Oletools 0.51 Cheat Sheet

Homepage: <https://decalage.info/python/oletools>

Doc: <https://github.com/decalage2/oletools/wiki>

Issues/Questions: <https://github.com/decalage2/oletools/issues>

# Install - Update

Install/Update **latest release version**:

pip install -U oletools

\* On Linux, add “**sudo -H**” before pip.

Note: this will automatically create **shortcuts** to run oletools from any folder: olevba, mraptor, oleid, etc

Install/Update **latest development version**:

pip install -U https://github.com/decalage2/oletools/archive/master.zip

More options: <https://github.com/decalage2/oletools/wiki/Install>

# Common Options

Options common to several oletools:

|  |  |
| --- | --- |
| -r | find files recursively in subdirectories |
| -z <password> | Open a password-protected zip file  Ex: -z infected |
| -f <filespec> | Files to be processed within a zip file. Wildcards supported.  Default: \* (all)  Ex: -f word/\*.bin |
| -l LEVEL  --loglevel=LEVEL | logging level = debug, info, warning, error or critical (default=warning)  Ex: -l debug |
| -h | Show help |

# oleid – Quick Check for security issues

oleid <file>

Checks: file format, application, encryption, macros, Flash objects, OLE objects.

# olevba – Extract and scan VBA Macros

olevba [options] <file1> [file2 ...]

|  |  |
| --- | --- |
| -a  --analysis | display only analysis results, not the macro source code |
| -c  --code | display only VBA source code, do not analyze it |
| --decode | display all the obfuscated strings with their decoded content (Hex, Base64, StrReverse, Dridex, VBA) |
| --attr | display the attribute lines at the beginning of VBA source code |
| --reveal | display the macro source code after replacing all the obfuscated strings by their decoded content |
| --deobf | Attempt to deobfuscate VBA expressions (slow) |
| --relaxed | Do not raise errors if opening of substream fails |
| -t  --triage | triage mode, display results as a summary table (default for multiple files) |
| -d  --detailed | detailed mode, display full results (default for single file) |
| -j  --json | json mode, detailed in json format |

# mraptor – Detect Malicious Macros

mraptor [options] <file1> [file2 ...]

|  |  |
| --- | --- |
| -m  --matches | Show matched strings |

An exit code is returned based on the analysis result:

|  |  |
| --- | --- |
| 0: No Macro  1: Not MS Office  2: Macro OK | 10: ERROR  20: SUSPICIOUS |

# rtfobj – OLE objects in RTF

rtfobj [options] <file1> [file2 ...]

|  |  |
| --- | --- |
| -s <obj#>  --save=<obj#> | Save the object corresponding to the provided number to a file, for example "-s 2". Use "-s all" to save all objects at once. |

# Supported Formats

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Tool | doc  xls  ppt | docx/m  xlsx/m  pptx/m | rtf | mht  mhtml | Word 2003 xml | pub  vsd |
| oleid | X | - | - | - | - | X |
| olevba | X | X | - | X | X | X |
| mraptor | X | X | - | X | X | X |
| rtfobj | - | - | X | - | - | - |