

**YALDA**

**Yalda, Automated**

**Bulk Intelligence**

**Gita Ziabari**

# Yalda, Automated Bulk Intelligence

## Introduction

Massive amounts of logs, suspect files, alerts and data makes it impossible to respond everything on daily basis. It is essential to automate process of analyzing files and apply intelligence in collecting data and clustering files based on the similarities. Yalda is an automated bulk intelligence tool that scans and analyzes files with different techniques and algorithms to collect detailed information on each file and extract malicious URLs and domains from each. Yalda categorizes files as malicious, suspicious or clear with a severity of 1 to 5, 1 meaning *clear* and 5 *malicious*. The information that Yalda provides on each file could be used to save hours of manual analysis on a single file. The structured clustered results make Yalda useful for analysts, data scientist, researchers and whoever interested in automated bulk intelligence.

Following are some of the domains that Yalda covers:

* Automated Bulk Intelligence Collection tool.
* File Scanner & Analyzer to collect detailed information of the file.
* Collects embedded objects, URLs and Domains.
* Categorizes Files as malicious, suspicious or clear with severity from 1 to 5(1 meaning clear and 5 malicious).
* Clusters malicious hashes and list of associated strings with them.

Yalda results could be used in the following domains:

* Input of different type of feeds such as malicious hashes, malicious URLs, malicious domains, etc.
* A scanning tool for researchers and analysts to extract selective information on a large scale.
* Generating a collection for writing Yara rules on specific malware characteristics.
* Testing tool for testing the capability of your system in detecting and categorizing malicious files.
* A smart feed to cuckoo sandbox that makes it possible to send selective data to cuckoo sandbox.

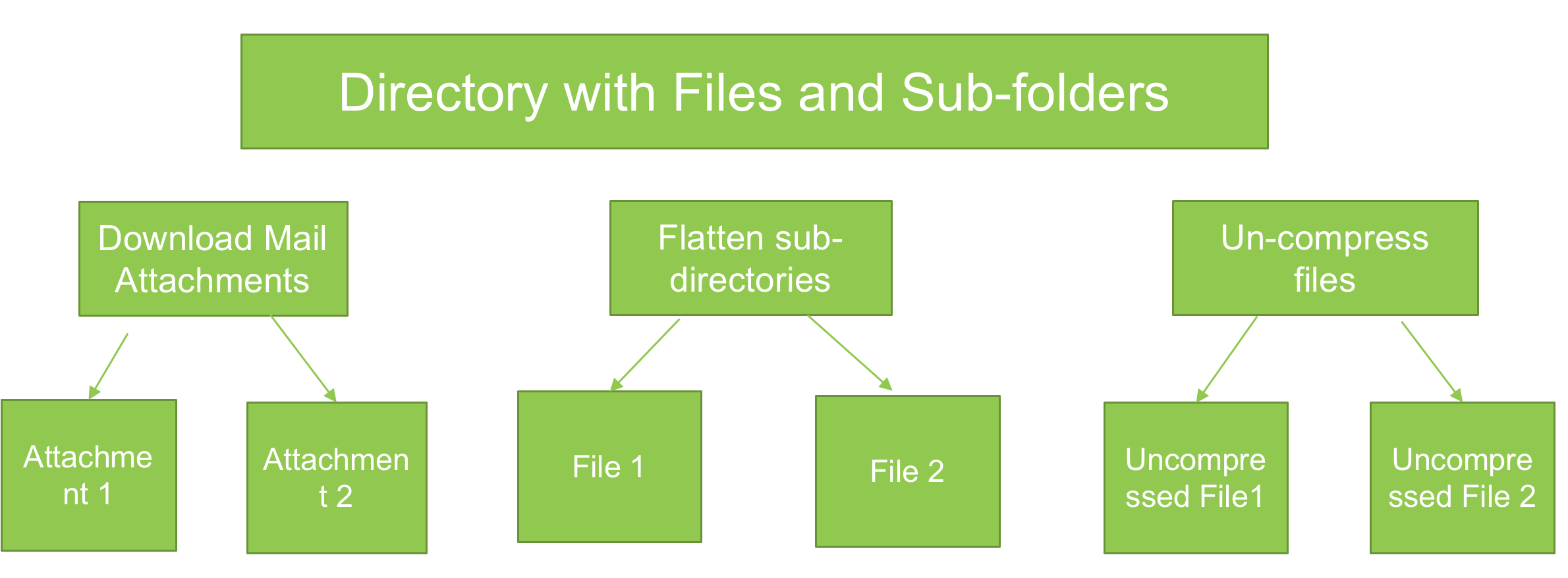
Yalda Framework

Yalda evaluates files in four main phases: extracting files, scanning and analyzing files, applying decoders, collecting detailed information about the file through different techniques and inserting the obtained results in database.

Due to clustering techniques introduced in Yalda, it gets more powerful as running more samples through it.

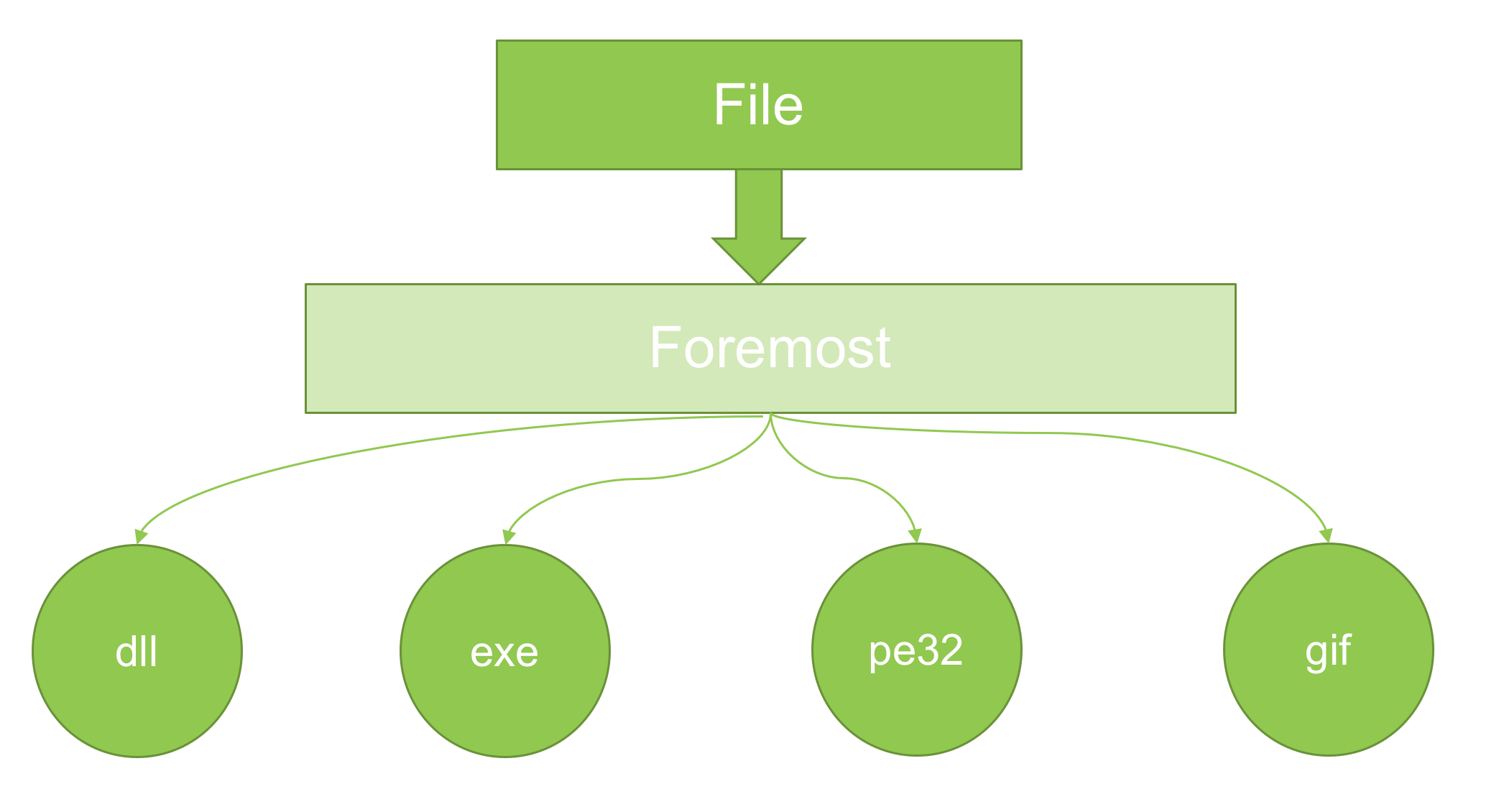
## Extracting Files

Yalda supports a large variety of file types. It walks through the given directory and subdirectories and extracts all compressed, embedded or even the attachments in emails and analyze each of them in detail. Following diagram shows how Yalda extracts files:



### Extracting Embedded Objects

Foremost is used to extract embedded objects. The extracted objects also get analyzed in detail by Yalda and collected information gets inserted in the database. Following diagram displays the method of extracting objects through each file:

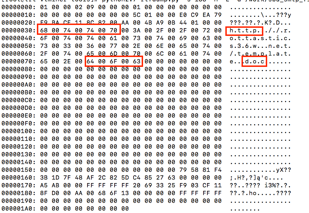


## Scanning & Analyzing Files

### Decoders

Yalda includes pre-defined decoders that is written for a malware family or a malware campaign. Based on the file type, a group of decoders get applied to the file and the file get flagged as malicious, suspicious or clear. Based on the match, data such as URLs and domains may also get extracted.

As the new malware families get introduced, more decoders would be added to Yalda. An example is the recently added CVE-2017-0199 decoder. Manual analysis of infected rtf file and embedded OLE object shows that the file is downloading a file from an embedded URL in the document:



The common section in all rtf files of this campaign is the same and the files could be determined through following code and the URL get extracted through defined regex:

pattern = re.compile('68007400740070[\w\d]{,400}64006f0063')

result = pattern.search(string)

if result:

link = ''.join(binascii.unhexlify(str(result.group(0))).split('\x00'))

**YARA Rules**

If the user selects to apply YARA rules, the rules should be placed in the directory defined in the config file. If there is a match with one or more rules, the name of the matched function would be included in the result.

### Yalda Scoring

The next step in decoding process is applying Yalda scoring on the file. The algorithm in Yalda scoring is based on clustering data obtained from malicious files. As Yalda analyzes more files, it gets stronger in detecting malicious files due to building a bigger cluster of malicious data. Yalda scoring is done through two methods: strings clustering and combination of pe-sections names and Shannon entropy clustering in executable files

##### Strings Clustering

Strings clustering is used for clustering malicious data. It applies strings on files and getting list of strings in each file. If the file is detected malicious through decoders or YARA rules analysis, the string list get converted to a SHA1 hash and get clustered in a separate collection.

When analyzing a file, the list of strings in the file get compared with the list of clustered strings and if number of common strings is more than a threshold excluding the whitelisted strings, the file gets marked suspicious with a severity from 1 to 5 depending on level of severity of the file determined through the algorithm.

##### PE Sections &Shannon Entropy\*[[1]](#footnote-1)

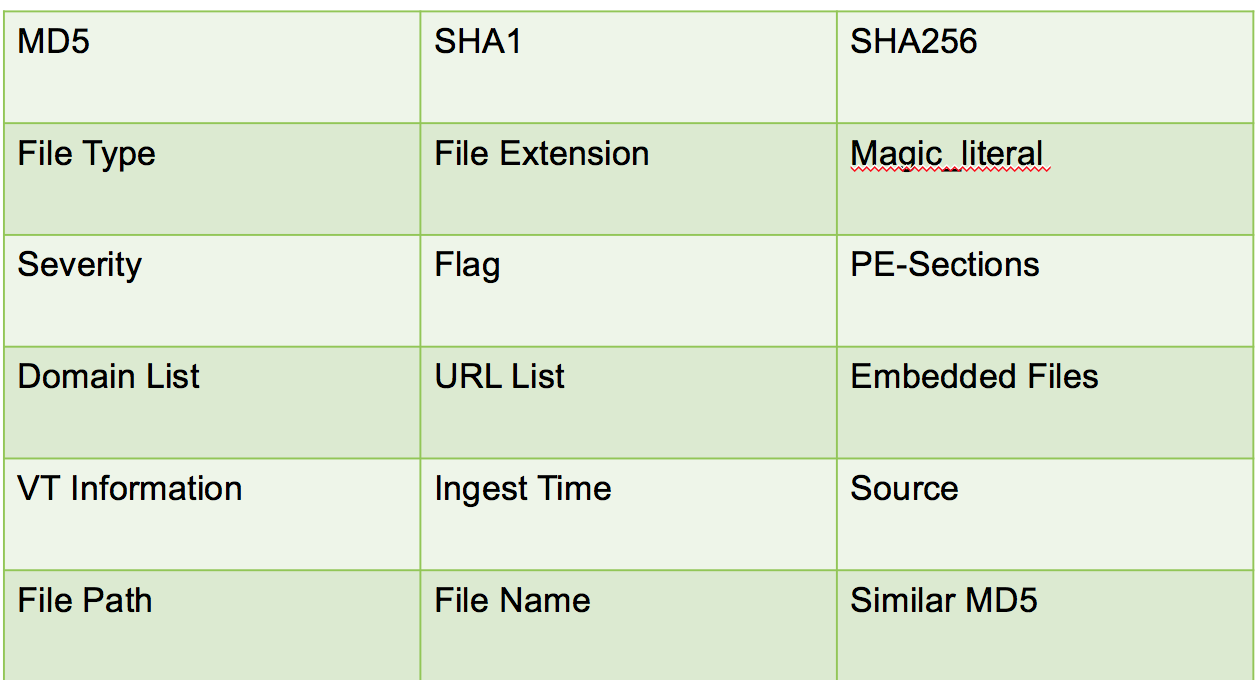
Names of PE-sections and Shannon entropy of malicious files get clustered in a separate collection. The section names get converted to SHA1 and get clustered in a separate collection in combination with Shannon entropy of the original name.

When analyzing a file, the list of pe-sections name and Shannon entropy get compared with the list of clustered data of malicious files and if number of matches is greater than a predefined threshold, it gets marked suspicious with a severity of 1 to 5 depending on the level of severity of the file determined through the algorithm.

**Techniques for further Analysis**

Multiple techniques are applied on the file for extracting more information such as MD5, SHA1, SHA256, detailed information of PE-sections, list of embedded objects, list of parent files with their MD5 if any, list of similar top ten clustered malicious files, file size, file type, magical literal and etc.

The following table shows the indicators that get extracted from each file:



## Flag and Severity

Based on the result of the applied scanning and analyzing phase and extracted information from files, each file get evaluated and get flagged as malicious, suspicious or clear. Also, a severity of 1 to 5 get assigned to the file based on level of maliciousness of the file. Severity 1 stands for clear and 5 stands for very malicious.

## Whitelisting

Whitelisting in Yalda is done through two different approaches: auto generation whitelisting and predefined logs. The algorithm defined for auto generation whitelisting is based on clustering strings of files detected clear by Yalda. The list of strings in the files get converted to SHA1 and get clustered in a different collection to be used as a whitelisting filter in Yalda scoring. Predefined logs containing hashes, URLs, domains data also get used in whitelisting. The predefined whitelist could also get customizes by users based on their field’s needs.

## Sample of Output

Following is an example of obtained results from analyzing a file:

{'SHA1': 'd279069217528344bf47076bab2ae86071605155', 'Magic\_literal': 'ASCII text, with very long lines, with no line terminators', 'Severity': 5, 'VT\_Info': {'positives': None, 'paramalink': None, 'vt\_exist': False}, 'File\_Type': 'WSF', 'File\_Name': 'Delivery-Receipt-02496750.doc.wsf', 'embedded\_files': [], 'Source': 'yalda\_mining\_data', 'SHA256': 'f6b21cd3a131a800cd99afb7a9106a22cfbc64e3c9273b9bf464e9f8be507989', 'MD5': 'bbe07815b10f90bea8b2c0d9f52384da', 'File\_Type\_Extension': 'wsf', '\_id': ObjectId('59720d3fde55594501c54b84'), 'Similar\_MD5': [], 'IngestTime': '2017-07-21T10:18:39.664388', 'Flag': 'malicious', 'Yara\_Attr': [], 'Domain\_lst': ['mercadoatlantico.com.br', 'www.linguaeworld.it', 'med-lex.com', 'instalaciondeairesplit.com', 'marklaapage.com'], 'PE\_sections': [], 'File\_Path': '/work/samples\_mime/json\_2016-12-20\_13-45\_1\_Delivery-Receipt-02496750/Delivery-Receipt-02496750.doc.wsf', 'Size': 621}

**Yalda Requirements**

Yalda runs on Linux environments and it has been tested with python 2.7. You would need mongodb to save the collected data in database. You would also need to customize config file, which is described in detail in “How to Use Yalda” section. The following python modules need to be installed prior running the script:

|  |  |  |  |
| --- | --- | --- | --- |
| Magic | json | email | Mimetypes |
| Globe | mailbox | Base64 | binascii |
| Pymongo | Crypto | Pefile | Yara |

## Yalda at Fidelis Cybersecurity GitHub

Yalda is available to download for free from Fidelis Cybersecurity github:

<https://github.com/fideliscyber/yalda>

## How to Use Yalda

Yalda is an automated tool that returns detailed scan results on the file with minimum false positives. The end user would need to download the tool from Fidelis Cybersecurity Github and install the required python modules.

Yalda comes in three folders: bin, src and yara\_dir. bin contains the main modules being used in the tool. Src contains two files: config.py and yalda\_file\_analyzer.py. yara\_dir contains a sample yara rule. Users would need to place yara rules compatible with yara command to be able to use this feature.

To configure Yalda, the end user would need to navigate to src folder and configure config file with following information:

* bin\_dir = <Specify full path of bin directory>
* data\_dir = <Specify full path of the directory that you would place files/subdirectories>
* clean\_up\_mime\_directory = <set it to 1 if you would like to remove extracted files in the mime\_attachment\_directory directory or 0 if you would like to keep the files.
* mime\_attachment\_directory = <specify full path of the directory that extracted/downloaded files would be placed>
* mongodb config:
  + local = <IP address of mongodb>
  + port = <Port number to connect to mongodb>
  + db\_name = by default the database name is defined amfm\_db
  + collection\_name = by default the main collection name is defined yalda\_collection
* yara\_check = <1 = enabled, 0= disabled> enable it if you would like to run yara rules on the samples. Please note that the format of yara rules should be compatible with yara command.
* yara\_rules\_dir = <Indicate full path of the yara rule directory>
* vt\_check = <1= enabled, 0 = disabled> enable it if you would like the sample be checked in Virus total and include a section of virus total such as number of AV engines detecting the sample, paramalink and if the file exists in virus total.
* vt\_key = <place your VT key in this section>

## License

Yalda is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation, either version 3 of the License, or any later version.

Yalda is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU General Public License for more details. You should have received a copy of the GNU General Public License along with Yalda.  If not, see <http://www.gnu.org/licenses/>.

## Conclusion

Yalda is an automated bulk intelligence tool and could be a useful tool for anyone who is interested in getting clustered data and a detailed scan of the large scaled data. It saves hours of manual analyses on each file and results could be used based on individual needs an interest is domains such as source of feeds, intelligence clustered data, feeding cuckoo sandbox, testing tool, source for writing YARA rules based on clustered strings in malicious files, etc.

1. Stay tuned for this feature! [↑](#footnote-ref-1)