Static Malware Analysis for IR CheatSheet

Mallicious Document Anallysis

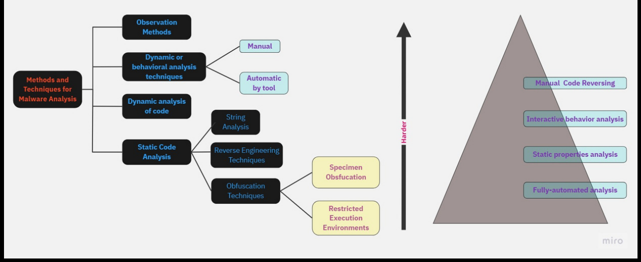
"Reverse-engineering malicious documents is the focus of this cheat sheet, which provides guidance and tool recommendations for analyzing files like Microsoft Office (DOC, XLS, PPT) and Adobe Acrobat (PDF) to uncover potential threats."

Approach

1. Recognize and analyze potentially malicious embedded code, such as shellcode, VBA macros, or JavaScript.
2. Isolate and extract any suspicious code present in the file.
3. If relevant, perform disassembly and/or debugging of shellcode.
4. If applicable, deobfuscate and thoroughly examine JavaScript, ActionScript, or VB macro code.
5. Acquire a comprehensive understanding of the succeeding steps in the infection chain.

Microsoft office file types:

Malware Analysis Techniques


Methods and Techniques for Malware Analysis
Observation Methods
Dynamic or behavioral analysis techniques
Manual
Automatic by tool
Dynamic analysis of code
Static Code Analysis
String Analysis
Reverse Engineering Techniques
Obfuscation Techniques
Specimen Obsfucation
Restricted Execution Environments
Harder
Manual Code Reversing
Interactive behavior analysis
Static properties analysis
Fully-automated analysis
miro

Static Analysis:

Conducting basic static analysis obviates the necessity of code execution and, instead, involves inspecting the file for indicators of malicious intent. This approach proves valuable in identifying malevolent infrastructure, libraries, or packed files.

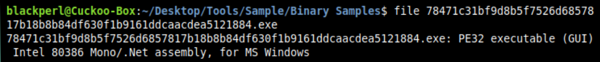
Commands & Tools

To view metadata about the malware sample

exiftool <File Name>

To view the file system about the malware samples

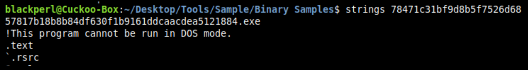
file <file type>



```
blackperl@Cuckoo-Box:~/Desktop/Tools/Sample/Binary Samples$ file 78471c31bf9d8b5f7526d68578
17b18b8b84df630f1b9161ddcaacdea5121884.exe
78471c31bf9d8b5f7526d6857817b18b8b84df630f1b9161ddcaacdea5121884.exe: PE32 executable (GUI)
 Intel 80386 Mono/.Net assembly, for MS Windows
```

To return the string characters into files

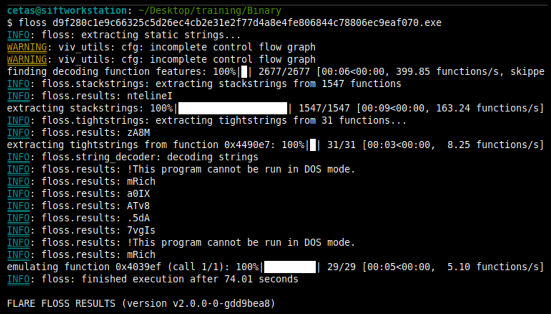
strings <file name>



```
blackperl@Cuckoo-Box:~/Desktop/Tools/Sample/Binary Samples$ strings 78471c31bf9d8b5f7526d68
57817b18b8b84df630f1b9161ddcaacdea5121884.exe
!This program cannot be run in DOS mode.
.text
`.rsrc
```

To automatically deobfuscate strings from malware binaries

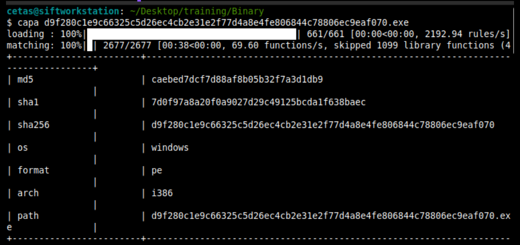
floss <file name>



```
cetas@siftworkstation: ~/Desktop/training/Binary
$ floss d9f280c1e9c66325c5d26ec4cb2e31e2f77d4a8e4fe806844c78806ec9eaf070.exe
INFO: floss: extracting static strings...
WARNING: viv_utils: cfg: incomplete control flow graph
WARNING: viv_utils: cfg: incomplete control flow graph
finding decoding function features: 100%|█| 2677/2677 [00:06<00:00, 399.85 functions/s, skippe
INFO: floss.stackstrings: extracting stackstrings from 1547 functions
INFO: floss.results: ntelineI
extracting stackstrings: 100%|████████████████| 1547/1547 [00:09<00:00, 163.24 functions/s]
INFO: floss.tightstrings: extracting tightstrings from 31 functions...
INFO: floss.results: zA8M
extracting tightstrings from function 0x4490e7: 100%|█| 31/31 [00:03<00:00,  8.25 functions/s]
INFO: floss.string_decoder: decoding strings
INFO: floss.results: !This program cannot be run in DOS mode.
INFO: floss.results: mRich
INFO: floss.results: a0IX
INFO: floss.results: ATv8
INFO: floss.results: .5dA
INFO: floss.results: 7vgIs
INFO: floss.results: !This program cannot be run in DOS mode.
INFO: floss.results: mRich
emulating function 0x4039ef (call 1/1): 100%|████████| 29/29 [00:05<00:00,  5.10 functions/s]
INFO: floss: finished execution after 74.01 seconds

FLARE FLOSS RESULTS (version v2.0.0-0-gdd9bea8)
```

To detect capabilities in executable files

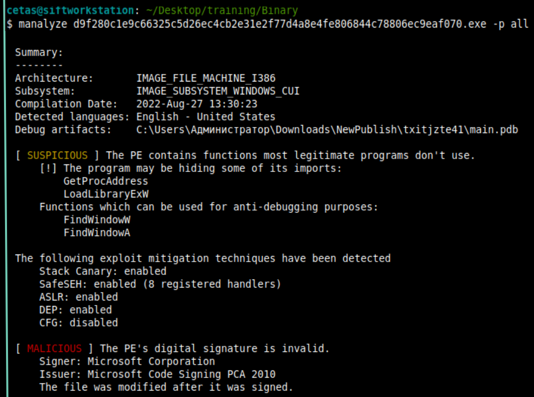
Capa <file name>



```
cetas@siftworkstation: ~/Desktop/training/Binary
$ capa d9f280c1e9c66325c5d26ec4cb2e31e2f77d4a8e4fe806844c78806ec9eaf070.exe
loading : 100%|████████████████████████████████████████| 661/661 [00:00<00:00, 2192.94 rules/s]
matching: 100%|█| 2677/2677 [00:38<00:00, 69.60 functions/s, skipped 1099 library functions (4
+------------------------+------------------------------------------------------------------
----------------+
| md5                    | caebed7dcf7d88af8b05b32f7a3d1db9
                |
| sha1                   | 7d0f97a8a20f0a9027d29c49125bcda1f638baec
                |
| sha256                 | d9f280c1e9c66325c5d26ec4cb2e31e2f77d4a8e4fe806844c78806ec9eaf070
                |
| os                     | windows
                |
| format                 | pe
                |
| arch                   | i386
                |
| path                   | d9f280c1e9c66325c5d26ec4cb2e31e2f77d4a8e4fe806844c78806ec9eaf070.ex
e               |
+------------------------+------------------------------------------------------------------
----------------+
```

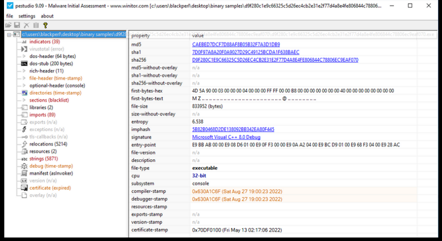
To conduct primary assessment on malware executable

manalyze <file name> -p all



```
cetas@siftworkstation: ~/Desktop/training/Binary
$ manalyze d9f280c1e9c66325c5d26ec4cb2e31e2f77d4a8e4fe806844c78806ec9eaf070.exe -p all

 Summary:
 --------
 Architecture:       IMAGE_FILE_MACHINE_I386
 Subsystem:          IMAGE_SUBSYSTEM_WINDOWS_CUI
 Compilation Date:   2022-Aug-27 13:30:23
 Detected languages: English - United States
 Debug artifacts:    C:\Users\Администратор\Downloads\NewPublish\txitjzte41\main.pdb

 [ SUSPICIOUS ] The PE contains functions most legitimate programs don't use.
     [!] The program may be hiding some of its imports:
         GetProcAddress
         LoadLibraryExW
     Functions which can be used for anti-debugging purposes:
         FindWindowW
         FindWindowA

 The following exploit mitigation techniques have been detected
     Stack Canary: enabled
     SafeSEH: enabled (8 registered handlers)
     ASLR: enabled
     DEP: enabled
     CFG: disabled

 [ MALICIOUS ] The PE's digital signature is invalid.
     Signer: Microsoft Corporation
     Issuer: Microsoft Code Signing PCA 2010
     The file was modified after it was signed.
```

To analyze executable files for various Windows operating systems



pestudio 9.09 - Malware Initial Assessment - www.winitor.com [c:\users\blackperl\desktop\binary samples\d9f280c1e9c66325c5d26ec4cb2e31e2f77d4a8e4fe806844c78806...

c:\users\blackperl\desktop\binary samples\d9f28...
- indicators (39)
- virustotal (error)
- dos-header (64 bytes)
- dos-stub (200 bytes)
- rich-header (11)
- file-header (time-stamp)
- optional-header (console)
- directories (time-stamp)
- sections (blacklist)
- libraries (2)
- imports (89)
- exports (n/a)
- exceptions (n/a)
- tls-callbacks (n/a)
- relocations (5214)
- resources (2)
- strings (5871)
- debug (time-stamp)
- manifest (asInvoker)
- version (n/a)
- certificate (expired)
- overlay (n/a)

property	value
md5	CAEBED7DCF7D88AF8B05B32F7A3D1DB9
sha1	7D0F97A8A20F0A9027D29C49125BCDA1F638BAEC
sha256	D9F280C1E9C66325C5D26EC4CB2E31E2F77D4A8E4FE806844C78806EC9EAF070
md5-without-overlay	n/a
sha1-without-overlay	n/a
sha256-without-overlay	n/a
first-bytes-hex	4D 5A 90 00 03 00 00 00 04 00 00 00 FF FF 00 00 B8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00
first-bytes-text	M Z .. .. @ .. ..
file-size	833952 (bytes)
size-without-overlay	n/a
entropy	6.538
imphash	5B82B0468D2DE138092BB342EA80F445
signature	Microsoft Visual C++ 8.0 Debug
entry-point	E9 BB AB 00 00 E9 08 D6 01 00 E9 0F F3 00 00 E9 0A A2 04 00 E9 BC D9 01 00 E9 68 F3 04 00 E9 28 AC
file-version	n/a
description	n/a
file-type	executable
cpu	32-bit
subsystem	console
compiler-stamp	0x630A1C6F (Sat Aug 27 19:00:23 2022)
debugger-stamp	0x630A1C6F (Sat Aug 27 19:00:23 2022)
resources-stamp	
exports-stamp	n/a
version-stamp	n/a
certificate-stamp	0x70DF0100 (Fri May 13 02:17:06 2022)

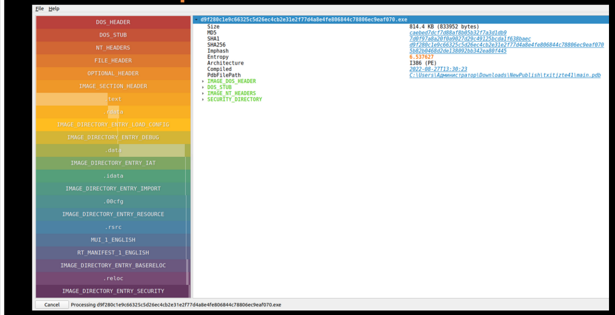
blackperldfir.com

Malware Binary Analysis Cheatsheet

Static Analysis

To view and dump in-memory PE files, as well as perform import table reconstruction

pe-tree <file name>


File Help
DOS_HEADER
DOS_STUB
NT_HEADERS
FILE_HEADER
OPTIONAL_HEADER
IMAGE_SECTION_HEADER
.text
.rdata
IMAGE_DIRECTORY_ENTRY_LOAD_CONFIG
IMAGE_DIRECTORY_ENTRY_DEBUG
.data
IMAGE_DIRECTORY_ENTRY_IAT
.idata
IMAGE_DIRECTORY_ENTRY_IMPORT
.00cfg
IMAGE_DIRECTORY_ENTRY_RESOURCE
.rsrc
MUI_1_ENGLISH
RT_MANIFEST_1_ENGLISH
IMAGE_DIRECTORY_ENTRY_BASERELOC
.reloc
IMAGE_DIRECTORY_ENTRY_SECURITY
Cancel

```
d9f280c1e9c66325c5d26ec4cb2e31e2f77d4a8e4fe806844c78806ec9eaf070.exe
  Size            814.4 KB (833952 bytes)
  MD5             caebed7dcf7d88af8b05b32f7a3d1db9
  SHA1            7d0f97a8a20f0a9027d29c49125bcda1f638baec
  SHA256          d9f280c1e9c66325c5d26ec4cb2e31e2f77d4a8e4fe806844c78806ec9eaf070
  Imphash         5b82b0468d2de138092bb342ea80f445
  Entropy         6.537627
  Architecture    I386 (PE)
  Compiled        2022-08-27T13:30:23
  PdbFilePath     C:\Users\Администратор\Downloads\NewPublish\txitjzte41\main.pdb
▸ IMAGE_DOS_HEADER
▸ DOS_STUB
▸ IMAGE_NT_HEADERS
▸ SECURITY_DIRECTORY
```

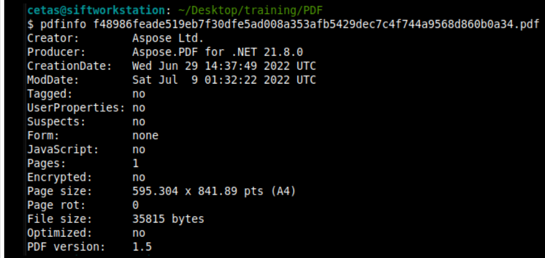
Processing d9f280c1e9c66325c5d26ec4cb2e31e2f77d4a8e4fe806844c78806ec9eaf070.exe

Malicious pdf Analysis

A PDF file describes text, graphics, and images in a device-independent format and resolution. It consists of objects that define the display of one or more pages.

To view info about the malicious pdf

pdfinfo <file name>



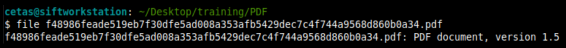
```
cetas@siftworkstation: ~/Desktop/training/PDF
$ pdfinfo f48986feade519eb7f30dfe5ad008a353afb5429dec7c4f744a9568d860b0a34.pdf
Creator:        Aspose Ltd.
Producer:       Aspose.PDF for .NET 21.8.0
CreationDate:   Wed Jun 29 14:37:49 2022 UTC
ModDate:        Sat Jul  9 01:32:22 2022 UTC
Tagged:         no
UserProperties: no
Suspects:       no
Form:           none
JavaScript:     no
Pages:          1
Encrypted:      no
Page size:      595.304 x 841.89 pts (A4)
Page rot:       0
File size:      35815 bytes
Optimized:      no
PDF version:    1.5
```

To view the metadata of the malicious pdf

exiftool <file name>

To determine the type of a file

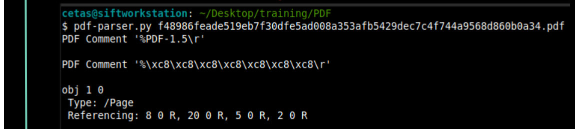
file <malicious file>



```
cetas@siftworkstation: ~/Desktop/training/PDF
$ file f48986feade519eb7f30dfe5ad008a353afb5429dec7c4f744a9568d860b0a34.pdf
f48986feade519eb7f30dfe5ad008a353afb5429dec7c4f744a9568d860b0a34.pdf: PDF document, version 1.5
```

To analyze and dissect PDF (Portable Document Format) files

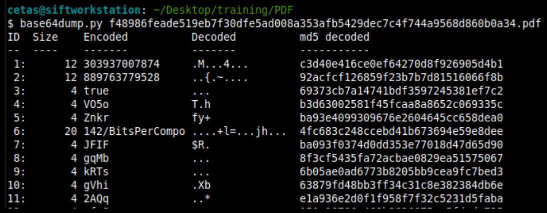
pdf-parser.py <malicious pdf file>



```
cetas@siftworkstation: ~/Desktop/training/PDF
$ pdf-parser.py f48986feade519eb7f30dfe5ad008a353afb5429dec7c4f744a9568d860b0a34.pdf
PDF Comment '%PDF-1.5\r'

PDF Comment '%\xc8\xc8\xc8\xc8\xc8\xc8\xc8\r'

obj 1 0
 Type: /Page
 Referencing: 8 0 R, 20 0 R, 5 0 R, 2 0 R
```

To Extract base64 strings from file

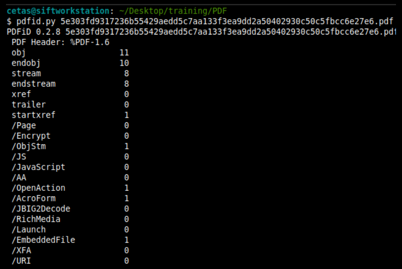
base64dump.py <malicious file>



```
cetas@siftworkstation: ~/Desktop/training/PDF
$ base64dump.py f48986feade519eb7f30dfe5ad008a353afb5429dec7c4f744a9568d860b0a34.pdf
ID  Size    Encoded          Decoded          md5 decoded
--  ----    -------          -------          -----------
 1:      12 303937007874     .M...4...        c3d40e416ce0ef64270d8f926905d4b1
 2:      12 889763779528     ..{.~....        92acfcf126859f23b7b7d81516066f8b
 3:       4 true             ...              69373cb7a14741bdf3597245381ef7c2
 4:       4 VO5o             T.h              b3d63002581f45fcaa8a8652c069335c
 5:       4 Znkr             fy+              ba93e4099309676e2604645cc658dea0
 6:      20 142/BitsPerCompo ....+l=...jh...  4fc683c248ccebd41b673694e59e8dee
 7:       4 JFIF             $R.              ba093f0374d0dd353e77018d47d65d90
 8:       4 gqMb             ...              8f3cf5435fa72acbae0829ea51575067
 9:       4 kRTs             ...              6b05ae0ad6773b8205bb9cea9fc7bed3
10:       4 gVhi             .Xb              63879fd48bb3ff34c31c8e382384db6e
11:       4 2AQq             ..*              e1a936e2d0f1f958f7f32c5231d5faba
```

To analyzing and identifying potential security risks in PDF (Portable Document Format) files

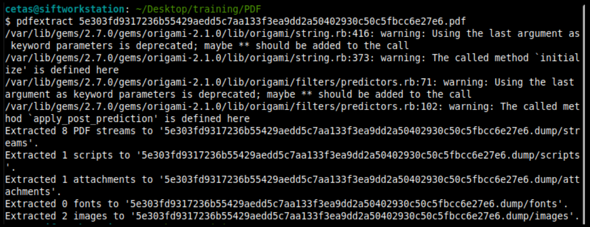
pdfid.py <malicious file>



```
cetas@siftworkstation: ~/Desktop/training/PDF
$ pdfid.py 5e303fd9317236b55429aedd5c7aa133f3ea9dd2a50402930c50c5fbcc6e27e6.pdf
PDFiD 0.2.8 5e303fd9317236b55429aedd5c7aa133f3ea9dd2a50402930c50c5fbcc6e27e6.pdf
 PDF Header: %PDF-1.6
 obj                   11
 endobj                10
 stream                 8
 endstream              8
 xref                   0
 trailer                0
 startxref              1
 /Page                  0
 /Encrypt               0
 /ObjStm                1
 /JS                    0
 /JavaScript            0
 /AA                    0
 /OpenAction            1
 /AcroForm              1
 /JBIG2Decode           0
 /RichMedia             0
 /Launch                0
 /EmbeddedFile          1
 /XFA                   0
 /URI                   0
```

To extracting specific content or elements from PDF (Portable Document Format) files

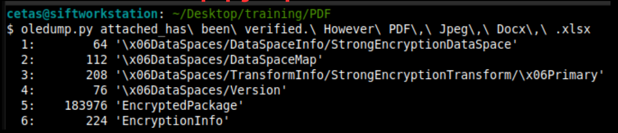
pdfextract <malicious pdf>



```
cetas@siftworkstation: ~/Desktop/training/PDF
$ pdfextract 5e303fd9317236b55429aedd5c7aa133f3ea9dd2a50402930c50c5fbcc6e27e6.pdf
/var/lib/gems/2.7.0/gems/origami-2.1.0/lib/origami/string.rb:416: warning: Using the last argument as
 keyword parameters is deprecated; maybe ** should be added to the call
/var/lib/gems/2.7.0/gems/origami-2.1.0/lib/origami/string.rb:373: warning: The called method `initial
ize' is defined here
/var/lib/gems/2.7.0/gems/origami-2.1.0/lib/origami/filters/predictors.rb:71: warning: Using the last
argument as keyword parameters is deprecated; maybe ** should be added to the call
/var/lib/gems/2.7.0/gems/origami-2.1.0/lib/origami/filters/predictors.rb:102: warning: The called met
hod `apply_post_prediction' is defined here
Extracted 8 PDF streams to '5e303fd9317236b55429aedd5c7aa133f3ea9dd2a50402930c50c5fbcc6e27e6.dump/str
eams'.
Extracted 1 scripts to '5e303fd9317236b55429aedd5c7aa133f3ea9dd2a50402930c50c5fbcc6e27e6.dump/scripts
'.
Extracted 1 attachments to '5e303fd9317236b55429aedd5c7aa133f3ea9dd2a50402930c50c5fbcc6e27e6.dump/att
achments'.
Extracted 0 fonts to '5e303fd9317236b55429aedd5c7aa133f3ea9dd2a50402930c50c5fbcc6e27e6.dump/fonts'.
Extracted 2 images to '5e303fd9317236b55429aedd5c7aa133f3ea9dd2a50402930c50c5fbcc6e27e6.dump/images'.
```

To analyze Microsoft Office files (such as Word, Excel, PowerPoint) and other OLE (Object Linking and Embedding) files, which are compound files that can contain various embedded objects like macros, scripts, links, and other components.

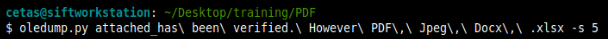
Oledump.py <pdf-extract file >



```
cetas@siftworkstation: ~/Desktop/training/PDF
$ oledump.py attached_has\ been\ verified.\ However\ PDF\,\ Jpeg\,\ Docx\,\ .xlsx
  1:        64 '\x06DataSpaces/DataSpaceInfo/StrongEncryptionDataSpace'
  2:       112 '\x06DataSpaces/DataSpaceMap'
  3:       208 '\x06DataSpaces/TransformInfo/StrongEncryptionTransform/\x06Primary'
  4:        76 '\x06DataSpaces/Version'
  5:    183976 'EncryptedPackage'
  6:       224 'EncryptionInfo'
```

To extract streams from an OLE file

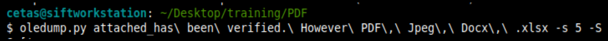
oledump.py <file-name> -s <stream-value>



```
cetas@siftworkstation: ~/Desktop/training/PDF
$ oledump.py attached_has\ been\ verified.\ However\ PDF\,\ Jpeg\,\ Docx\,\ .xlsx -s 5
```

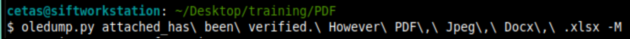
To analyze and display specific information about the embedded streams present in an OLE file

oledump.py <file-name> -s <stream-value> -S



```
cetas@siftworkstation: ~/Desktop/training/PDF
$ oledump.py attached_has\ been\ verified.\ However\ PDF\,\ Jpeg\,\ Docx\,\ .xlsx -s 5 -S
```

To enable the extraction of metadata for the selected embedded object (stream)



```
cetas@siftworkstation: ~/Desktop/training/PDF
$ oledump.py attached_has\ been\ verified.\ However\ PDF\,\ Jpeg\,\ Docx\,\ .xlsx -M
```

blackperldfir.com