Every Signature is Broken:
On the Insecurity of Microsoft
Office’s OOXML Signatures

A digital signature is an electronic, encrypted, stamp of authentication [...].
A signature confirms that the information originated from the signer and has not been altered.

- Microsoft

RUHR-UNIVERSITÄT BOCHUM

Simon Rohlmann, Vladislav Mladenov, Christian Mainka,
Daniel Hirschberger, Jörg Schwenk
OOXML Structure

signed_document.docx.zip

This Document is protected with a digital signature
This Document is protected with a digital signature

Partial coverage: Only <Relationship> Elements

Full coverage

No Protection
Specification Flaw #1/3: Content Injection Attack

- OOXML Signatures = partial Signatures
- Add unsigned files to show new content
Content Injection Attack

1. Attacker retrieves signed document

2. Attacker manipulates signed document
   • Shows manipulated content
   • Keeps signature valid
Content Injection Attack

XML
document.xml.rels

XML
document.xml

XML
fontTable.xml

XML
styles.xml

XML
people.xml

This Document was manipulated

Partial coverage: Only <Relationship> Elements

Full coverage

No Protection

Christian Mainka | @CheariX | Every Signature is Broken: On the Insecurity of Microsoft Office’s OOXML Signatures
Implementation Flaw #1/2: Universal Signature Forgery

• Extract valid XML Signature from ODF, SAML, ...
• Embed in OOXML
Universal Signature Forgery

This Document was manipulated

Christian Mainka | @CheariX | Every Signature is Broken: On the Insecurity of Microsoft Office’s OOXML Signatures
Evaluation Results

“Every Signature is Broken”
“Every Signature is Broken”

<table>
<thead>
<tr>
<th>Microsoft Office</th>
<th>Build</th>
<th>CIA</th>
<th>Specification Flaws</th>
<th>Implementation Flaws</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>Content Masking</td>
<td>USF</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Attack</td>
<td>Malicious Repair</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Font Inj.</td>
<td>Attack</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Style Inj.</td>
<td>Dup. Doc.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Legacy Wrapping</td>
<td>Evil Type</td>
</tr>
<tr>
<td>Windows</td>
<td>2013 15.0.5423.1000</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>2016 16.0.5278.1000</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>2019 16.0.10386.20017</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>2021 16.0.14332.20303</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>365 16.0.15028.20248</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>macOS</td>
<td>2019 16.61.22050700</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>2021 16.61.22050700</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td>365 16.61.22050700</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>OnlyOffice Desktop</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Windows</td>
<td>7.1.1.57</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>macOS</td>
<td>7.1.1 (533)</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td>Linux</td>
<td>7.1.1.57</td>
<td>✗</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

Legend: **✓**: Not Vulnerable  **✗**: Vulnerable  **○**: Limited Vulnerability

Christian Mainka | @CheariX | Every Signature is Broken: On the Insecurity of Microsoft Office's OOXML Signatures
Conclusion
Conclusion & Lessons Learned

• Major Issues
  • OOXML uses partial signatures
  • Rendering flow involves signed and unsigned data
  • Cryptographic verification is complex for documents

• Content vs Metadata
  • Do not render people.xml, styles.xml, ...

• PoC Files
  • github.com/RUB-NDS/OOXML_Signature_Security

• Details in the Paper