Build Automated Malware Lab with CERT.PL Open Source Tools

Paweł Srokosz
Paweł Pawliński

CERT.PL>_

34th Annual FIRST Conference
27th June 2022, Dublin

Automated malware lab - why?

CERT.PL: who are we

- Established in 1996
- National CERT role formalized in the cybersecurity law in 2018
- Constituency: everything in Poland (*)

  (*) except government, military, critical infrastructure
- Part of NASK (research institute & .pl registry)

We are in threat intelligence business

- Monitoring threats to millions of users
- Malware incidents: 2nd most common (after phishing)
- We want to:
    - detect malware campaigns
    - warn potential victims
    - mitigate

  as early as possible

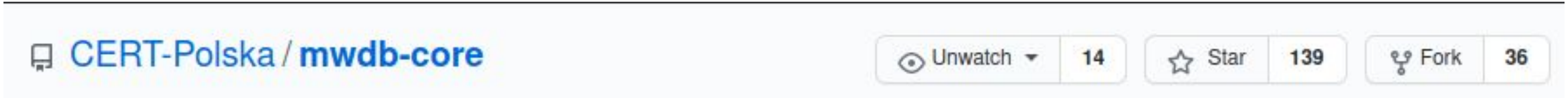
Evolution of our malware tooling

- Initially: tools developed case-by-case
- Early 2010s: rise of the banking trojans
- Mid 2010s: first automated malware analysis pipeline
- Late 2010s: live tracking of multiple botnets
- 2020s: era of open source analysis tools

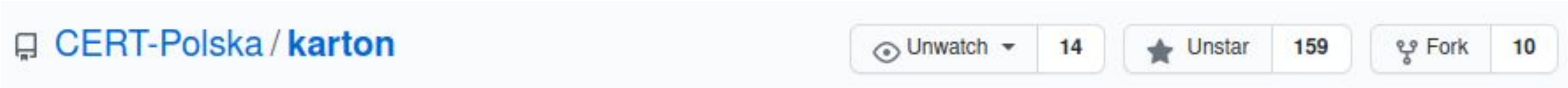
Basic ingredients of malware analysis lab

- **Collect:** repository to collect and search samples, IoCs, etc. from various sources (internal and external)
- **Analyze:** framework to integrate analytical tools focused on specific threats
- **Share:** provide threat intelligence to constituents / peers / customers

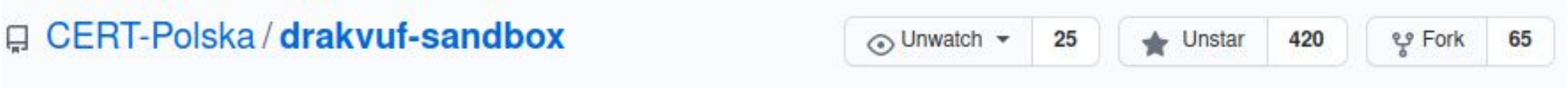
Main components of our lab



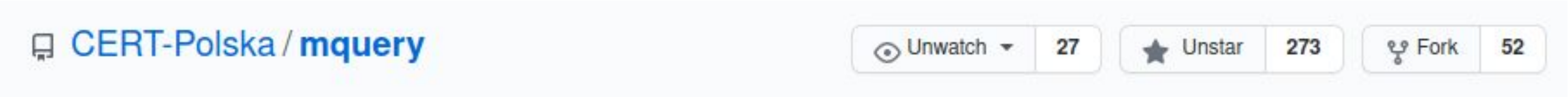
CERT-Polska / **mwdb-core**



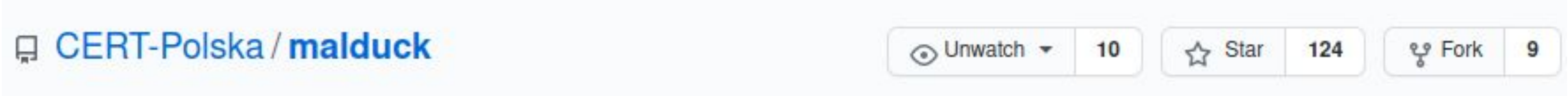
CERT-Polska / **karton**



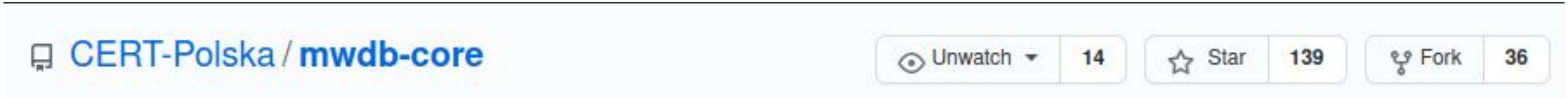
CERT-Polska / **drakvuf-sandbox**



CERT-Polska / **mquery**



CERT-Polska / **malduck**



CERT-Polska / mwdb-core

Collect: MWDB Core

What is MWDB Core?

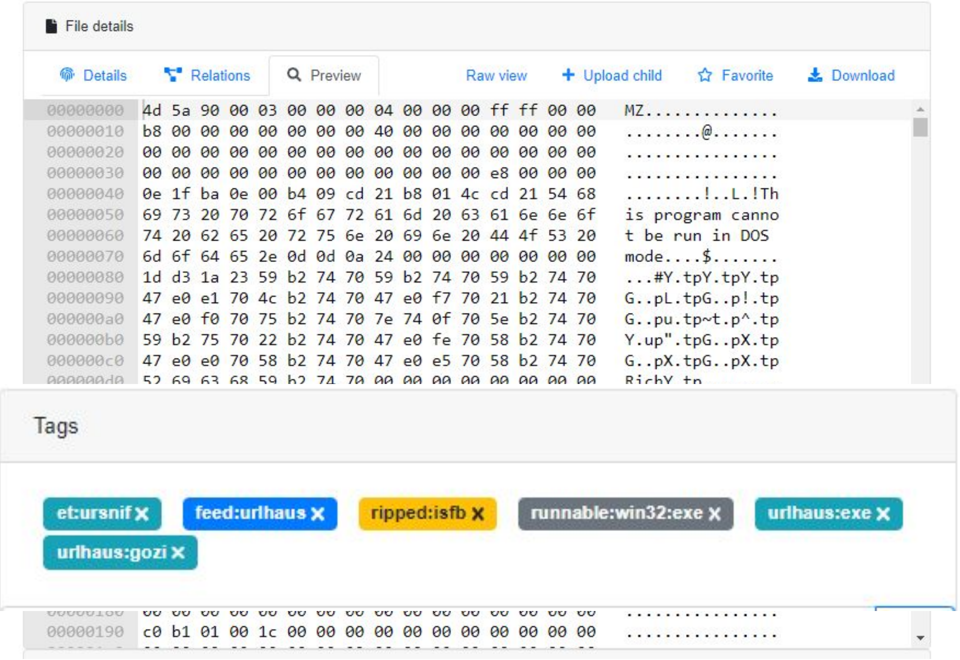
- Central component of our lab
- Repository for organizing and sharing malware intelligence
- Open-source
- Easy integration with other tools:
  - plugins
  - Karton
- Supported by CERT.PL and (small) community

MWDB Data model

- MWDB is made by analysts for analysts
- Not really a general purpose threat information sharing system
- Three basic object types:
    - Files
    - Configurations
    - Blobs
- Structured metadata for all objects

MWDB: Files

- The most basic object type
- Tags: file type, source, classification, …
- Attributes: source URL, Yara matches, AV detection, …



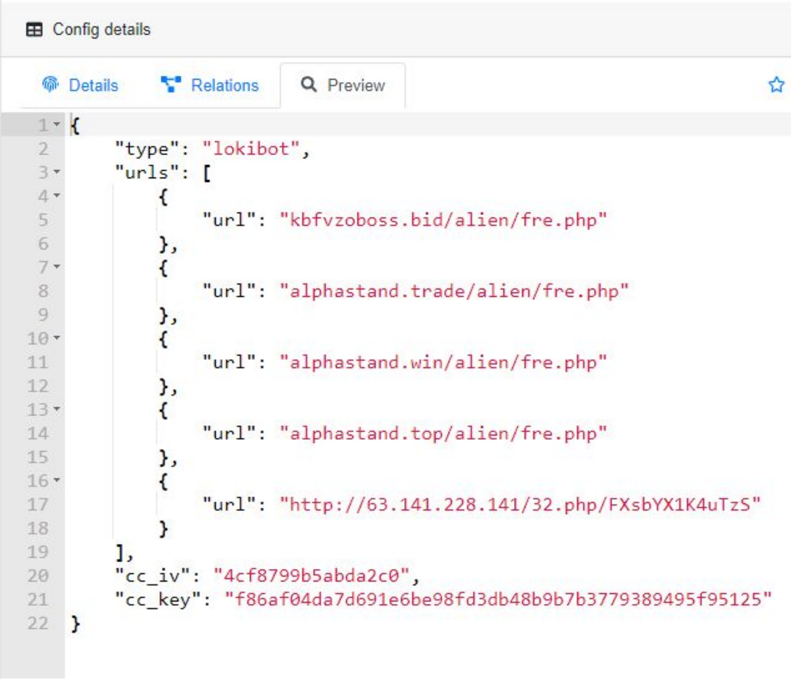
```
00000000 4d 5a 90 00 03 00 00 00 04 00 00 00 ff ff 00 00   MZ..............
00000010 b8 00 00 00 00 00 00 00 40 00 00 00 00 00 00 00   ........@.......
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00000030 00 00 00 00 00 00 00 00 00 00 00 00 e8 00 00 00   ................
00000040 0e 1f ba 0e 00 b4 09 cd 21 b8 01 4c cd 21 54 68   ........!..L.!Th
00000050 69 73 20 70 72 6f 67 72 61 6d 20 63 61 6e 6e 6f   is program canno
00000060 74 20 62 65 20 72 75 6e 20 69 6e 20 44 4f 53 20   t be run in DOS 
00000070 6d 6f 64 65 2e 0d 0d 0a 24 00 00 00 00 00 00 00   mode....$.......
00000080 1d d3 1a 23 59 b2 74 70 59 b2 74 70 59 b2 74 70   ...#Y.tpY.tpY.tp
00000090 47 e0 e1 70 4c b2 74 70 47 e0 f7 70 21 b2 74 70   G..pL.tpG..p!.tp
000000a0 47 e0 f0 70 75 b2 74 70 7e 74 0f 70 5e b2 74 70   G..pu.tp~t.p^.tp
000000b0 59 b2 75 70 22 b2 74 70 47 e0 fe 70 58 b2 74 70   Y.up".tpG..pX.tp
000000c0 47 e0 e0 70 58 b2 74 70 47 e0 e5 70 58 b2 74 70   G..pX.tpG..pX.tp
000000d0 52 69 63 68 59 b2 74 70 00 00 00 00 00 00 00 00   RichY.tp........
```

et:ursnif ✕ feed:urlhaus ✕ ripped:isfb ✕ runnable:win32:exe ✕ urlhaus:exe ✕ urlhaus:gozi ✕

```
00000180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00   ................
00000190 c0 b1 01 00 1c 00 00 00 00 00 00 00 00 00 00 00   ................
```

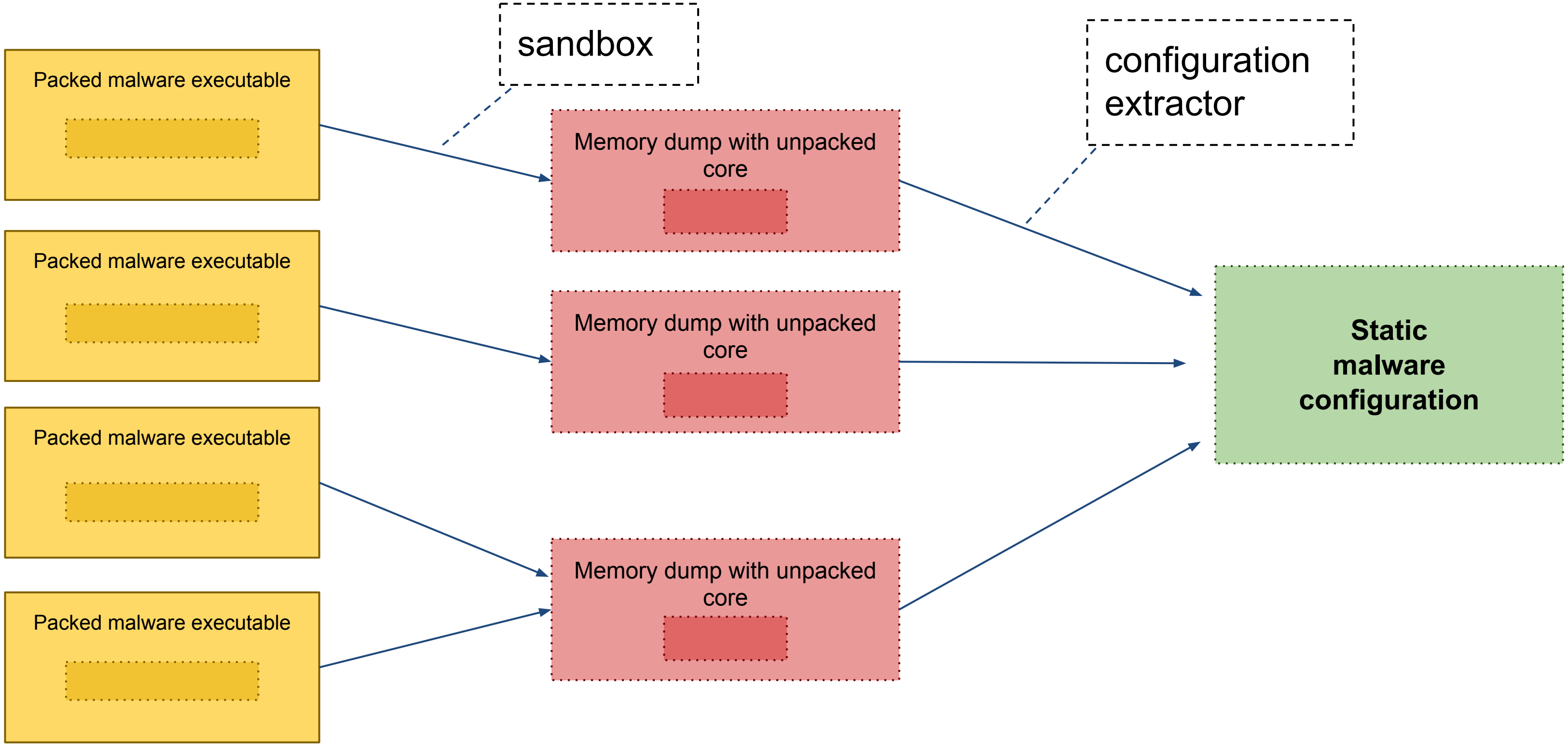
MWDB: Configurations

- Embedded in binary (static)
- Downloaded from C2 (dynamic)
- JSON
- Well-defined keys per malware family
- Structure determined by internal configuration format
- **End-goal of a typical malware analysis task** (automated by us for families of interest)



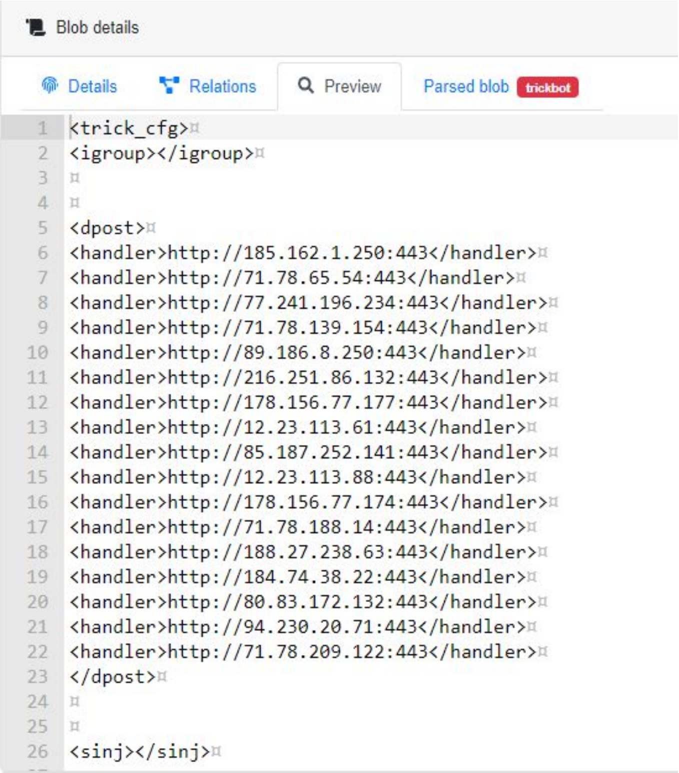
```
{
    "type": "lokibot",
    "urls": [
        {
            "url": "kbfvzoboss.bid/alien/fre.php"
        },
        {
            "url": "alphastand.trade/alien/fre.php"
        },
        {
            "url": "alphastand.win/alien/fre.php"
        },
        {
            "url": "alphastand.top/alien/fre.php"
        },
        {
            "url": "http://63.141.228.141/32.php/FXsbYX1K4uTzS"
        }
    ],
    "cc_iv": "4cf8799b5abda2c0",
    "cc_key": "f86af04da7d691e6be98fd3db48b9b7b3779389495f95125"
}
```

Basic processing pipeline


sandbox
configuration extractor
Packed malware executable
Packed malware executable
Packed malware executable
Packed malware executable
Memory dump with unpacked core
Memory dump with unpacked core
Memory dump with unpacked core
Static malware configuration

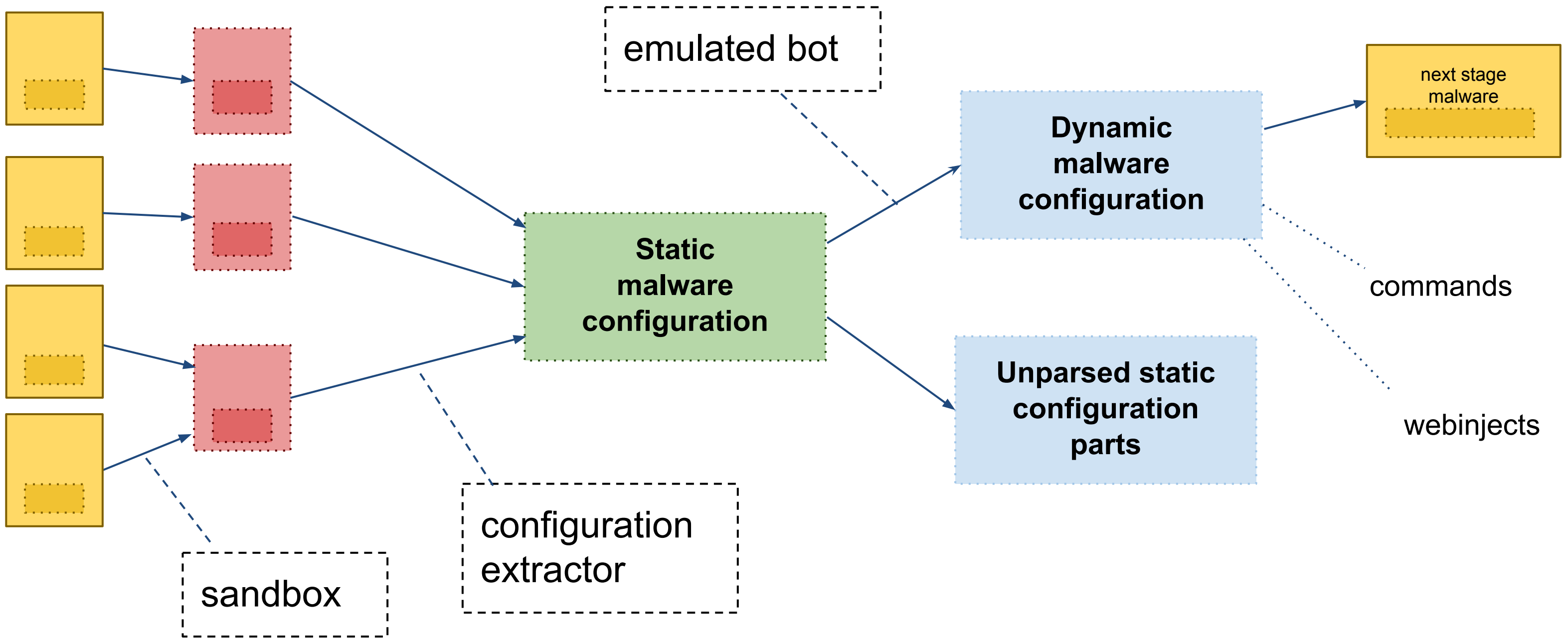
MWDB: Blobs

- Unstructured
- Decrypted data, webinjects, commands, lists of peers, …
- Stored for later processing or human inspection
- Full-text search



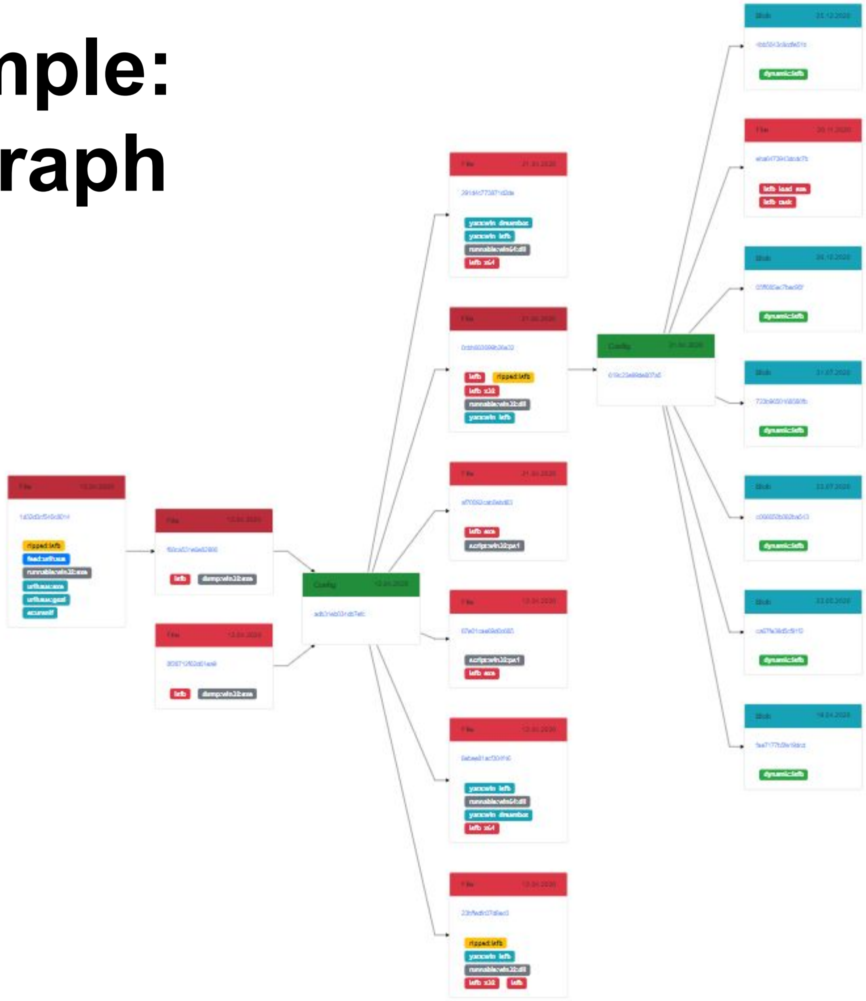
```
<trick_cfg>
<igroup></igroup>


<dpost>
<handler>http://185.162.1.250:443</handler>
<handler>http://71.78.65.54:443</handler>
<handler>http://77.241.196.234:443</handler>
<handler>http://71.78.139.154:443</handler>
<handler>http://89.186.8.250:443</handler>
<handler>http://216.251.86.132:443</handler>
<handler>http://178.156.77.177:443</handler>
<handler>http://12.23.113.61:443</handler>
<handler>http://85.187.252.141:443</handler>
<handler>http://12.23.113.88:443</handler>
<handler>http://178.156.77.174:443</handler>
<handler>http://71.78.188.14:443</handler>
<handler>http://188.27.238.63:443</handler>
<handler>http://184.74.38.22:443</handler>
<handler>http://80.83.172.132:443</handler>
<handler>http://94.230.20.71:443</handler>
<handler>http://71.78.209.122:443</handler>
</dpost>


<sinj></sinj>
```

Pipeline for botnet monitoring



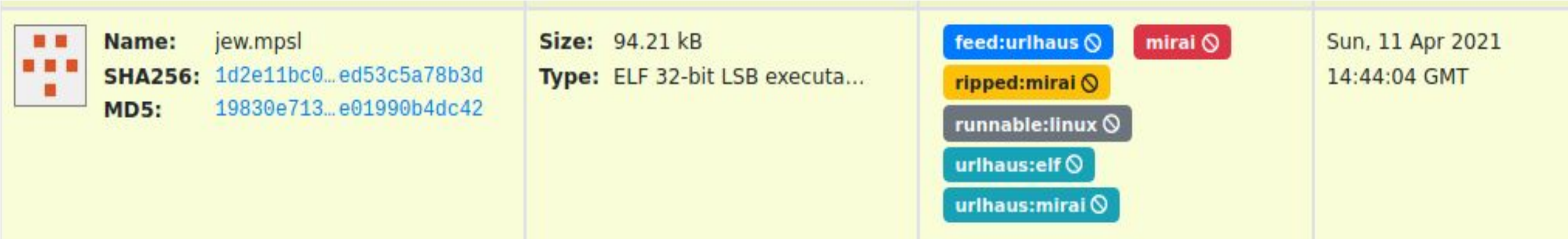
emulated bot
next stage malware
Dynamic malware configuration
Static malware configuration
commands
Unparsed static configuration parts
webinjects
sandbox
configuration extractor

Real-life example: ISFB (Gozi) graph



CERT.PL>_

Metadata: tags



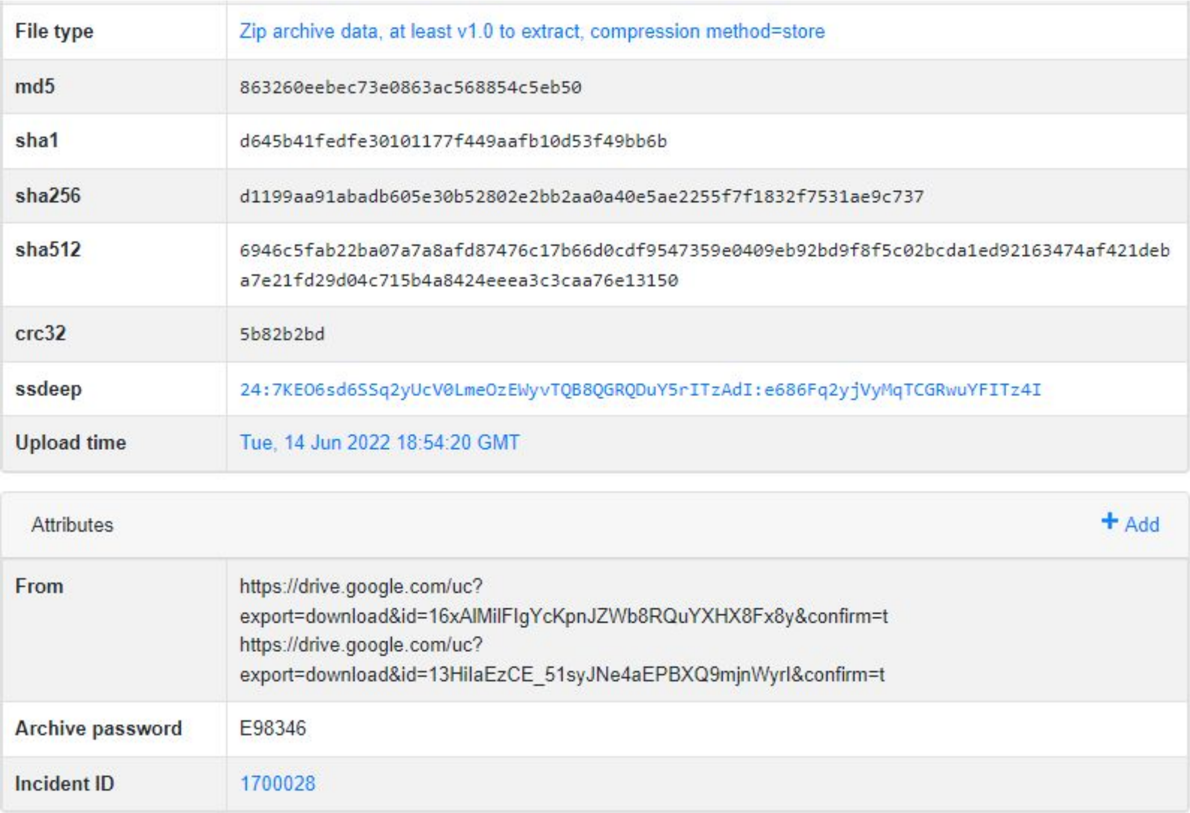
Name: jew.mpsl
SHA256: 1d2e11bc0…ed53c5a78b3d
MD5: 19830e713…e01990b4dc42

Size: 94.21 kB
Type: ELF 32-bit LSB executa...

feud:urlhaus ⃠ mirai ⃠
ripped:mirai ⃠
runnable:linux ⃠
urlhaus:elf ⃠
urlhaus:mirai ⃠

Sun, 11 Apr 2021
14:44:04 GMT

Metadata: attributes

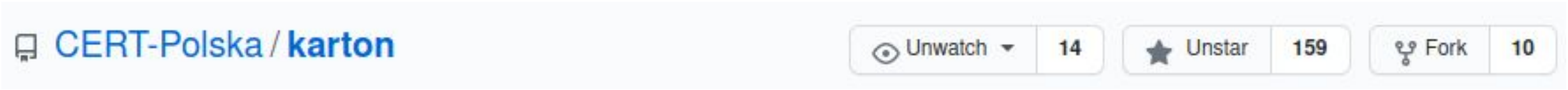


File type	Zip archive data, at least v1.0 to extract, compression method=store
md5	863260eebec73e0863ac568854c5eb50
sha1	d645b41fedfe30101177f449aafb10d53f49bb6b
sha256	d1199aa91abadb605e30b52802e2bb2aa0a40e5ae2255f7f1832f7531ae9c737
sha512	6946c5fab22ba07a7a8afd87476c17b66d0cdf9547359e0409eb92bd9f8f5c02bcda1ed92163474af421deb a7e21fd29d04c715b4a8424eeea3c3caa76e13150
crc32	5b82b2bd
ssdeep	24:7KEO6sd6SSq2yUcV0LmeOzEWyvTQB8QGRQDuY5rITzAdI:e686Fq2yjVyMqTCGRwuYFITz4I
Upload time	Tue, 14 Jun 2022 18:54:20 GMT

Attributes

From	https://drive.google.com/uc?
	export=download&id=16xAIMiIFIgYcKpnJZWb8RQuYXHX8Fx8y&confirm=t
	https://drive.google.com/uc?
	export=download&id=13HilaEzCE_51syJNe4aEPBXQ9mjnWyrl&confirm=t

Archive password	E98346
Incident ID	1700028



Analyze: Karton

Pareto rule

- **20% efforts, 80% effect**

writing an actual script to process a malware feed

- **80% efforts, 20% effect**

polling for data, queueing, integration with other scripts, logging, proper error handling, maintenance…

Pareto rule

- **20% efforts, 80% effect**

writing an actual script to process a malware feed

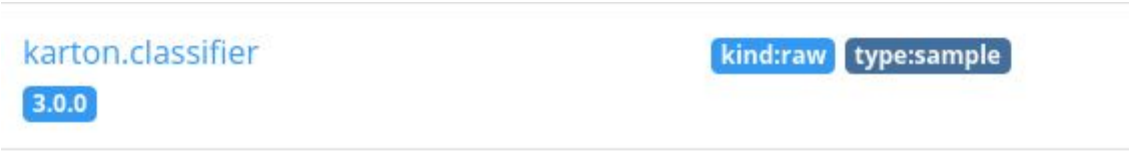
- **80% efforts, 20% effect**

(handle all of the common things with some common approach)

Karton design

Inspiration: Assembly Line by
Canadian Centre for Cyber Security

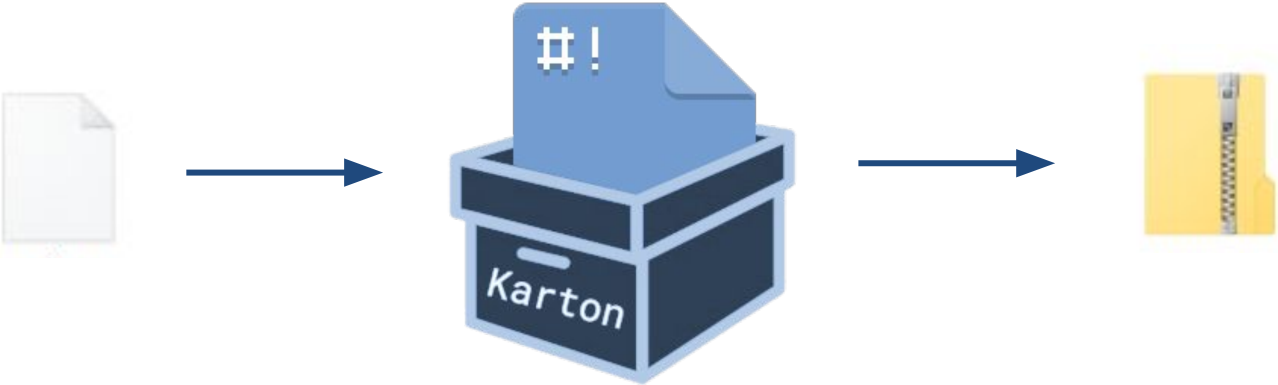
- Queue-based data processing pipelines
- Data-driven routing of tasks
- Lightweight
- Based on Redis (KV store) and S3-compatible object stores
- Built for microservices:
  - each processing module is focused on one task
  - “Plug and Play”, researcher should be able to easily add a new service
- Management interface

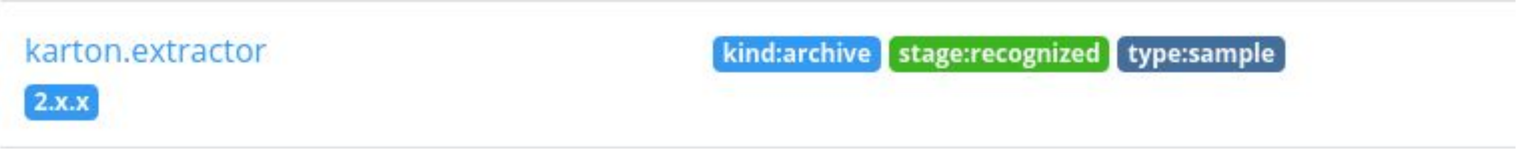


karton.classifier

3.0.0

kind:raw type:sample

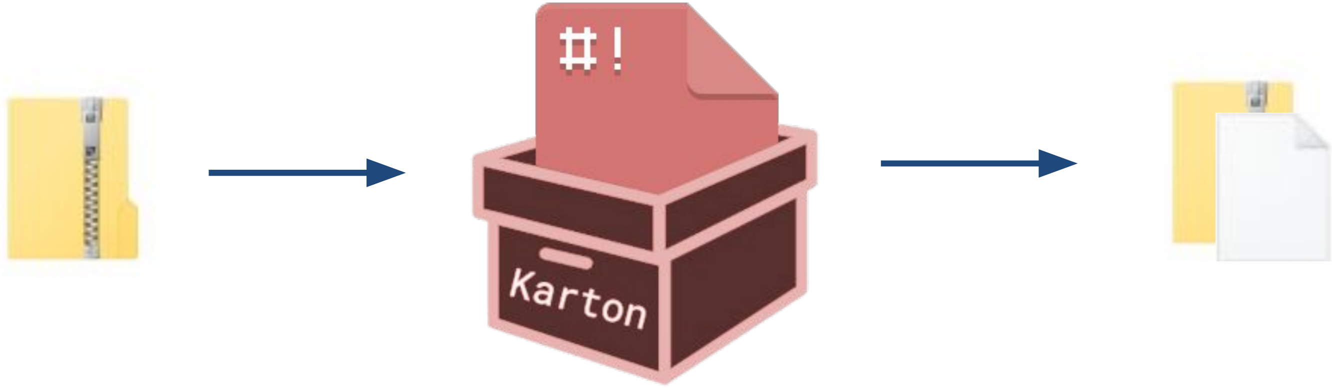




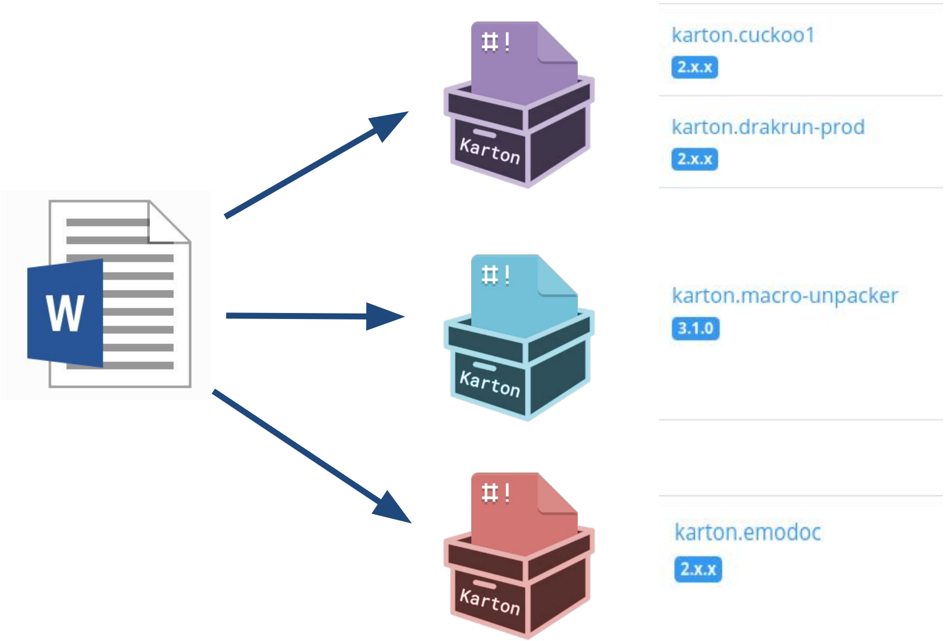
karton.extractor

2.x.x

kind:archive stage:recognized type:sample



Example: consumers of Office documents

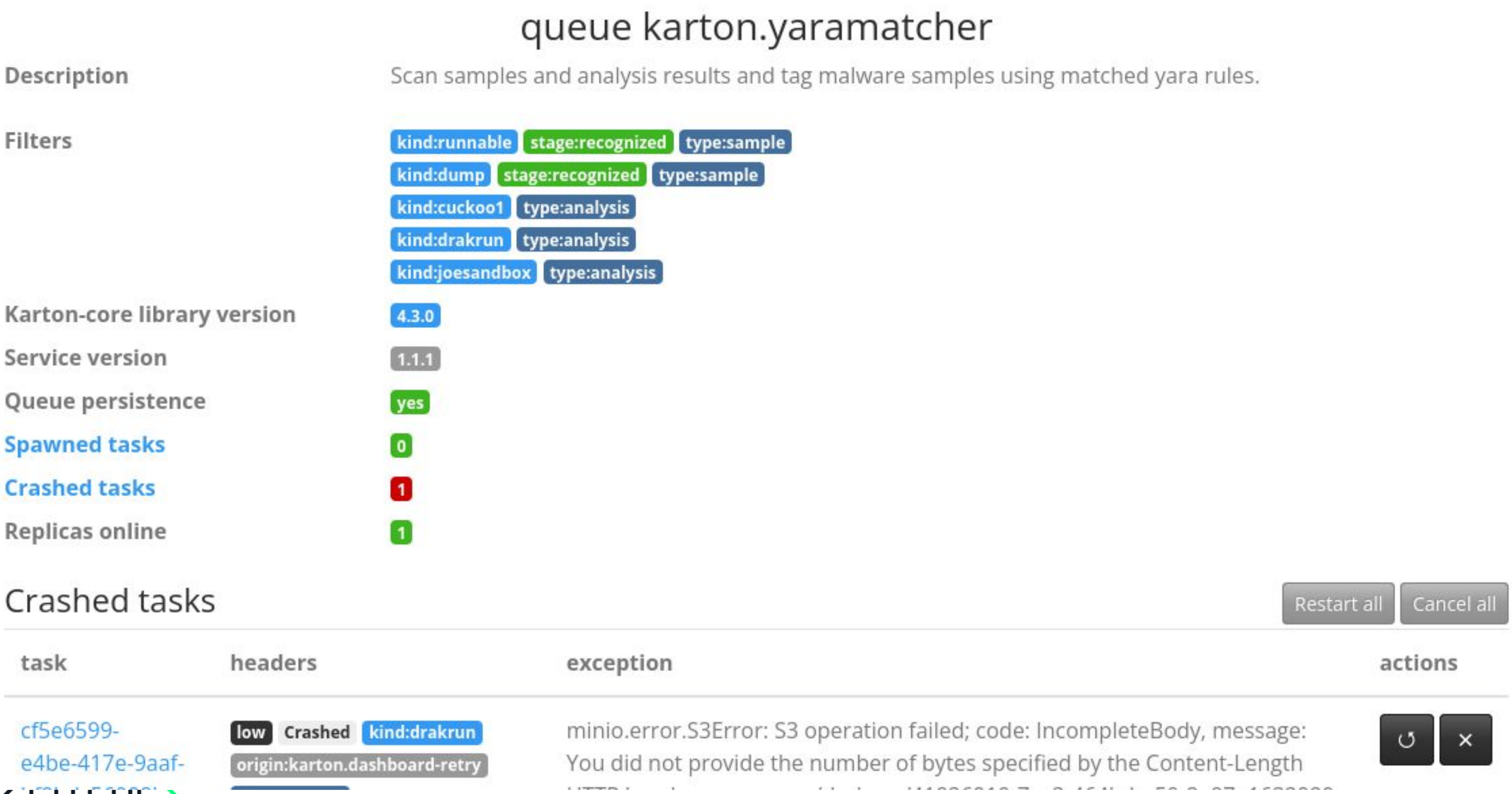


karton.cuckoo1
2.x.x

karton.drakrun-prod
2.x.x

karton.macro-unpacker
3.1.0

karton.emodoc
2.x.x



queue karton.yaramatcher

Description	Scan samples and analysis results and tag malware samples using matched yara rules.
Filters	kind:runnable stage:recognized type:sample kind:dump stage:recognized type:sample kind:cuckoo1 type:analysis kind:drakrun type:analysis kind:joesandbox type:analysis
Karton-core library version	4.3.0
Service version	1.1.1
Queue persistence	yes
Spawned tasks	0
Crashed tasks	1
Replicas online	1

Crashed tasks

task	headers	exception	actions
cf5e6599-e4be-417e-9aaf-	low Crashed kind:drakrun origin:karton.dashboard-retry	minio.error.S3Error: S3 operation failed; code: IncompleteBody, message: You did not provide the number of bytes specified by the Content-Length	↺ ✕

Share: mwdb.cert.pl

Providing threat intelligence

- Making our know-how & data available for defenders
- Access to our MWDB instance
  - samples
  - configurations
  - output of our private analyzers
- Free service: https://mwdb.cert.pl/
- Open registration + manual vetting

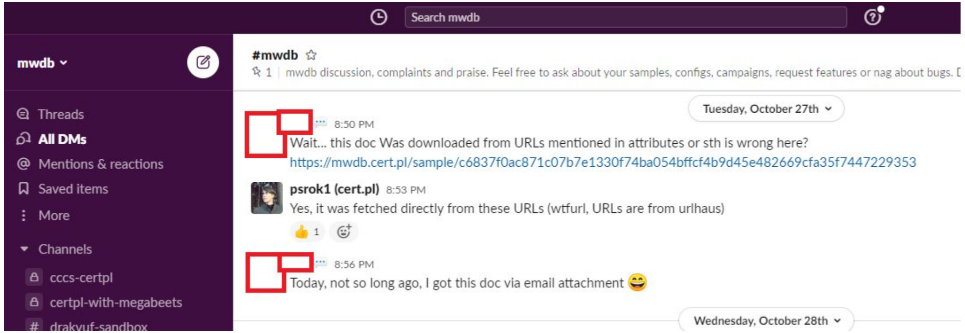
Statistics

- 1000+ accounts
- Extractors for 133 families (*)

    (*) not all work with current variants

- 2.4M+ samples
- 67k+ configurations
- 700/day avg new samples

Working with the community



#mwdb

1 | mwdb discussion, complaints and praise. Feel free to ask about your samples, configs, campaigns, request features or nag about bugs. [

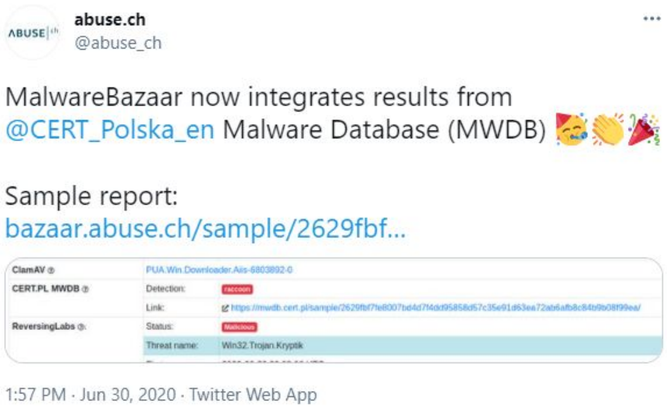
Tuesday, October 27th

[redacted] 8:50 PM
Wait... this doc Was downloaded from URLs mentioned in attributes or sth is wrong here?
https://mwdb.cert.pl/sample/c6837f0ac871c07b7e1330f74ba054bffcf4b9d45e482669cfa35f7447229353

psrok1 (cert.pl) 8:53 PM
Yes, it was fetched directly from these URLs (wtfurl, URLs are from urlhaus)
👍 1

[redacted] 8:56 PM
Today, not so long ago, I got this doc via email attachment 😄

Wednesday, October 28th



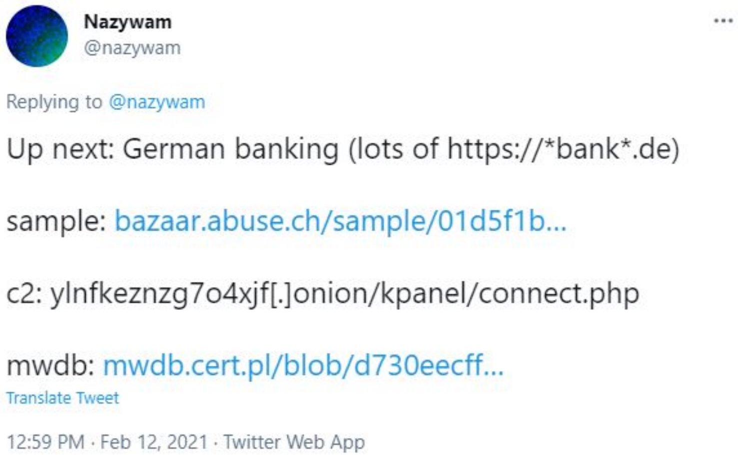
abuse.ch
@abuse_ch

MalwareBazaar now integrates results from @CERT_Polska_en Malware Database (MWDB) 🥳👏🎉

Sample report:
bazaar.abuse.ch/sample/2629fbf...

ClamAV	PUA.Win.Downloader.Aiis-6803892-0	
CERT.PL MWDB	Detection:	raccoon
	Link:	https://mwdb.cert.pl/sample/2629fbf7fe8007bd4d7f4dd95858d57c35e91d63ea72ab6afb8c84b9b08f99ea/
ReversingLabs	Status:	Malicious
	Threat name:	Win32.Trojan.Kryptik

1:57 PM · Jun 30, 2020 · Twitter Web App



Nazywam
@nazywam

Replying to @nazywam

Up next: German banking (lots of https://*bank*.de)

sample: bazaar.abuse.ch/sample/01d5f1b...

c2: ylnfkeznzg7o4xjf[.]onion/kpanel/connect.php

mwdb: mwdb.cert.pl/blob/d730eecff...

Translate Tweet

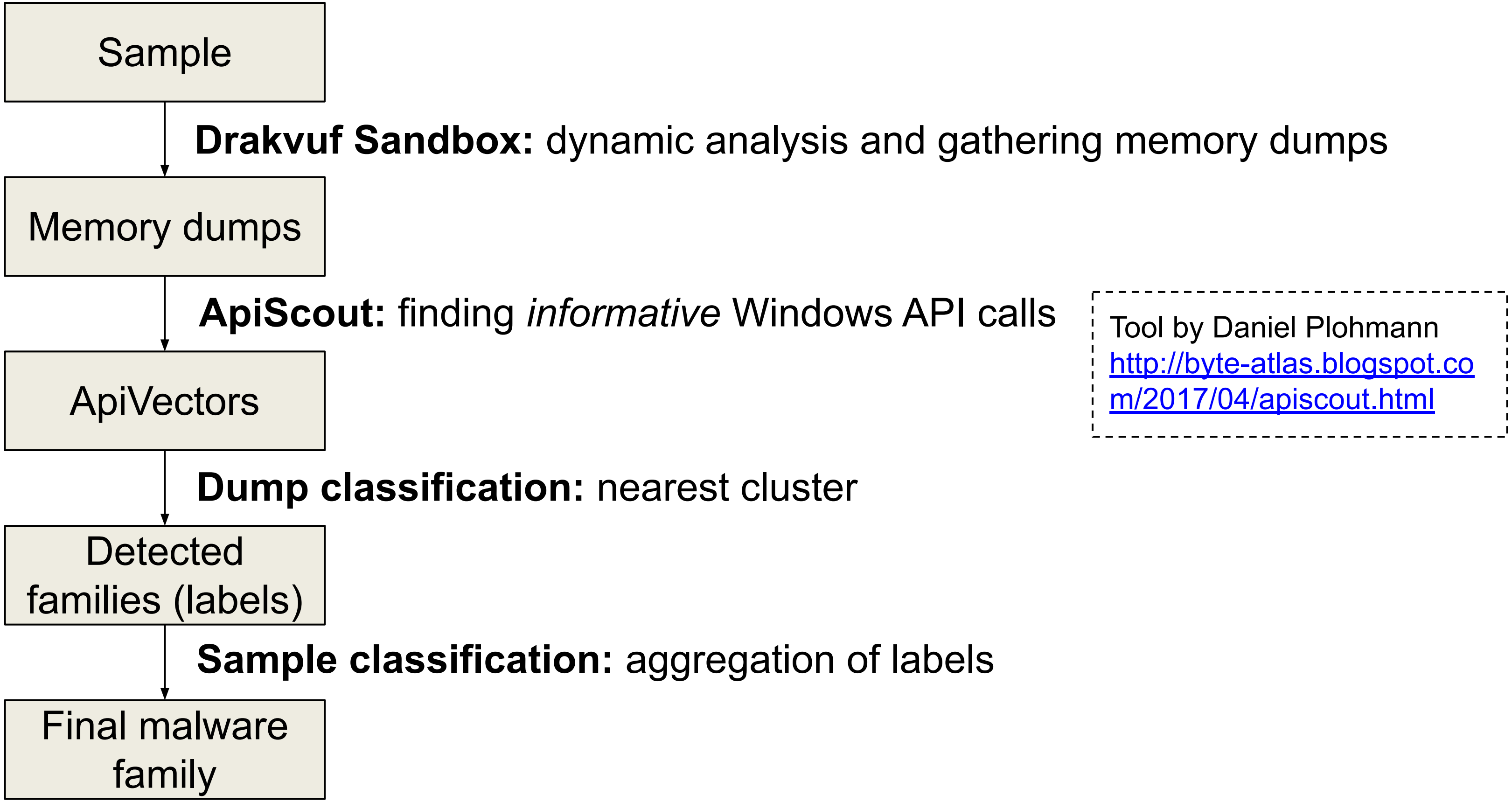
12:59 PM · Feb 12, 2021 · Twitter Web App

Plugin showcase: malware similarity

Finding similar samples

- Objectives:
  - classify malware family
  - discover clusters
- Can be used to detect new variants
- No reversing & development of analysis modules necessary
- Better understanding of the development of threats
- Common use case: support attribution

Using Windows API for classification



Sample

Drakvuf Sandbox: dynamic analysis and gathering memory dumps

Memory dumps

ApiScout: finding *informative* Windows API calls

Tool by Daniel Plohmann
http://byte-atlas.blogspot.com/2017/04/apiscout.html

ApiVectors

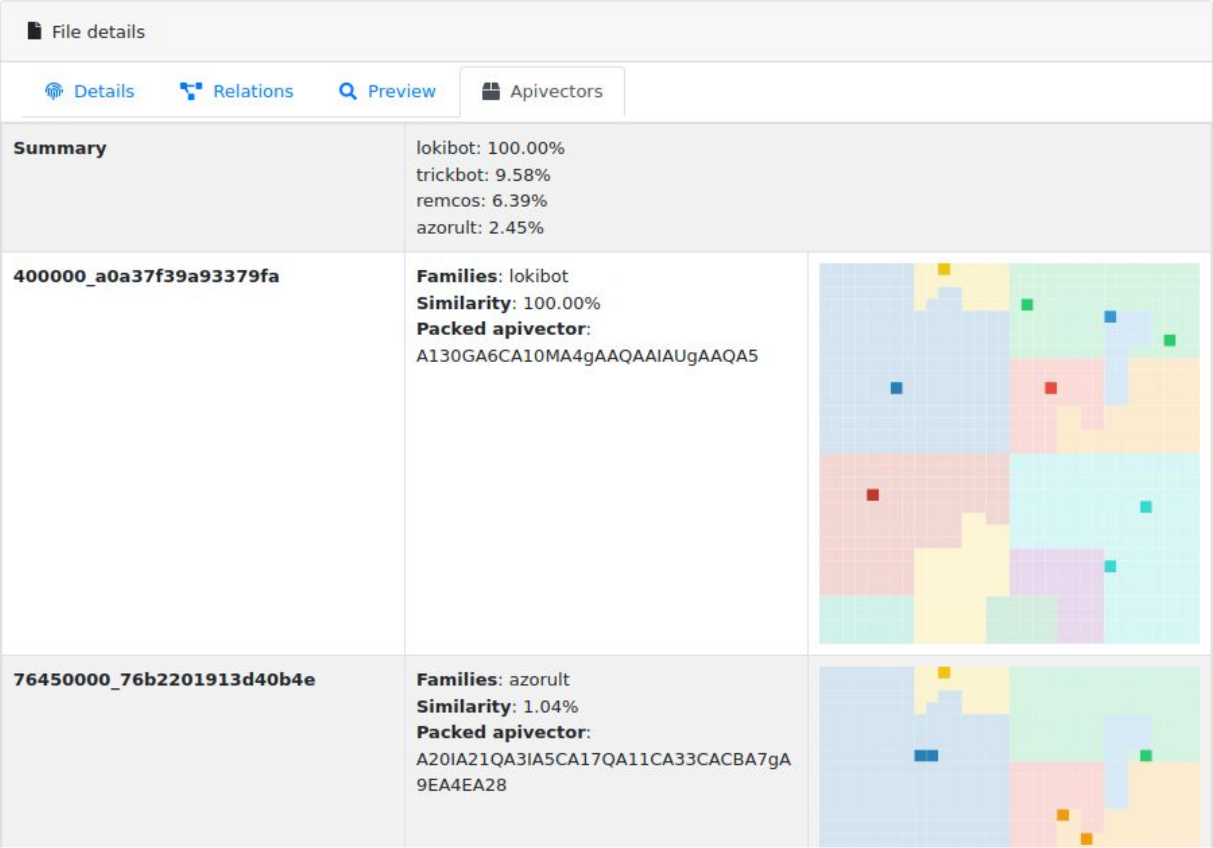
Dump classification: nearest cluster

Detected families (labels)

Sample classification: aggregation of labels

Final malware family

Classification results

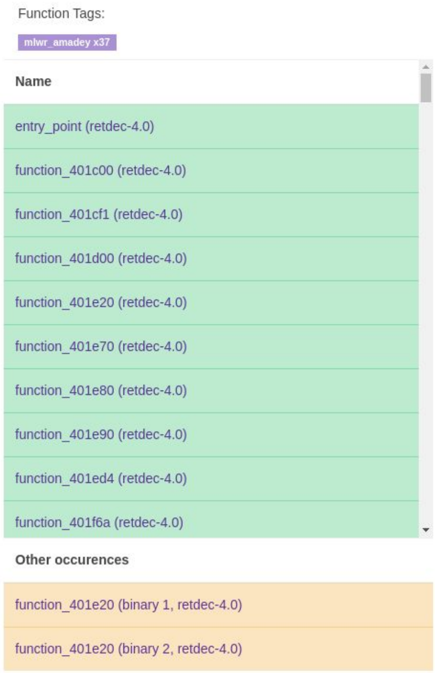


Summary	lokibot: 100.00% trickbot: 9.58% remcos: 6.39% azorult: 2.45%	
400000_a0a37f39a93379fa	**Families**: lokibot **Similarity**: 100.00% **Packed apivector**: A130GA6CA10MA4gAAQAAIAUgAAQA5	
76450000_76b2201913d40b4e	**Families**: azorult **Similarity**: 1.04% **Packed apivector**: A20IA21QA3IA5CA17QA11CA33CACBA7gA9EA4EA28	

Upcoming integration: msource

- Finding similar code in malware binaries
- Function-level comparison
- Flexible backend: currently multiple disassemblers
- Internal web interface for analysts and administrators
- PoC plugin for MWDB in 2021, improved version coming soon

msource: behind the scenes

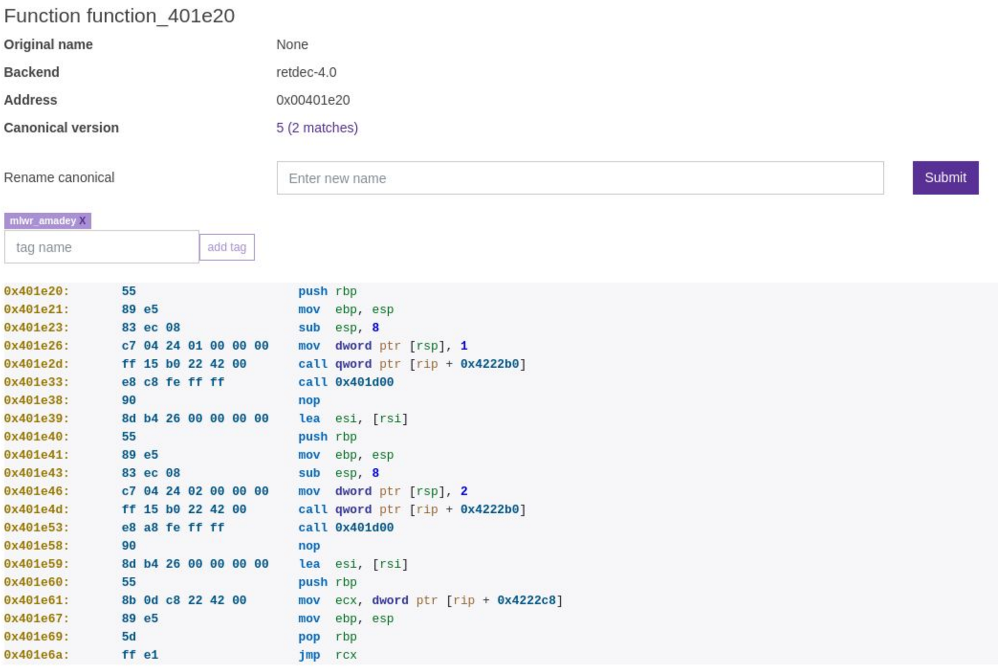


Function Tags:

mlwr_amadey x37

Name
entry_point (retdec-4.0)
function_401c00 (retdec-4.0)
function_401cf1 (retdec-4.0)
function_401d00 (retdec-4.0)
function_401e20 (retdec-4.0)
function_401e70 (retdec-4.0)
function_401e80 (retdec-4.0)
function_401e90 (retdec-4.0)
function_401ed4 (retdec-4.0)
function_401f6a (retdec-4.0)

Other occurences
function_401e20 (binary 1, retdec-4.0)
function_401e20 (binary 2, retdec-4.0)



Function function_401e20

Original name	None
Backend	retdec-4.0
Address	0x00401e20
Canonical version	5 (2 matches)

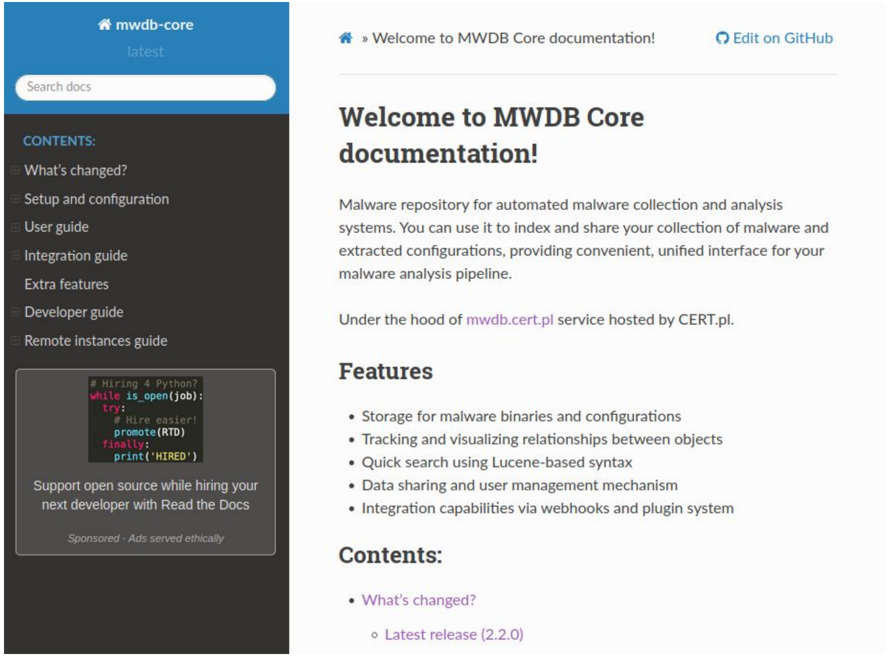
Rename canonical

```
0x401e20:    55                      push rbp
0x401e21:    89 e5                   mov  ebp, esp
0x401e23:    83 ec 08                sub  esp, 8
0x401e26:    c7 04 24 01 00 00 00    mov  dword ptr [rsp], 1
0x401e2d:    ff 15 b0 22 42 00       call qword ptr [rip + 0x4222b0]
0x401e33:    e8 c8 fe ff ff          call 0x401d00
0x401e38:    90                      nop
0x401e39:    8d b4 26 00 00 00 00    lea  esi, [rsi]
0x401e40:    55                      push rbp
0x401e41:    89 e5                   mov  ebp, esp
0x401e43:    83 ec 08                sub  esp, 8
0x401e46:    c7 04 24 02 00 00 00    mov  dword ptr [rsp], 2
0x401e4d:    ff 15 b0 22 42 00       call qword ptr [rip + 0x4222b0]
0x401e53:    e8 a8 fe ff ff          call 0x401d00
0x401e58:    90                      nop
0x401e59:    8d b4 26 00 00 00 00    lea  esi, [rsi]
0x401e60:    55                      push rbp
0x401e61:    8b 0d c8 22 42 00       mov  ecx, dword ptr [rip + 0x4222c8]
0x401e67:    89 e5                   mov  ebp, esp
0x401e69:    5d                      pop  rbp
0x401e6a:    ff e1                   jmp  rcx
```

How to get started

MWDB Core: official docs

https://mwdb.readthedocs.io/



Welcome to MWDB Core documentation!

Malware repository for automated malware collection and analysis systems. You can use it to index and share your collection of malware and extracted configurations, providing convenient, unified interface for your malware analysis pipeline.

Under the hood of mwdb.cert.pl service hosted by CERT.pl.

Features

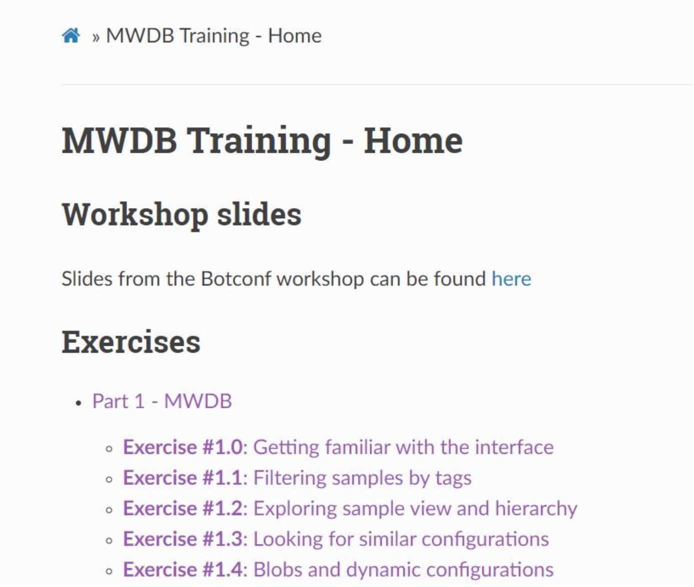
- Storage for malware binaries and configurations
- Tracking and visualizing relationships between objects
- Quick search using Lucene-based syntax
- Data sharing and user management mechanism
- Integration capabilities via webhooks and plugin system

Contents:

- What's changed?
  - Latest release (2.2.0)

Online training materials

https://training-mwdb.readthedocs.io/



» MWDB Training - Home

MWDB Training - Home

Workshop slides

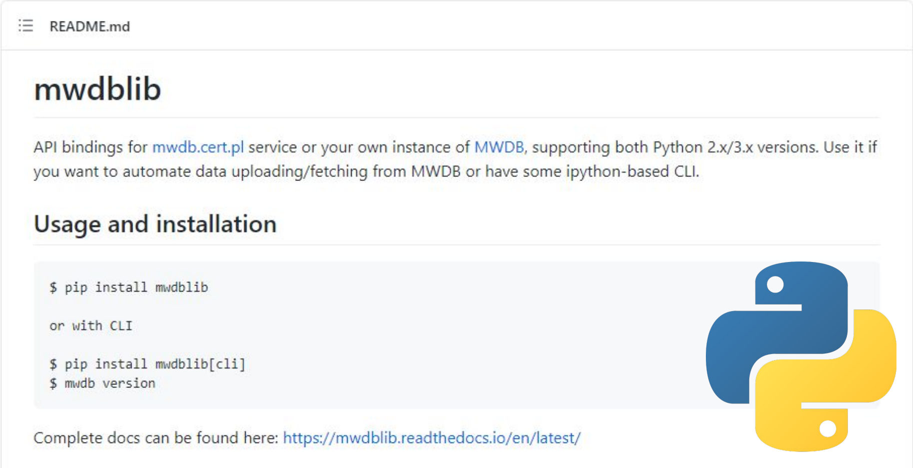
Slides from the Botconf workshop can be found here

Exercises

- Part 1 - MWDB
  - **Exercise #1.0**: Getting familiar with the interface
  - **Exercise #1.1**: Filtering samples by tags
  - **Exercise #1.2**: Exploring sample view and hierarchy
  - **Exercise #1.3**: Looking for similar configurations
  - **Exercise #1.4**: Blobs and dynamic configurations

mwdblib: automation library for MWDB

https://github.com/CERT-Polska/mwdblib



README.md

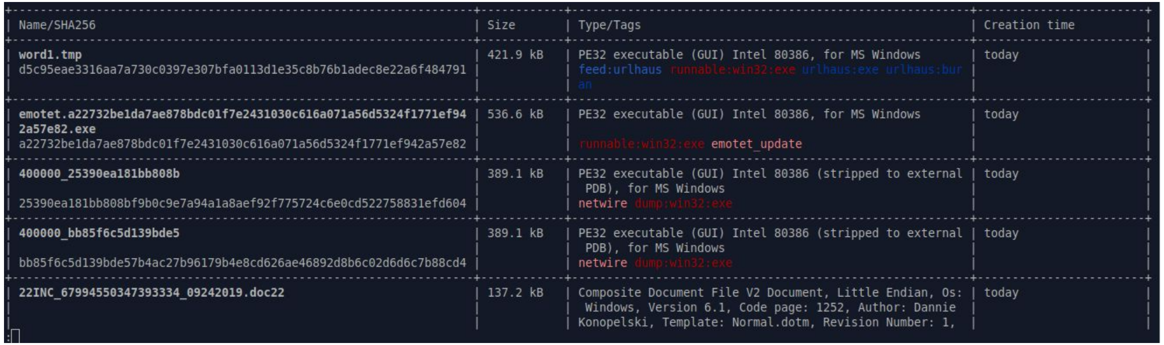
mwdblib

API bindings for mwdb.cert.pl service or your own instance of MWDB, supporting both Python 2.x/3.x versions. Use it if you want to automate data uploading/fetching from MWDB or have some ipython-based CLI.

Usage and installation

```
$ pip install mwdblib

or with CLI

$ pip install mwdblib[cli]
$ mwdb version
```

Complete docs can be found here: https://mwdblib.readthedocs.io/en/latest/



```
+------------------------------------------------------------------+------------+----------------------------------------------------------+----------------------+
| Name/SHA256                                                      | Size       | Type/Tags                                                | Creation time        |
+------------------------------------------------------------------+------------+----------------------------------------------------------+----------------------+
| word1.tmp                                                        | 421.9 kB   | PE32 executable (GUI) Intel 80386, for MS Windows        | today                |
| d5c95eae3316aa7a730c0397e307bfa0113d1e35c8b76b1adec8e22a6f484791 |            | feed:urlhaus runnable:win32:exe urlhaus:exe urlhaus:bur |                      |
|                                                                  |            | an                                                       |                      |
+------------------------------------------------------------------+------------+----------------------------------------------------------+----------------------+
| emotet.a22732be1da7ae878bdc01f7e2431030c616a071a56d5324f1771ef94 | 536.6 kB   | PE32 executable (GUI) Intel 80386, for MS Windows        | today                |
| 2a57e82.exe                                                      |            |                                                          |                      |
| a22732be1da7ae878bdc01f7e2431030c616a071a56d5324f1771ef942a57e82 |            | runnable:win32:exe emotet_update                         |                      |
+------------------------------------------------------------------+------------+----------------------------------------------------------+----------------------+
| 400000_25390ea181bb808b                                          | 389.1 kB   | PE32 executable (GUI) Intel 80386 (stripped to external | today                |
|                                                                  |            |  PDB), for MS Windows                                    |                      |
| 25390ea181bb808bf9b0c9e7a94a1a8aef92f775724c6e0cd522758831efd604 |            | netwire dump:win32:exe                                   |                      |
+------------------------------------------------------------------+------------+----------------------------------------------------------+----------------------+
| 400000_bb85f6c5d139bde5                                          | 389.1 kB   | PE32 executable (GUI) Intel 80386 (stripped to external | today                |
|                                                                  |            |  PDB), for MS Windows                                    |                      |
| bb85f6c5d139bde57b4ac27b96179b4e8cd626ae46892d8b6c02d6d6c7b88cd4 |            | netwire dump:win32:exe                                   |                      |
+------------------------------------------------------------------+------------+----------------------------------------------------------+----------------------+
| 22INC_679945503473933334_09242019.doc22                          | 137.2 kB   | Composite Document File V2 Document, Little Endian, Os: | today                |
|                                                                  |            |  Windows, Version 6.1, Code page: 1252, Author: Dannie  |                      |
|                                                                  |            | Konopelski, Template: Normal.dotm, Revision Number: 1,  |                      |
:
```

malduck: supports malware analysis

- Open-source configuration extractor engine, written in Python
- Collection of common algorithms and utilities for extracting data from binaries

github.com/CERT-Polska/malduck

README.md

🦆 Malduck

Installation ⚙️ | Docs 📚

SPARTA

Co-financed by the Connecting Europe Facility of the European Union

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 830892.

The contents of this presentation are the sole responsibility of CERT.PL / NASK and do not necessarily reflect the opinion of the European Union.

Contact:

pawel.srokosz@cert.pl
pawel.pawlinski@cert.pl
info@cert.pl

https://github.com/CERT-Polska/