INTERNET STORM CENTER

oledump.py Quick Reference

Nov 2020

Didier Stevens

oledump.py is a Python tool designed to analyze OLE2 files (aka Structured Storage, Compound File Binary Format).

blog.didierstevens.com/programs/oledump-py/

Microsoft Office Files

Prior to Office 2007 (.doc, .xls):
Microsoft Office files are binary (OLE2). File system like structure:
streams: data (cfr. files)
storages: streams/storages (cfr. Folders)

Office 2007+ (.docx, .xlsx, .docm, .xlsm):
Office files are ZIP containers (OOXML) which may include binary files.

VBA macros are always stored inside OLE2 files. For OOXML files, VBA macros are stored in vbaProject.bin inside the ZIP container.

Office file formats other than OLE2/OOXML are rarely used for VBA maldocs. RTF files for example cannot contain macros. But they are used for other exploits.

Command Line Arguments

```
oledump.py [options] filename
```

-m /-h	Manual/help page
--version	Show version
-s [number]	Select item for dumping
-d	dump
-x	hexdump
-a	ASCII dump
-A	ASCII dump and RLE
-S	Dump strings
-T	Do head & tail
-v	Decompress VBA
-r	Read raw file (with -v / -p)
-t	String translation
-e	Extract OLE embedded file
-i	Additional Info for item
-p PLUGINS	Load plugin
-q	Quiet
-y [file]	Load Yara File
-D [name]	Load decoder
-M	Print metadata
-c	Add calculated data like hashes to output
-V	verbose
-C [n]	Cut data
--storages	Include storages in report
-j	Json output
--password	ZIP password (default: infected)

oledump.py Output

```
#   Ind.  Size  Name
 1:       108   '\x01CompObj'
 2:  M    985   '_VBA_PROJECT_CUR/
                VBA/Sheet1
 3:  m    312   '_VBA_PROJECT_CUR/
                ThisWorkbook
```

Output Columns

Columns:

# - stream number
ind: indicators
- M – VBA Macro with Code
- m – VBA Macro with attributes only
- E – corrupt VBA code
- ! – unusual VBA code
- O – embedded object
- . – Storage
- R – Root entry

Tip

Use the -I option for a first analysis. This will add counters for compressed and compiled VBA source code.

Next, look for streams that contains macros ("M") and select them:

```
oledump.py -s 2 sample.xls
```

Use Cases

First analysis of a new sample:

```
oledump.py -i example.xls
```

Viewing a stream:

```
oledump.py -s 3 example.xls
```

Viewing compressed VBA source code from stream #7

```
oledump.py -s 7 -v example.xls
```

Extracting URLs from a downloader using the http heuristics plugin:

```
oledump.py -p \
plugin_http_heuristics \
example.xls
```

Analyze Excel 4 macros:

```
oledump.py -p blugin_biff \
--pluginoptions "-x" \
example.xls
```

Extracting all VBA code from a document

```
oledump.py -s a -v example.xls
```

Scanning a document with YARA rules:

```
oledump.py -y sample.yara \
example.xls
```

Using a decoder to brute force XOR key before matching Yara rules:

```
oledump.py -y sample.yara \
-D decoder_xor1.py example.xls
```

For more example, see Didier's ISC diaries:
https://isc.sans.edu/tag.html?tag=oledump

oledump.py Plugins

oledump.py supports plugins: small Python scripts to extend the functionality of oledump.py. Several plugins are distributed together with oledump. Plugins are invoked with option -p.

`plugin_biff`
Parse BIFF format in .xls files (e.g. Excel 4 macros)

`plugin_clsid`
display CLSID (if present)

`plugin_dridex`
identify/decode Dridex macros

`plugin_hifo`
Extract URLs from user forms

`plugin_http_heuristics`
Find obfuscated URLs

`plugin_jumplist`
Analyze Windows jump lists

`plugin_linear`
Linear cryptoanalysis

`plugin_msg`
identify Outlook MSG files

`plugin_msg_summary`
display summary (mail headers) for Outlook MSG file.

`plugin_msi`
Analyze MSI files (Windows Installer Package file)

`plugin_office_crypto`
Detect encryption method used

`plugin_ppt`
Analyze VBA macros in PowerPoint documents

`plugin_str_sub`
de-obfuscate strings by removing padding characters.

`plugin_stream_o`
Extract values from "stream o" (text boxes)

`plugin_stream_sample`
sample plugin to start developing your own

`plugin_vba`
searched for string concatenation in VBA code

`plugin_vba_dco`
Search for "Declare" statements and "CreateObject" calls

`plugin_vba_routines`
Search for "Sub" and "Function" in VBA code.

`plugin_vba_summary`
Summarizes VBA code (function names...)

`plugin_vbaproject`
Decrypt VBA Project password hash

`plugin_version_vba`
Identify the Office version used to create the document