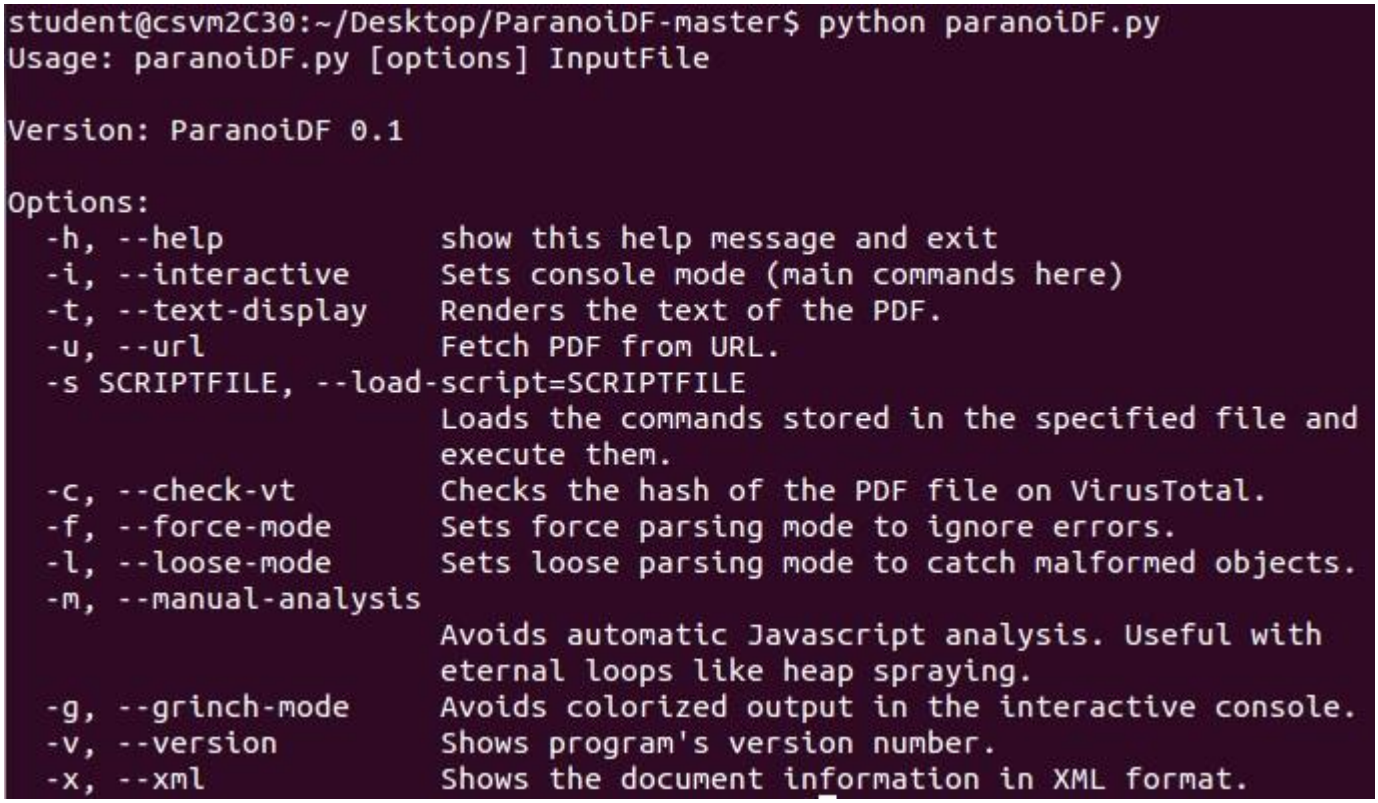
Welcome to the ParanoiDF wiki.



Tools/functions when running the main script: Usage: paranoiDF.py [options] InputFile

- **-h Help.** Displays a help message for the main script, detailing what each option does (no input file needed).
- **-i Interactive Console.** Executes the interactive console. This console contains most of the main tools of this tool (no input file needed).
- **-t Text Display.** This option parses and renders all pure text inside a PDF. It does this by executing the tool pdf2txt.py (from PDFMINER), through an OS call.
- **-u URL.** Downloads the PDF from the link and saves it in a new directory named after the website it was obtained from. This option simply uses an OS call to the command WGET.
- **-s Script.** Executes a file containing commands to be executed on the interactive console. Just a simple .script file works, and each command should be on a new line.
- **-f Force mode.** If a PDF being input is having problems parsing, try this. It ignores parsing errors. Beware though as doing this mode will mean non-PDF files can be analysed!
- **-l Loose mode.** This mode is similiar to Force mode, except it is designed to catch malformed objects (which are common in malicious PDFs.
- **-m Manual Analysis.** This avoids automatic JavaScript analysis. Useful with eternal loops like heap spraying.
- **-g Grinch mode.** This avoids colourized output in the interactive console.
- **-v Version.** Displays the tools version.
- **-x XML.** Shows the PDF analysis in XML format.

Interactive Console: Type "help" to get a list of commands. Type "help [command]" to get a description/usage on specific command.

```
student@csvm2C30:~/Desktop/ParanoiDF-master$ python paranoiDF.py -i
ParanoiDF> help

Documented commands (type help <topic>):
========================================
bytes           errors       js_join           rawobject     set
changelog       exit         js_unescape       rawstream     show
crackpw         extractJS    log               redact        stream
decode          filters      malformed_output  references    tree
decrypt         hash         metadata          removeDRM     vtcheck
embedf          help         modify            replace       xor
embedjs         info         object            reset         xor_search
encode          js_beautify  offsets           save
encode_strings  js_code      open              save_version
encrypt         js_jjdecode  quit              search

ParanoiDF>
```

- **bytes** This shows or stores a specified number of bytes of a file from the beginning of a specific offset.
- **changelog** Displays the change log of a PDF document or version of the document.
- **crackpw** This executes a PDF cracking tool called "pdfcrack" by performing an OS call. The command allows the user to input a custom dictionary, perform a benchmark or continue from a saved state file. If no custom dictionary is input, this command will attempt to brute force a password using a modifiable charset text file in directory "ParanoiDF/pdfcrack".
- **decode** Decodes the content of the specified variable, file or raw bytes using algorithms such as Base64, LZW, FlateDecode Etc. (See help for more algorithms it supports)
- **encode** Encodes the content of the specified variable, file or raw bytes using algorithms such as Base64, LZW, FlateDecode Etc. (See help for more algorithms it supports)
- **decrypt** This uses an OS call to tool "QPDF" which decrypts the PDF document and outputs the decrypted file. This requires the user-password.
- **encrypt** Encrypts an input PDF document with any password you specify. Uses 128-bit RC4 encryption.
- **encode_strings** Encodes the strings and names included in the file, object or trailer.
- **embedf** Create a blank PDF document with an embedded file. This is for research purposes to show how files can be embedded in PDFs. This command imports Didier Stevens Make-pdf-embedded.py script as a module.
- **embedjs** Similiar to "embedf", but embeds custom JavaScript file inside a new blank PDF document. If no custom JavaScript file is input, a default app.alert messagebox is embedded.
- **errors** Shows the errors of the file or object (object ID, xref or trailer).
- **exit** Exits the Interactive Console.
- **extractJS** This attempts to extract any embedded JavaScript in a PDF document. It does this by importing Jsunpackn's "pdf.py" JavaScript tool as a module, then executing it on the file.
- **filters** Shows the filters found in the stream object or set the filters in the object (first filter is used first). Valid filters, for example, are: LZW, FlateDecode and JBIG2Decode.
- **hash** Generates the hash (MD5/SHA1/SHA256) of the specified source: raw bytes of the file, objects and streams and content of files or variables.

- **info** Displays information about a PDF document, or object (object ID, xref or trailer).
- **js_beautify** Beautifies the JavaScript code stored in the specified variable, file or object.
- **js_code** Shows the JavaScript code found in an object.
- **js_jjdecode** Decodes the JavaScript code stored in a specific variable, file or object using the jjencode/decode algorithm by Yosuke Hasegawa (http://utf-8.jp/public/jjencode.html).
- **js_join** Joins some strings separated by quotes and stored in the specified variable or file in a unique one.
- **js_unescape** Unescapes the escaped characters stored in the specified variable or file.
- **log** Shows the state of current logging. Allows you to start logging in a specified file.
- **malformed_output** Enables malformed output when saving the file.
- **metadata** Shows the metadata of the PDF document or version of the document.
- **modify** Modifies the object or stream specified. It's possible to use a file to retrieve the stream content (ONLY for stream content).
- **object** Shows the content of the object after being decoded and decrypted.
- **offsets** Shows the physical map of the file or the specified version of the document.
- **open** Opens and parses the specified PDF file. (If exception raised, try -f Force mode or -l Loose mode).
- **quit** Exits the Interactive Console.
- **rawobject** Shows the content of the object without being decoded or decrypted (object_id, xref, trailer).
- **rawstream** Shows the stream content of the specified document version before being decoded and decrypted.
- **redact** Generate a list of words that will fit inside a redaction box in a PDF document. The words (with a custom sentence) can then be parsed in a grammar parser and a custom amount can be displayed depending on their score. This command requires a tutorial to use. Please read "redactTutorial.pdf" in directory "ParanoiDF/docs".
- **references** Shows the references in the object or to the object in the specified version of the document.
- **removeDRM** Remove DRM (editing, copying etc.) restrictions from PDF document and output to a new file. This does not need the owner-password and there is a possibility the document will lose some formatting. This command works by calling Calibre's "ebook-convert" tool.
- **replace** Replace a specified string with another one in the PDF document.
- **reset** Cleans the console and resets the stored variable value to the default one if applicable.
- **save** Save file to disk.
- **save_version** Save the selected file version to disk.
- **search** Search the specified string or hexadecimal string in the objects (decoded and encrypted streams included).
- **set** Sets the specified variable value or creates one with this value. Without parameters all the variables are shown. (Do command "help set" for more information).
- **show** Shows the value of the specified variable.
- **stream** Shows the object stream content of the specified version after being decoded and decrypted (if necessary)
- **tree** Shows the tree graph of the file or specified version.
- **vtcheck** Checks the hash of the specified source on VirusTotal: raw bytes of the file, objects and streams, and the content of files or variables. If no parameters are specified then the hash of the PDF document will be checked.
- **xor** Performs an XOR operation using the specified key with the content of the specified file or variable, raw bytes of the file or stream/rawstream. If the key is not specified then a bruteforcing XOR is performed.

- **xor_search** Searches for the specified string in the result of an XOR brute forcing operation with the content of the specified file or variable, raw bytes of the file or stream/rawstream. The output shows the offset/s where the string is found.