PEEPDF – PDF Analysis Tool

Jose Miguel Esparza
@EternalTodo

http://peepdf.eternal-todo.com

@peepdf

peepdf – PDF Analysis Tool

- Characteristics
  - **Python**
  - Command line
  - **Interactive Console**
  - Command file option → Batch, Automation
  - **XML/JSON Output**

peepdf – PDF Analysis Tool

- Characteristics
  - Continue development → **Alive project!!**
  - **Update command** ← GIT files
  - Included in **BackTrack/Kali Linux** and **REMnux**
  - **All in One**

    "peepdf sounds like the Swiss army knife of PDF security apps"

peepdf – PDF Analysis Tool

- Why choosing **peepdf**?
  - Support for:
    - **Encryption**
    - ***Object Streams*** (compressed objects)
    - Most used filters
    - *FlateDecode / LZWDecode* **Parameters**
  - **Javascript Analysis**
  - **Shellcode emulation**

peepdf – PDF Analysis Tool

- Why choosing **peepdf**?
  - Shows **Suspicious Elements**
  - Shows potential **Vulnerabilities**
  - **Powerful Interactive Console**
  - **Easy extraction of objects / JS code / shellcode**
  - **PDF Obfuscation (bypassing AVs)**
  - **Alive project!!**

peepdf – PDF Analysis Tool

- Analysis
  - **Encrypted** files
  - **Compressed** objects
  - **Malformed documents** support
  - Decoding: hexadecimal, octal, names
  - Most used filters (7)
  - **References** in objects and to objects
  - **ASCII and HEX search** (including *streams*)

peepdf – PDF Analysis Tool

- Analysis
  - **Physical structure** (offsets)
  - Tree structure (**logical**)
  - **Metadata**
  - Changes between versions (**changelog**)
  - Extraction of different **document versions**
  - **Scoring system (new!)**

peepdf – PDF Analysis Tool

- Analysis
  - **Javascript** analysis and modification (*V8*)
    - beautify, unescape, replace, join, eval, jjdecode
  - **Shellcode** analysis (*pylibemu*)
  - Use of **variables** to improve analysis (*set* command)
  - Easy **extraction** of information (> , >>, $>, $>>)
  - **XOR search** and **brute force**

peepdf – PDF Analysis Tool

- Creation / Modification (Pentesting)
  - Basic PDF creation
  - Creation of PDF with **Javascript execution**
  - Object **compression** (object streams)
  - **Encrypted** files
  - **Nested PDFs** creation

peepdf – PDF Analysis Tool

- Creation / Modification (Pentesting)
  - **Malformed** PDFs
  - Strings and names **codification**
  - **Filter** modification
  - Object **modification**

peepdf – PDF Analysis Tool

- Execution

```
Usage: peepdf [options] PDF_file

Version: peepdf 0.3 r253

Options:
  -h, --help                    Shows 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.
  -j, --json                    Shows the document information in JSON format.
```

peepdf – PDF Analysis Tool

- Execution
  - Basic
    - Shows document information
    - Not interactive
    - *peepdf.py sample.pdf*

peepdf – PDF Analysis Tool

- Execution
  - Interactive console
    - Shows advanced information
    - Permits interact with the document → commands
    - Powerful
    - *peepdf.py -i* [*sample.pdf*]

peepdf – PDF Analysis Tool

- Execution
  - Interactive console: commands

```
PPDF> help

Documented commands (type help <topic>):
========================================
bytes           exit          js_join          quit          set
changelog       filters       js_unescape      rawobject     show
create          hash          js_vars          rawstream     stream
decode          help          log              references    tree
decrypt         info          malformed_output replace       vtcheck
embed           js_analyse    metadata         reset         xor
encode          js_beautify   modify           save          xor_search
encode_strings  js_code       object           save_version
encrypt         js_eval       offsets          sctest
errors          js_jjdecode   open             search
```

peepdf – PDF Analysis Tool

- Execution
  - Batch
    - Mix of basic and interactive modes
    - Not interactive, but…
    - …permits execution of interactive commands in batch
    - Commands stored in a file
    - *peepdf.py -s command_file.txt sample.pdf*

peepdf – PDF Analysis Tool

- Commands

  https://github.com/jesparza/peepdf/wiki/Commands

peepdf – PDF Analysis Tool

- Commands
  - Console
    - **help** – Shows help
    - **log** – Permits logging commands to a file
    - **open** – Opens a new PDF file
    - **reset** – Resets variables or screen
    - **quit**
    - **exit**

peepdf – PDF Analysis Tool

- Commands
  - Showing information
    - Whole document
      - **info** – Shows information of objects and document
      - **tree** – Shows the logical structure of the document
      - **offsets** – Shows the physical structure
      - **hash** – Permits making a hash of some raw bytes
      - **bytes** – Shows raw bytes of the document

peepdf – PDF Analysis Tool

- Commands
  - Showing information
    - Whole document
      - **metadata** – Shows metadata information
      - **changelog** – Shows changes between versions
      - **save_version** – Saves one specific version
      - **errors** – Shows parsing errors
      - **vtcheck** – Checks the hash on VirusTotal

peepdf – PDF Analysis Tool

- Commands
  - Showing information
    - Objects
      - **object** – Shows objects, after decryption/decoding
      - **rawobject** – Shows raw objects
      - **stream** – Shows streams, after decryption/decoding
      - **rawstream** – Shows raw streams
      - **references** – Shows references in and to objects
      - **hash** – Permits making a hash of objects, streams…
      - **vtcheck** – Checks the hash on VirusTotal

peepdf – PDF Analysis Tool

- Commands
  - Extracting information
    - Shell redirection is easier ;)
      - **Files**
        - » *stream 6 > stream6_file*
        - » *js_code 12 >> pdf_js_code_file*
      - **Variables**
        - » *js_unescape variable myVar $> unescaped_sh*
        - » *rawstream 5 $>> all_my_rawstreams_var*

peepdf – PDF Analysis Tool

- Commands
  - Javascript functions
    - **js_code** – Shows the Javascript code of an object
    - **js_eval** – Runs PyV8 with the given JS code
    - **js_analyse** – Tries to execute and analyze the JS code
    - **js_beautify** – Beautifies the Javascript code
    - **js_unescape** – Unescapes the escaped JS code
    - **js_join** – Joins separated Javascript strings
    - **js_vars** – Shows the defined variables and their content
    - **js_jjdecode** – Decoded the code obfuscated with jjencode

peepdf – PDF Analysis Tool

- Commands
  - Shellcode emulation
    - **sctest** – Libemu sctest wrapper for Python

peepdf – PDF Analysis Tool

- Commands
  - Modification / Creation
    - **modify** – Modifies objects
    - **filters** – Modifies or removes the filter of a given stream
    - **decode** – Decodes raw bytes / streams / files / variables
    - **encode** – Encodes raw bytes / streams / files / variables
    - **encode_strings** – Obfuscates strings / names of objects

peepdf – PDF Analysis Tool

- Commands
  - Modification / Creation
    - **embed** – Embeds a file in the PDF document
    - **encrypt** – Encrypts the document
    - **malformed_output** – Writes malformed documents
    - **create** – Creates basic PDF documents (JS execution too)
    - **save** – Saves the document after modifications

peepdf – PDF Analysis Tool

- Commands
  - Misc
    - **set** – Creates a variable with the given value
    - **search** – Searches the document for ASCII and HEX chars
    - **show** – Shows the content of the given variable
    - **xor** – Performs XOR operations over streams/bytes/files…
    - **xor_search** – Performs XOR and searches some pattern

peepdf – PDF Analysis Tool

- Commands
  - Not just for PDF documents
    - **decode**
    - **encode**
    - **hash**
    - **js_***
    - **replace**
    - **sctest**
    - **vtcheck**
    - **xor***

peepdf – PDF Analysis Tool

- Scoring system
  - Google Summer of Code (GSoC)
  - Developed by Rohit Dua
  - Objective: give advice about the maliciousness
  - Status: beta
  - Based on a list of indicators
    - Testing
    - Experience

```
File: exploit_adobe_flatecode_predictor02.pdf
MD5: a1e391e96e17b21a1cf0c519b0fc14b6
SHA1: 727b5cccd367333de37144adafed2923b6c724a7
Size: 2730 bytes
Pages Number: 1
Maliciousness Score: 6.6/10
Version: 1.5
Binary: True
Linearized: False
Encrypted: False
Updates: 0
Objects: 7
Streams: 2
Comments: 0
Errors: 0

Version 0:
        Catalog: 1
        Info: No
        Objects (7): [1, 2, 3, 4, 5, 6, 7]
        Streams (2): [6, 7]
                Encoded (2): [6, 7]
        Objects with JS code (1): [6]
        Suspicious elements:
                /OpenAction: [1]
                /JS: [5]
                /JavaScript: [5]

        Suspicious Indicators:
                Obfuscated names: [1, 2, 3, 4, 5, 6, 7]
```

peepdf – PDF Analysis Tool

- Scoring system
  - Based on different indicators
    - Number of pages
    - Number of stream filters
    - Broken/Missing cross reference table
    - Obfuscated elements: names, strings, Javascript code.
    - Malformed elements: garbage bytes, missing tags…
    - Encryption with default password
    - Suspicious elements: Javascript, event triggers, actions, known vulns…
    - Big streams and strings
    - Objects not referenced from the Catalog

peepdf – PDF Analysis Tool

- Using *peepdf* as a library
  - SWF Mastah (Brandon Dixon)

```
__description__ = 'Snatch the SWF!'
__author__ = 'Brandon Dixon'
__version__ = '1.0'
__date__ = '2011/11/07'

import simplejson as json
import optparse
from PDFConsole import PDFConsole
from PDFCore import PDFParser

def snatch(file, out):
    pdfParser = PDFParser()
    ret,pdf = pdfParser.parse(file, True, False)
    statsDict = pdf.getStats()
    objs = []
    count = 0
    for version in range(len(statsDict['Versions'])):
        body = pdf.body[count]
        objs = body.objects
```

peepdf – PDF Analysis Tool

- **TODO**
  - Support more PDF vulnerabilities
  - Missing filters: JBIG2, JPX
  - Improve automatic Javascript analysis
    - Add support for PDF JS functions (getAnnots…)
  - Web interface?

peepdf – PDF Analysis Tool

- Analysis examples
  - CVE-2013-2729 PDF exploit analysis
    http://eternal-todo.com/blog/cve-2013-2729-exploit-zeusp2p-gameover
    http://eternal-todo.com/blog/CVE-2013-2729-obfuscated-pdf-exploits
  - Extracting streams and shellcodes, the easy way
    http://eternal-todo.com/blog/extract-streams-shellcode-peepdf
  - CVE-2011-2462 PDF exploit Analysis
    http://eternal-todo.com/blog/cve-2011-2462-exploit-analysis-peepdf
  - SEO Sploit Pack Analysis
    http://eternal-todo.com/blog/seo-sploit-pack-pdf-analysis
  - Analyzing PDF files with peepdf (Lenny Zeltser)
    http://blog.zeltser.com/post/6780160077/peepdf-malicious-pdf-analysis

peepdf – PDF Analysis Tool

- References
  - GitHub Project
    https://github.com/jesparza/peepdf
  - Installation
    https://github.com/jesparza/peepdf/wiki/Installation
  - Execution
    https://github.com/jesparza/peepdf/wiki/Execution
  - Commands
    https://github.com/jesparza/peepdf/wiki/Commands

peepdf – PDF Analysis Tool

- Who is using peepdf?
  - Viper
    https://github.com/viper-framework/viper
  - Cuckoo modified (Accuvant)
    https://github.com/brad-accuvant/cuckoo-modified/tree/00ad13c94cc7453c40ed6152d16009ca1c8ed6f2/lib/cuckoo/common/peepdf
  - Thug
    https://github.com/buffer/thug
  - PDF X-RAY
    https://github.com/9b/pdfxray_public

peepdf – PDF Analysis Tool

Thanks!!

Jose Miguel Esparza
http://eternal-todo.com
@EternalTodo