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 Output**

peepdf – PDF Analysis Tool

- Characteristics
  - Continue development → **Alive project!!**
  - **Update command** ← SVN files
  - Included in **BackTrack** 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 (5)
  - **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**

peepdf – PDF Analysis Tool

- Analysis
  - **Javascript** analysis and modification (*Spidermonkey*)
    - beautify, unescape, replace, join
  - **Shellcode** analysis (*sctest, Libemu*)
  - 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**
  - **Filters** modification
  - Object **modification**

peepdf – PDF Analysis Tool

- Execution

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

Version: peepdf 0.1 r72

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.
  -f, --force-mode      Sets force parsing mode to ignore errors.
  -l, --loose-mode      Sets loose parsing mode to catch malformed objects.
  -u, --update          Updates peepdf with the latest files from the repository.
  -v, --version         Shows program's version number.
  -x, --xml             Shows the output of the analysis in XML 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

```
$ ./peepdf -i


PPDF> help

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

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

  http://code.google.com/p/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

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…

peepdf – PDF Analysis Tool

- Commands
  - Extracting information
    - Output redirection is possible
      - **set**
        - » *set output file path_to_my_file*
        - » *set output variable myVar*

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** – Runs Spidermonkey 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

peepdf – PDF Analysis Tool

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

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

- **TODO**
  - Nested PDFs analysis
  - Missing filters: CCITTFax, JBIG2, DCT, JPX
  - Improve automatic Javascript analysis
    - Add support for PDF JS functions (getAnnots…)
  - GUI?
  - ActionScript?

peepdf – PDF Analysis Tool

- Analysis examples
  - 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
  - Google Code Project
    http://peepdf.googlecode.com
  - Installation
    http://code.google.com/p/peepdf/wiki/Installation
  - Execution
    http://code.google.com/p/peepdf/wiki/Execution
  - Commands
    http://code.google.com/p/peepdf/wiki/Commands

Thanks!!

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

http://peepdf.eternal-todo.com **@peepdf**