

Oletools

🔗♂ Guide: Analyzing .docx and .xlsm Files with oletools

🔗 0. Installing oletools

```
pip install oletools
```

🔗 Analyzing .docx Files (OpenXML Word Documents)

1. Run olevba on the .docx file

```
olevba file.docx
```

“ ⚠ .docx files usually do **not** contain VBA macros. But it's always good to check.

2. Unzip the .docx to check for embedded objects

```
unzip file.docx -d extracted/
```

3. Look for embedded OLE objects

Check this directory:

```
extracted/word/embeddings/
```

You may find:

```
oleObject1.bin
```

```
oleObject2.bin
```

4. Analyze embedded OLE objects

```
olevba extracted/word/embeddings/oleObject1.bin
```

Repeat for any other `.bin` files.

5. Use `oleid` for file metadata overview

```
oleid file.docx
```

This gives a quick summary of:

- Macros present
- External relationships
- Encryption
- Suspicious flags

☐☐ Analyzing `.xlsm` Files (Macro-Enabled Excel)

1. Run olevba on the file

```
olevba file.xlsm
```

This will:

- Extract all VBA macros
- Show suspicious keywords
- Show auto-executing macros
- Print the actual macro code

2. Look for these red flags

Type	Keyword	Why it matters
AutoExec	Workbook_Open , Auto_Open	Executes macro automatically on open
Suspicious	Shell , CreateObject , WScript.Shell	Can run system commands
IOC	http , .exe , powershell , registry edits	Potential malware payloads
Obfuscation	Chr , Base64 , StrReverse , Split	Used to hide real behavior

3. Extract VBA code to a file (optional)

```
olevba file.xlsm > macro_output.txt
```

4. Use oleid for metadata

```
oleid file.xlsm
```

🔑 Key Indicators to Watch

Category	Example	Why It's Suspicious
AutoExec	Auto_Open , Workbook_Open	Runs automatically

Category	Example	Why It's Suspicious
Command Exec	Shell , powershell , cmd	System command execution
Obfuscation	Chr(72) & Chr(84) , Base64	Payload hiding
Downloader	URLDownloadToFile , XMLHTTP	Drops additional malware
Persistence	Registry writes, Startup	Foothold creation
Encoding	Long strings, encoded blobs	Shellcode or encoded scripts

🔍 Quick Command List (Cheat Sheet)

```
# General analysis
olevba file.xlsm
olevba file.docx

# Extract and analyze embedded objects
unzip file.docx -d extracted/
olevba extracted/word/embeddings/oleObject1.bin

# Inspect file metadata
oleid file.xlsm
oleid file.docx

# Save macro code to a text file
olevba file.xlsm > macros.txt

# Identify embedded objects
ls extracted/word/embeddings/
file extracted/word/embeddings/oleObject1.bin
```

🔍 Pro Tips

- Use `olevba` with caution on unknown files — always analyze in a **VM/sandbox**.
- Combine `olevba` with `strings`, `file`, or `hexdump` for deeper inspection.

- Copy suspicious VBA code for manual review or deobfuscation if needed.
 - Use `oleid` first for a fast overview before diving into macro code.
-

Revision #4

Created 8 September 2024 13:18:29 by Togoboi

Updated 1 April 2025 14:52:50 by Togoboi