# Instruction of running feature extraction tool

First you need to login to the system using your credentials.

## About Dataset

We have a collection of pdf files prepared fot this project. The /dataset/ folder contains two subfolders - malicious ( bad files ) - benign ( good files ) Each folder contains 5000 pdf files in them.

## Feature Extraction

We have prepare a feature extractor tool that will help you extracting relevant properties from pdf file. it accept 3 parameters: - 1. input path [–in](ex.%20/dataset/malicious) - 2. category [–cat ] ( malicious or benign) - 3. output path [–out](ex.%20features/)

usage for malicious : extract\_feature –in /dataset/malicious –cat malicious –out features usage for benign : extract\_feature –in /dataset/benign –cat benign –out features

You can find the extracted features in json format in respective category inside output directory . for above example . features/malicious/ will contains all the extracted feature for malicious category

This feature extractor utilize the open source library if you are intrested you can check this link https://pypdf2.readthedocs.io/en/3.0.0/

## About extracted Features

1. exif\_signatures Description: Metadata signatures obtained from the Exchangeable image file format (EXIF). Type: Dictionary Example: {}
2. file\_size Description: Size of the file in bytes. Type: Integer Example: 116616
3. pdfid\_signatures Description: Signatures obtained using PDFiD tool. Type: Dictionary Example: {}
4. pypdf\_uris Description: List of URIs (Uniform Resource Identifiers) extracted using PyPDF2 library. Type: List of Strings Example: perl [ “http://getpdf.pw/book?res=123&isbn=9780375864957&kwd=The%20Red%20Blazer%20Girls:%20The%20Secret%20Cellar”, “https://img1.wsimg.com/blobby/go/74966dab-a283-4c4c-a9b9-f0069f6329e4/the-reception-year-in-action-revised-and-updat.pdf”]
5. regex\_uris Description: List of URIs extracted using regular expressions. Type: List of Strings Example: [“https://static.s123-cdn-static-a.com/uploads/4660132/normal\_61b0fa3584419.pdf”]
6. regex\_urls Description: List of URL patterns extracted using regular expressions. Type: List of Strings Example:

[ “http://getpdf.pw/book?res=123”]

1. scripts Description: Information about scripts present in the document. Type: Dictionary Example: { “iframe”: [], “urls”: [] }
2. static\_properties Description: Various static properties extracted from the document. Type: Dictionary

* JBIG2Decode Description: This property indicates whether the JBIG2Decode filter is used in the PDF. JBIG2 is an image compression standard. Value: “0” (Not used), “1” (Used)
* XML\_forms Description: Indicates the presence of XML forms in the PDF document. Value: “0” (Not present), “1” (Present)
* acro\_form Description: Indicates the presence of AcroForms (interactive forms in PDF) in the document. Value: “0” (Not present), “1” (Present)
* auto\_action Description: Indicates whether there are automatic actions defined in the PDF. Value: “0” (Not defined), “1” (Defined)
* colors Description: Indicates the use of color in the PDF. Value: “0” (No color), “1” (Color used)
* cross\_reference\_table Description: Indicates the integrity of the cross-reference table in the PDF. Value: “0” (Not valid), “1” (Valid)
* embedded\_files Description: Indicates the presence of embedded files in the PDF. Value: “0” (Not present), “1” (Present)
* file\_size Description: Size of the file in bytes.
* java\_script Description: Indicates the presence of JavaScript in the PDF. Value: “0” (Not present), “1” (Present)
* js Description: Alias for JavaScript property.
* launch\_action Description: Indicates the presence of launch actions in the PDF. Value: “0” (Not present), “1” (Present)
* object\_end Description: Position of the end of an object in the PDF.
* object\_start Description: Position of the start of an object in the PDF.
* object\_streams Description: Indicates the use of object streams in the PDF. Value: “0” (Not used), “1” (Used)
* open\_action Description: Indicates the presence of open actions in the PDF. Value: “0” (Not present), “1” (Present)
* page\_count Description: Number of pages in the PDF.
* rich\_media Description: Indicates the presence of rich media (audio, video) in the PDF. Value: “0” (Not present), “1” (Present)
* start\_cross\_reference\_table Description: Indicates the start of the cross-reference table in the PDF. Value: “0” (Not present), “1” (Present)
* stream\_end Description: Position of the end of a stream in the PDF.
* stream\_start Description: Position of the start of a stream in the PDF.
* trailer\_dictionary Description: Indicates the presence of the trailer dictionary in the PDF. Value: “0” (Not present), “1” (Present)

example: { “JBIG2Decode”: “0”, “XML\_forms”: “0”, “acro\_form”: “0”, “auto\_action”: “0”, “colors”: “0”, “cross\_reference\_table”: “1”, “embedded\_files”: “0”, “file\_size”: “117”, “java\_script”: “0”, “js”: “0”, “launch\_action”: “0”, “object\_end”: “79”, “object\_start”: “79”, “object\_streams”: “0”, “open\_action”: “0”, “page\_count”: “8”, “rich\_media”: “0”, “start\_cross\_reference\_table”: “1”, “stream\_end”: “15”, “stream\_start”: “15”, “trailer\_dictionary”: “1” }

1. yara\_signatures Description: Signatures obtained using YARA rules. Type: List of Strings Example: [“without\_attachments”, “without\_images”, “with\_urls”, “invalid\_trailer\_structure”, “contentis\_base64”] NOTE: these yara rules differs with every files