent nor limit external entities resolution. This can expose the parser to an XML External Entities attack.		
Sink:	ExcelToHtmlConverter.java:93 tf.newTransformer() : XML document parsed allowing external entity resolution()		

```
91
92                  TransformerFactory tf = TransformerFactory.newInstance();
93                  Transformer serializer = tf.newTransformer();
94                  // TODO set encoding from a command argument
95                  serializer.setOutputProperty( OutputKeys.ENCODING, "UTF-8" );
```

WordToTextConverter.java, line 196 (XML External Entity Injection)			
Fortify Priority:	High	Folder	High

Kingdom:	Input Validation and Representation
Abstract:	XML parser configured in WordToTextConverter.java:196 does not prevent nor limit external entities resolution. This can expose the parser to an XML External Entities attack.
Sink:	WordToTextConverter.java:196 tf.newTransformer() : XML document parsed allowing external entity resolution()

```
194
195                 TransformerFactory tf = TransformerFactory.newInstance();
196                 Transformer serializer = tf.newTransformer();
197                 // TODO set encoding from a command argument
198                 serializer.setOutputProperty( OutputKeys.ENCODING, "UTF-8" );
```