

# Build Your Own Malware Analysis Pipeline Using New Open Source Tools

Paweł Srokosz

Jarosław Jedynak

Paweł Pawliński



FIRST Workshop Series

15th April 2021

# Agenda

- **mwdb.cert.pl**
  - What the heck is **MWDB**?
  - Tour de **mwdb.cert.pl**
  - Scripting and automation with **mwdblib**
- **mwdb-core** and **karton**
  - Run a **self-hosted** mwdb-core and karton instances
  - Experiment with **karton-playground**
  - Automated unpacking with **malduck**

# Prerequisites

Open a terminal and check if these tools are installed:

- `$ python3 -m pip`
- `$ git`
- `$ docker-compose`

<https://docs.docker.com/engine/install/ubuntu/>

<https://docs.docker.com/compose/install/>







# What the heck is MWDB?

## Introduction to the interface

CERT.PL > [Samples](#) [Configs](#) [Blobs](#) [Upload](#) [Yara search](#) [Search](#) [Groups](#) [Statistics](#) [About](#) Logged as: msm [Profile](#) [Logout](#)

[Search](#) [?](#)

Quick query: [Only uploaded by me](#) [Exclude public](#) [Favorites](#) [Exclude feed:\\*](#) [Only ripped:\\*](#) [Add +](#)

Name/Hash	Size/Type	Tags	First seen
 <b>Name:</b> mw.exe <b>SHA256:</b> baa6544042f59009f54e...245c59ea6345 <b>MD5:</b> 68a8b423155d6359627d6d837cb498e4	<b>Size:</b> 168.33 kB <b>Type:</b> ELF 32-bit MSB executable, MIPS, ...	<a href="#">runnable:linux</a>	Fri, 09 Apr 2021 20:10:24 GMT
 <b>Name:</b> 3715000_dcaa867147f9cb98 <b>SHA256:</b> dcaa867147f9cb98ac5a6...ba4ee425d6ab <b>MD5:</b> e99e2d754bf0d2119ff03ab19f32259	<b>Size:</b> 212.87 kB <b>Type:</b> SysEx File -	<a href="#">agenttesla</a> <a href="#">dump:win32:exe</a>	Fri, 09 Apr 2021 20:10:23 GMT
 <b>Name:</b> mw.exe <b>SHA256:</b> 826591a912f064d2dab8d...efefcdbe4df7 <b>MD5:</b> 1578c6fc3b83053227181b0edd79ead0	<b>Size:</b> 11.51 MB <b>Type:</b> PE32 executable (GUI) Intel 80386...	<a href="#">runnable:win32:exe</a>	Fri, 09 Apr 2021 20:08:35 GMT
 <b>Name:</b> 6da3196ca12bd21ec4bef0191960d55... <b>SHA256:</b> e4b7cf0c007f13c5fe3f3...902d198436cc <b>MD5:</b> 6ac19b1e6dc4007e3e2afe6c457493d8	<b>Size:</b> 541 kB <b>Type:</b> PE32 executable (GUI) Intel 80386...	<a href="#">feed:malwarebazaar</a> <a href="#">runnable:win32:exe</a>	Fri, 09 Apr 2021 20:07:48 GMT

mwdb.cert.pl



malware



malware

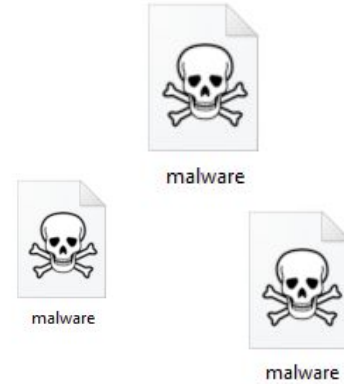


malware

mwdb.cert.pl

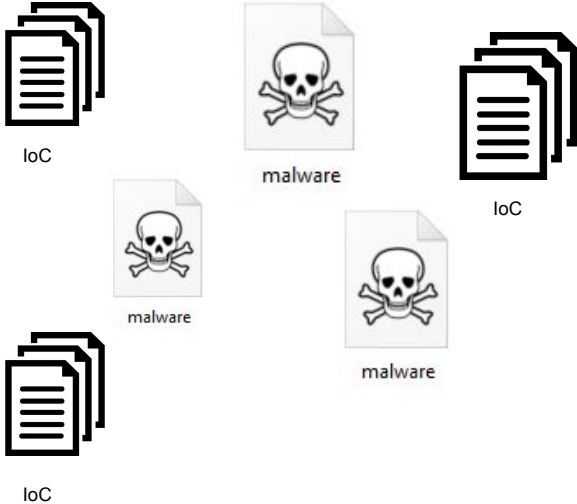


Malware reverse engineer



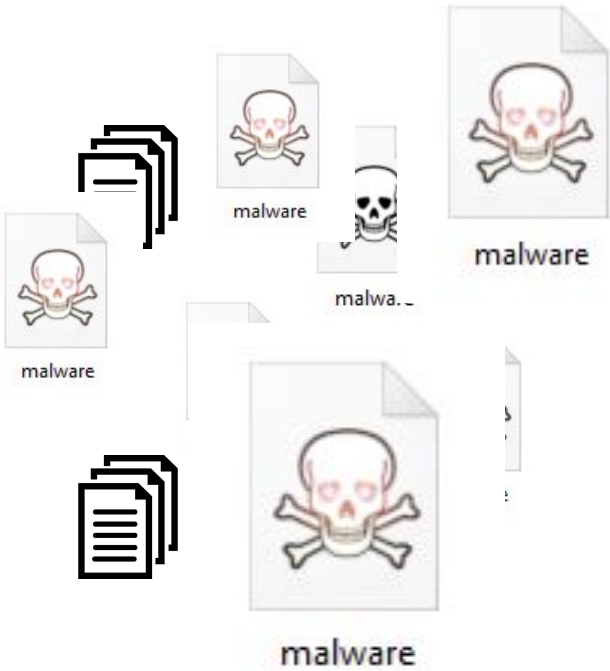


Malware reverse engineer





Malware reverse engineer







researcher.py



researcher.py



researcher.py





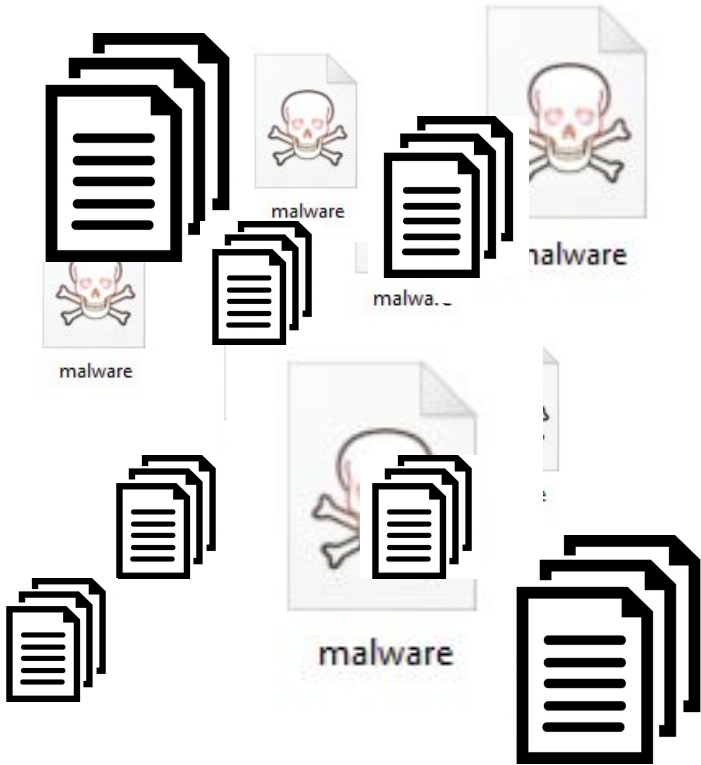
researcher.py



researcher.py



researcher.py





researcher.py



researcher.py



researcher.py

# MalWare DataBase

CERT.PL >\_ Configs Search Upload Stats VTI VTComms

Search

simple tag:neurus Search

Need Some Help?

Show 10 entries

Submitted	Hash
2017-12-08 23:01:05.206152	0c9e6170e18c390642f3c9ae6b96d696c656b5adfab34885f87538ee4d49112b
2017-12-08 23:01:03.870522	a8aa5ba5d1a021b22e6a8e040322128c44312a078ae1a007c47d002ac5d40569
2017-12-08 22:48:28.701221	3b71d1d33e94016210972b0ca49ab12632b0cd0e3e67dfcb304dd3be1ea5eb2a
2017-12-08 22:28:01.010423	be982b468a43082db4112cc75a08ea85701a223312ca2e5a01cfdcb8b45404
2017-12-08 22:27:35.564096	7bc9481bbb2762e1ab5dbdc65528f8aa678829beebc0c768e90fa1489d4c717
2017-12-08 21:01:50.572590	b9fe6cf4fce721ce0765996c661644d3e9413ce40d2ae012c655cfb3e0760e9c5d
2017-12-08 20:52:44.297410	b3c69b6170053f5f719ddd0fb7080ae9d4f8458516db9cf11b819cb934c3707
2017-12-08 20:52:43.191472	315e1142ad7c89ef8576a5d06398942849cf6b4bc7a062070108f851128856f
2017-12-08 20:31:54.993899	95c5d73611a64a81868ddfc2ad4dd389a9e1d9919fce7939b600092a5ca674
2017-12-08 20:31:54.051588	03a09a492531571198b642d48f2a49b7169fce53a0aa032180fb707c4316a81ab

Showing 1 to 10 of 182 entries



researcher.py



researcher.py



researcher.py

# MalWareDB

CERT.PL > Configs Search Upload Stats VTI VTComms

Search

simple tag:neurus Search

Need Some Help?

Show 10 entries

Submitted	Hash
2017-12-08 23:01:05.206152	0cfe6170e18c390642f3ceae6b96d696c656b5adfab34885f87538ee4d49112b
2017-12-08 23:01:03.870522	a8aa5ba5d1a021b22e6a8e040322128c44312a078ae1a007c47d002ac5d40569
2017-12-08 22:48:28.701221	3b71dd33e94016210972b0ca49ab12632b0cd0e3e67dfcb304dd3be1ea5eb2a
2017-12-08 22:28:01.010423	be982b468a43082db4112cc75a08ea85701a223312ca2e5a01cfdcb8b45404
2017-12-08 22:27:35.564096	7bc9481bbb2762e1ab5dbdc65528f8aae678829beebc0c768e90fa1489d4c717
2017-12-08 21:01:50.572590	b9fe6cf4fce721ce0765996c661644d4be9413ce40d2ae012c655cfb3e0760e9c5d
2017-12-08 20:52:44.297410	b3c69b6170053f5f719ddd0fb7080ae9d4f8458516db9cf11b819cb934c3707
2017-12-08 20:52:43.191472	315e1142ad7c89ef8576a5d06398942849cf6b4bc7a062070108fdfs1128856f
2017-12-08 20:31:54.993989	95c5d73611a64a81868ddfc2ad4dd389a9e1d9919fce7939b600092a5ca674
2017-12-08 20:31:54.051588	03a09a492531571198b642d48f2a49b7169ce53a0aa032f80fb707c4316a81ab

Showing 1 to 10 of 182 entries

mwdb.cert.pl



researcher.py



researcher.py



researcher.py

# MWDB

CERT.PL > Configs Search Upload Stats VTI VTComms

Search

simple tag:neurus Search

Need Some Help?

Show 10 entries

Submitted	Hash
2017-12-08 23:01:05.206152	0c9e6170e18c390642f3cea6b96d696c56b5adfab34885f87538ee4d49112b
2017-12-08 23:01:03.870522	a8aa5ba5d1a021b22e6a8e040322128c44312a078ae1a007c47d002ac5d40569
2017-12-08 22:48:28.701221	3b71d1d33e94016210972b0ca49ab12632b0cd0e3e67dfcb304dd3be1ea5eb2a
2017-12-08 22:28:01.010423	be982b468a43082db4112cc75a08ea85701a223312ca2e5a01cfdcb8b45404
2017-12-08 22:27:35.564096	7bc9481bbb2762e1ab5dbdc65528f8aae678829beebc0c768e90fa1489d4c717
2017-12-08 21:01:50.572590	b9fe6cf4fce721ce0765996c661644d9413ce40d2ae012c655cfb3e0760e9c5d
2017-12-08 20:52:44.297410	b3c69b6170053f5f719ddd0fb7080ae9d4f8458516db9cf11b819cb934c3707
2017-12-08 20:52:43.191472	315e1142ad7c89ef8576a5d06398942849cf6b4bc7a062070108fdfs1128856f
2017-12-08 20:31:54.993989	95c5d73611a64a81868ddfc2ad4dd389a9e1d9919fce7939b600092a5ca674
2017-12-08 20:31:54.051588	03a09a492531571198b642d48f2a49b7169fce53a0aa032180fb707c4316a81ab

Showing 1 to 10 of 182 entries

mwdb.cert.pl



# MWDB

## The Interface



CERT.PL >\_ Configs Search Upload Stats VTI VTComms

Search

simple tag:neurus Search

Need Some Help?

Show 10 entries

Submitted	Hash
2017-12-08 23:01:05.206152	0c6e9170e18c3906423ceae6b96d696c656b5ad1ab34889f87538ee4d491f2b
2017-12-08 23:01:03.870522	a8aa5ba5d1a021b22e6a8e04322128c44312a078ae1a007c47d002ac540569
2017-12-08 22:48:28.701221	3b71fd33e94016210972b0ca49ab12632b0c0e3e67dfcb3043d30be1ea5eb2a
2017-12-08 22:28:01.010423	be982b468a43082db412f0c75a08e85701a223312ca2e5a01cdfbc8b45404
2017-12-08 22:27:35.564096	7bc9481bb2762e1ab5dbdc65528f8aa6f78829bee0c78e90fa1489d4c717
2017-12-08 21:01:50.572590	b9f6c4f0e721ce076596c66f644dbe9413ce40z2ae012c65cb3e0760e9c5d
2017-12-08 20:52:44.297410	b3c696b170053f8719d4d0fb7080ae9448458516db9d11ba819cb934c3707
2017-12-08 20:52:43.191472	315e1142ad7c89ef8576a5d06396942849cfe4bc7a062070108f0b91128896f
2017-12-08 20:31:54.993989	95c5d73611a64a81868ddf2ad4d4d389a9e1d9919fca7939b600092a5ca074
2017-12-08 20:31:54.051588	03a09a49253157198b6424682a49b7169fce53a0aa032f80b707c4316a81ab

Showing 1 to 10 of 182 entries

mwdb.cert.pl



# MWDB

## The Interface



CERT.PL > Configs Search Upload Stats VTI VTComms

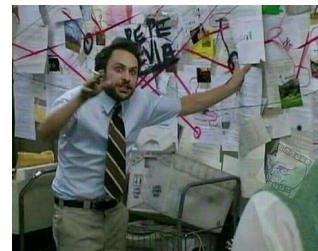
Search

simple tag:neurus Search

Show 10 entries [Need Some Help?](#)

Submitted	Hash
2017-12-08 23:01:05.206152	0c6e8170e18c3906423ceae6b96d696c5565adfab34888f87538ee449f12b
2017-12-08 23:01:03.870522	a8aa5ba5d1a021b22e6a8e040322128c44312a078ae1a007c47d002ac540569
2017-12-08 22:48:28.701221	3b71fd33e94016210972bcca49ab12632b0c0f0e3e67dfcb304d53be1ea5eb2a
2017-12-08 22:28:01.010423	be982b468a43082db412f0c75a08ea85701a223312ca2e5a01cdfbc8b454504
2017-12-08 22:27:35.564096	7bc9481bb2762e1ab5dbdc65a28f8aa6f78829beebc0c786e90fa1489d4c717
2017-12-08 21:01:50.572590	b9fe6cd4fce721ce076599c66f644dbe9413ce40z2ae012cb3c83e0760e9c5d
2017-12-08 20:52:44.297410	b3c696b170053f8719d4d0fb7080ae9448458516db9cf11b819c9934c3707
2017-12-08 20:52:43.191472	315e1142ad7c89ef8576a5d96396942849c8fbc4bc7a062070108fb91128896f
2017-12-08 20:31:54.993989	95c5d73611a64a81868ddf02d4d4d389a9e1d9919fca7939b600092a5ca074
2017-12-08 20:31:54.051588	03a09a492531571198b6424482a49b7169fce53a0aa032280b707c4316a81ab

Showing 1 to 10 of 182 entries



mwdb.cert.pl



# Tour de mwdb.cert.pl

<https://mwdb.cert.pl>



Training materials

<https://github.com/CERT-Polska/training-mwdb>



# mwdb.cert.pl



## Login with workshop creds (`first2020-xx`)

Check your inbox (information was sent in the last 24h)  
if you do not have credentials let us know now

← → ↻ 🏠 🔒 https://mwdb.cert.pl/login ⋮ 📄 🌙

**CERT.PL** > Register user About ▾

You need to authenticate before accessing this page

### Login

Please login using your credentials or [request an account using registration form](#) or write an e-mail to [info@cert.pl](mailto:info@cert.pl).

Login

Password

[Forgot password?](#)

Training materials

<https://github.com/CERT-Polska/training-mwdb>

mwdb.cert.pl



## Exercise #0: Getting familiar with the interface

<https://mwdb.readthedocs.io/en/latest/user-guide/1-Introduction-to-MWDB.html>






## Exercise #0: Getting familiar with the interface

CERT.PL > Samples Configs Blobs Upload Yara search Search Groups Statistics About Logged as: Profile Logout demo

X Search (Lucene query or hash)... Search ?

Quick query: Only uploaded by me Exclude public Favorites Exclude feed:\* Only ripped:\* Add +






Name/Hash	Size/Type	Tags	First seen
 <b>Name:</b> K8I57NYN3.exe <b>SHA256:</b> 902eb186a...1669bc377c4f <b>MD5:</b> b9842537f...68da8e0092f2	<b>Size:</b> 1.47 MB <b>Type:</b> PE32 executable (GUI) I...	<a href="#">feed:urlhaus</a> <a href="#">runnable:win32:exe</a> <a href="#">urlhaus:exe</a> <a href="#">urlhaus:finderbot</a> <a href="#">urlhaus:opendir</a>	Sun, 11 Apr 2021 14:44:28 GMT
 <b>Name:</b> jew.mpsl <b>SHA256:</b> 1d2e11bc0...ed53c5a78b3d <b>MD5:</b> 19830e713...e01990b4dc42	<b>Size:</b> 94.21 kB <b>Type:</b> ELF 32-bit LSB execut...	<a href="#">feed:urlhaus</a> <a href="#">mirai</a> <a href="#">ripped:mirai</a> <a href="#">runnable:linux</a> <a href="#">urlhaus:elf</a> <a href="#">urlhaus:mirai</a>	Sun, 11 Apr 2021 14:44:04 GMT
 <b>Name:</b> jew.mips <b>SHA256:</b> 081228dfb...ce6f03037316 <b>MD5:</b> 0d105f802...2959bfd827ef	<b>Size:</b> 90.21 kB <b>Type:</b> ELF 32-bit MSB execut...	<a href="#">feed:urlhaus</a> <a href="#">mirai</a> <a href="#">ripped:mirai</a> <a href="#">runnable:linux</a> <a href="#">urlhaus:elf</a> <a href="#">urlhaus:mirai</a>	Sun, 11 Apr 2021 14:44:03 GMT

Link to sample details

Click on tags to filter (or filter out)



## Exercise #0: Getting familiar with the interface

 <b>Name:</b> jew.mpsl <b>SHA256:</b> 1d2e11bc0...ed53c5a78b3d <b>MD5:</b> 19830e713...e01990b4dc42	<b>Size:</b> 94.21 kB <b>Type:</b> ELF 32-bit LSB execut...	<a href="#">feed:urlhaus</a>  <a href="#">ripped:mirai</a>  <a href="#">runnable:linux</a>  <a href="#">urlhaus:elf</a>  <a href="#">urlhaus:mirai</a> 	Sun, 11 Apr 2021 14:44:04 GMT
 <b>Name:</b> jew.mpsl <b>SHA256:</b> 1d2e11bc0...ed53c5a78b3d <b>MD5:</b> 19830e713...e01990b4dc42	<b>Size:</b> 94.21 kB <b>Type:</b> ELF 32-bit LSB execut...	<a href="#">feed:urlhaus</a>  <a href="#">ripped:mirai</a>  <a href="#">runnable:linux</a>  <a href="#">urlhaus:elf</a>  <a href="#">urlhaus:mirai</a> 	Sun, 11 Apr 2021 14:44:04 GMT



tag:"mirai" AND NOT tag:"feed:urlhaus"

10141 results found


Quick query: [Only uploaded by me](#) [Exclude public](#) [Favorites](#) [Exclude feed:\\*](#) [Only ripped:\\*](#) [Add +](#)



## Exercise #0: Getting familiar with the interface

X 5762523a60685aafa8a681672403fd19 | Search


Quick query: [Only uploaded by me](#) [Exclude public](#) [Favorites](#) [Exclude feed:\\*](#) [Only ripped:\\*](#) [Add +](#)

Name/Hash	Size/Type	Tags	First seen
 <b>Name:</b> 88527fb710478e1c54c6... <b>SHA256:</b> 88527fb71...7fa4102b8411	<b>Size:</b> 3.38 MB <b>Type:</b> Zip archive data, at lea...	<a href="#">runnable:android:apk</a>	Sun, 11 Apr 2021 15:18:08 GMT

X md5:5762523a60685aafa8a681672403fd19 | Search ?

1 results found

Quick query: [Only uploaded by me](#) [Exclude public](#) [Favorites](#) [Exclude feed:\\*](#) [Only ripped:\\*](#) [Add +](#)

Name/Hash	Size/Type	Tags	First seen
 <b>Name:</b> 7f3d1f38b49054fa1a3fc4... <b>SHA256:</b> d457da57c... 47c99283d9c1 <b>MD5:</b> 5762523a60685aafa8a681672403fd19	<b>Size:</b> 319.73 kB <b>Type:</b> RAR archive data, v4, os...	<a href="#">archive:rar</a> <a href="#">feed:malwarebazaar</a>	Thu, 08 Apr 2021 07:05:21 GMT

mwdb.cert.pl



## Exercise #1: Filtering samples by tags

**mwdb.cert.pl**



## Exercise #1: Filtering samples by tags

**formbook**

Simple tag, mostly used for marking artifacts that are associated with malware family

**feed:sample**

Tag describing the source of malware sample

**ripped:formbook**

Tag for the original sample, indicating recognized malware family

**runnable:win32:exe**

Tag describing the type of sample

**yara:win\_formbook**

Generic metadata tag with additional information that are useful for filtering

<https://mwdb.readthedocs.io/en/latest/user-guide/5-Tagging-objects.html#built-in-tag-conventions>



## Exercise #1: Filtering samples by tags

### Lucene-based query syntax



conditions  
<field>:<value>

operators  
AND, OR, NOT  
(uppercase only)

wildcards

<https://mwdb.readthedocs.io/en/latest/user-guide/7-Lucene-search.html>









## Everything related with formbook

8472 results found

Quick query:      

Name/Hash	Size/Type	Tags	First seen
 <b>Name:</b> 3310000_55996a8917b... <b>SHA256:</b> 55996a891...c1b1122ca8d7 <b>MD5:</b> cedc1abf7...2d89abd544fd	<b>Size:</b> 154.75 kB <b>Type:</b> data	<b>dump:win32:exe</b>  <b>formbook</b> 	Sun, 11 Apr 2021 08:13:01 GMT
 <b>Name:</b> 5f85c45be7b228140ce5... <b>SHA256:</b> b9a1fe9ef...bccca5623704 <b>MD5:</b> 430a47302...5a335d184bc4	<b>Size:</b> 551 kB <b>Type:</b> PE32 executable (GUI) I...	<b>et:formbook</b>  <b>feed:malwarebazaar</b>  <b>ripped:formbook</b>  <b>runnable:win32:exe</b>  <b>yara:win_formbook</b> 	Sun, 11 Apr 2021 07:59:08 GMT

mwdb.cert.pl



## Exercise #1: Filtering samples by tags Ranges

X

size:[10000 TO 15000]

X

upload\_time:<=2020-01-01

X

size:[10kB TO 15kB]

X

upload\_time:"<=2020-01-01 16:00"

X

size:<=10kB

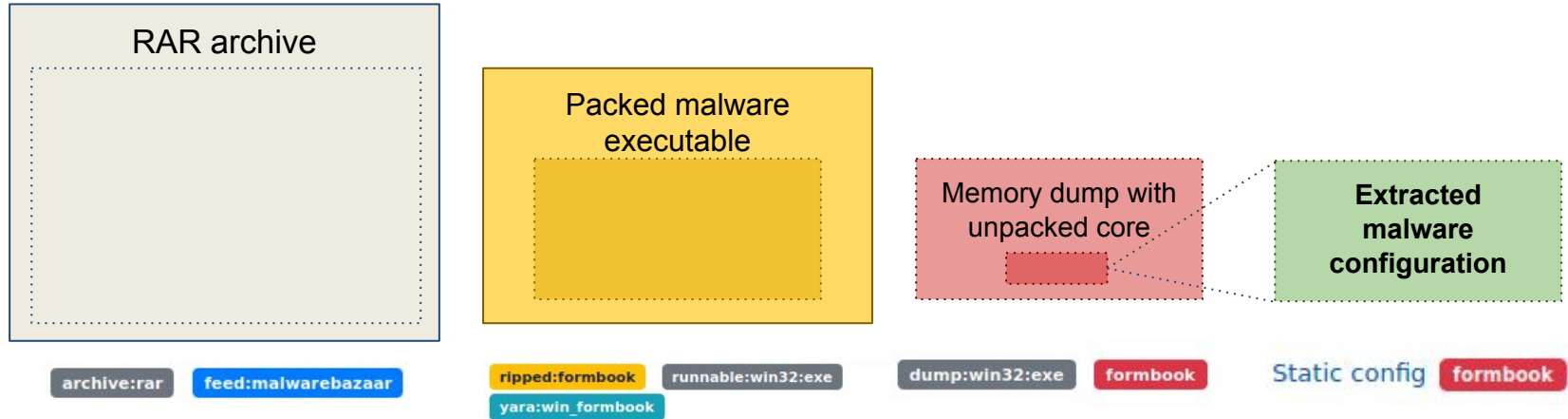
mwdb.cert.pl



## Exercise #2: Exploring sample view and hierarchy

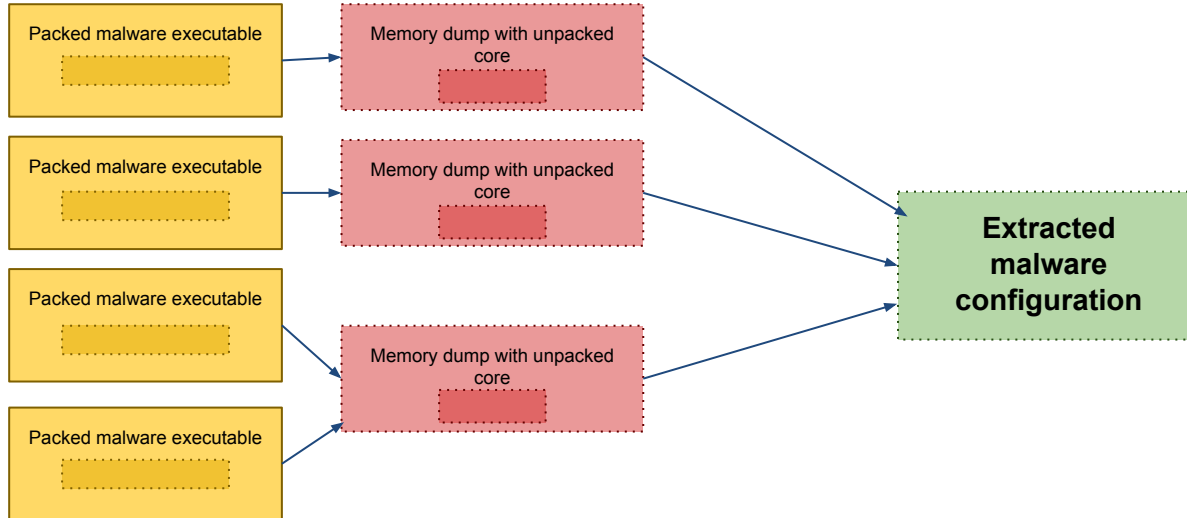


## Exercise #2: Exploring sample view and hierarchy





## Exercise #2: Exploring sample view and hierarchy



[mwdb.cert.pl](https://mwdb.cert.pl) 

When you upload 32 samples and  
all ripped to the same config



mwdb.cert.pl



## Exercise #3: Looking for similar configurations

mwdb.cert.pl

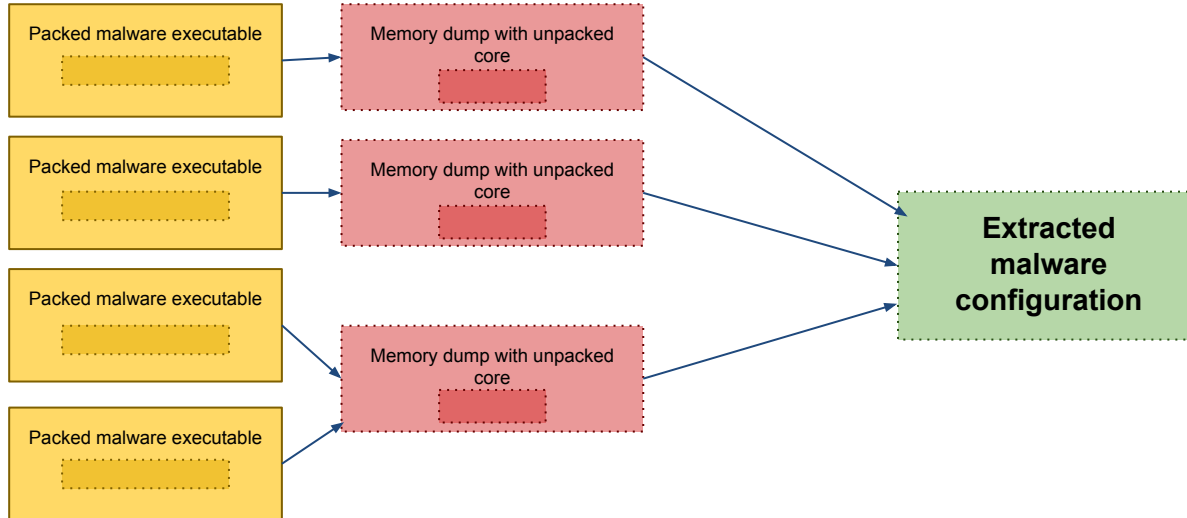


## Exercise #4: Blobs and dynamic configurations



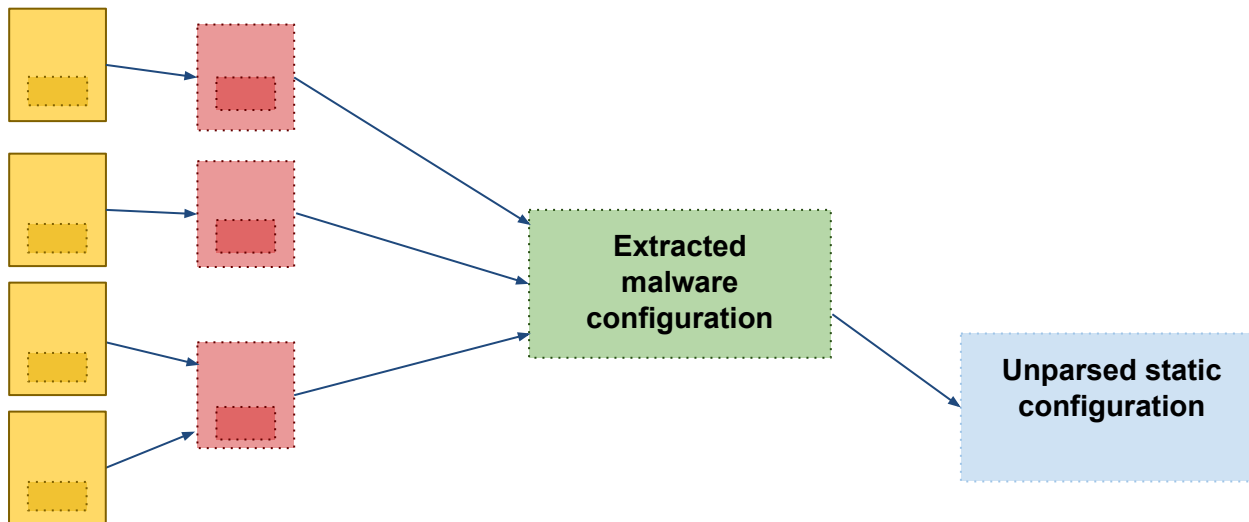


## Exercise #4: Blobs and dynamic configurations



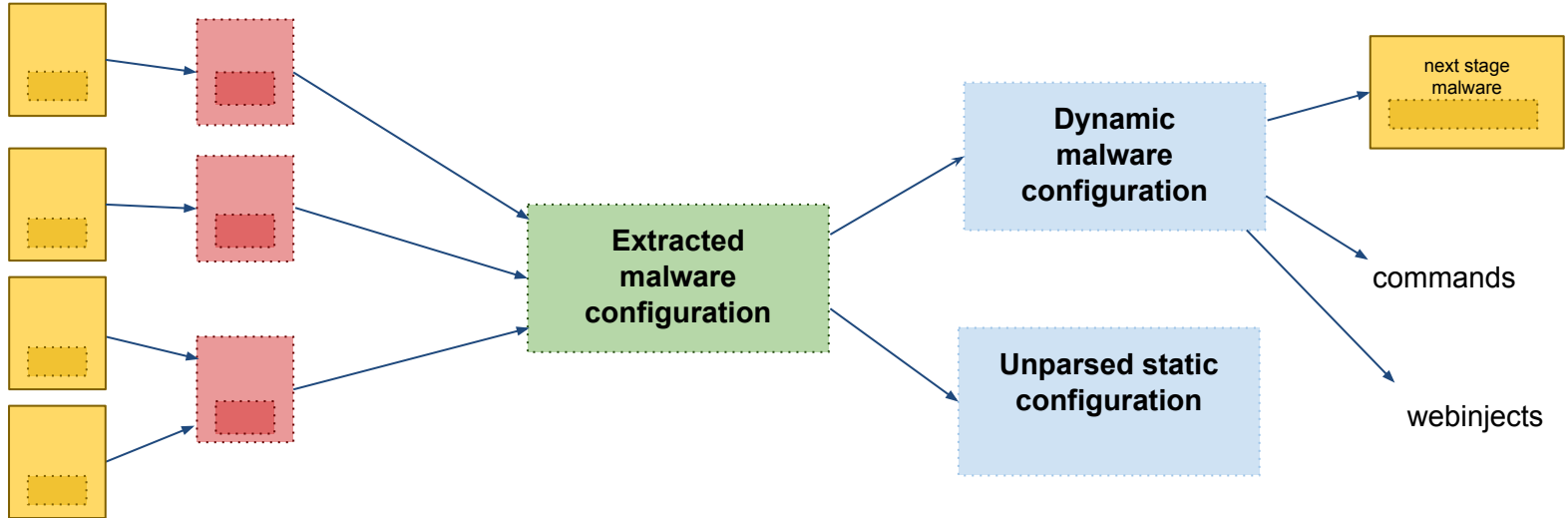


## Exercise #4: Blobs and dynamic configurations





## Exercise #4: Blobs and dynamic configurations



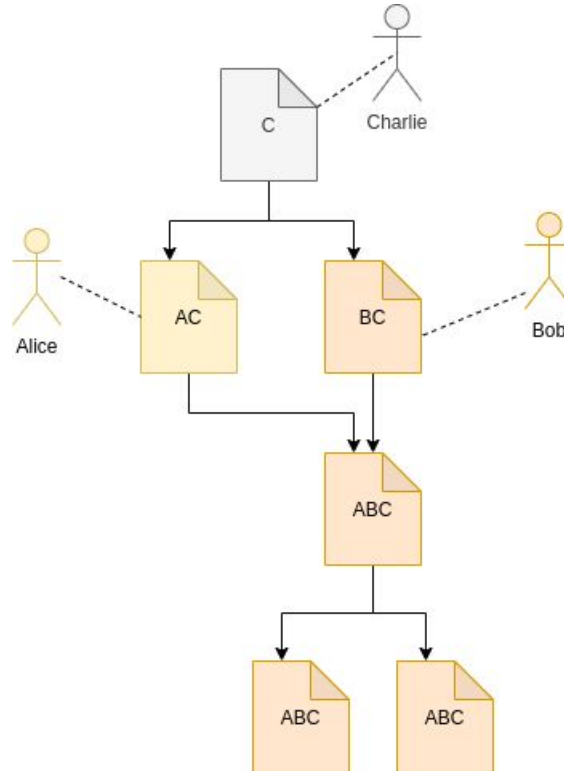
mwdb.cert.pl



## Exercise #5: Let's upload something!

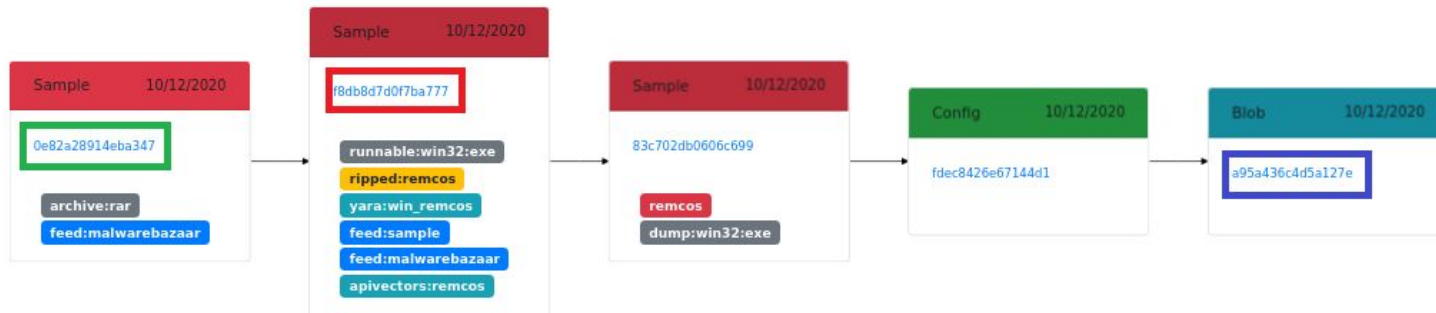


## Exercise #5: Let's upload something!





## Exercise #5: Let's upload something!



Shares

Group name	Reason	Access time
<a href="#">certpl-systems</a>	Added <a href="#">a95a436c4d5a127ed90a9c382476d9bc7c136efa39edbd912ab8935dc10d1b7d</a> by <a href="#">karton</a>	Mon, 12 Oct 2020 09:23:40 GMT
<a href="#">karton</a> (uploader)	Added <a href="#">a95a436c4d5a127ed90a9c382476d9bc7c136efa39edbd912ab8935dc10d1b7d</a> by <a href="#">karton</a>	Mon, 12 Oct 2020 09:23:40 GMT
<a href="#">public</a>	Added <a href="#">0e82a28914eba347b95549eb4b7c5d642c2ee6f747407b3a0d8c8c8ff988c215</a> by <a href="#">Alice</a>	Mon, 12 Oct 2020 09:23:40 GMT
<a href="#">Alice</a>	Added <a href="#">0e82a28914eba347b95549eb4b7c5d642c2ee6f747407b3a0d8c8c8ff988c215</a> by <a href="#">Alice</a>	Mon, 12 Oct 2020 09:23:40 GMT
<a href="#">Chris</a>	Added <a href="#">f8db8d7d0f7ba777f5f18452579f40a224a055c6624066b5dce501121cc91c5c</a> by <a href="#">Chris</a>	Tue, 13 Oct 2020 01:19:55 GMT

mwdb.cert.pl



# Scripting and automation with **mwdblib**



# mwdblib installation

## Setup environment

```
# Create virtualenv and activate
```

```
$ python3 -m venv venv
```

```
$ . venv/bin/activate
```

```
# Install mwdblib with CLI extras
```

```
(venv) $ pip install mwdblib[cli]
```

```
# ... and ipython for convenience
```

```
(venv) $ pip install ipython
```

If you don't know what is virtualenv, read more:

[Installing packages using pip and virtual environments — Python Packaging User Guide](#)



mwdb.cert.pl



## Exercise #6: Get recent files

mwdb.cert.pl



## Exercise #7: MWDBObject properties

See: <https://github.com/CERT-Polska/training-mwdb>

mwdb.cert.pl



## Exercise #8: Using mwdblib CLI

# mwdb.cert.pl



## After the workshop, feel free to register!

<https://mwdb.cert.pl/register>

During vetting, we want to be able to identify you and get an actionable e-mail in case of problems

### Hints for quick approval:

- use business e-mail (avoid gmail.com / protonmail.com / qq.com).
- provide some info about yourself (homepage, research blog, Twitter handle). Mention this workshop.
- don't worry to ping us if we don't answer for a few days



### Register user

Provided data are needed for vetting process. Keep in mind that all submissions are reviewed manually. Make sure you have read our [Terms of use](#).

Login

Login must contain only letters, digits, '.', and '-' characters, max 32 characters allowed.

Business e-mail

#### Additional information:

Affiliation

Provide name of company or university

Job Title

Provide your job title

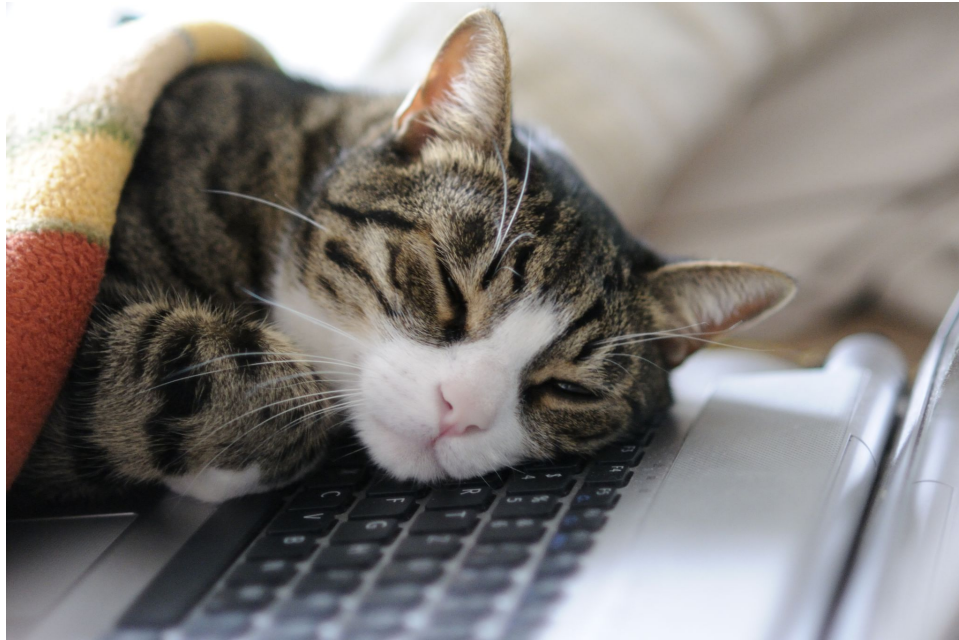
Job Responsibilities

Provide your job responsibilities and experience in the field of malware analysis

[mwdb.cert.pl](https://mwdb.cert.pl)



Coffee break



# Agenda

- **mwdb.cert.pl**
  - What the heck is **MWDB**
  - Tour de **mwdb.cert.pl**
  - Scripting and automation with **mwdblib**
- **mwdb-core and karton**
  - Run a **self-hosted** mwdb-core and karton instances
  - Experiment with **karton-playground**
  - Distributed collaboration with mwdb **remotes**
  - Advanced programming techniques with **malduck**



YOU ARE  
HERE

Learn **karton** with the **karton-playground**



Run a **self-hosted** mwdb-core and karton instance

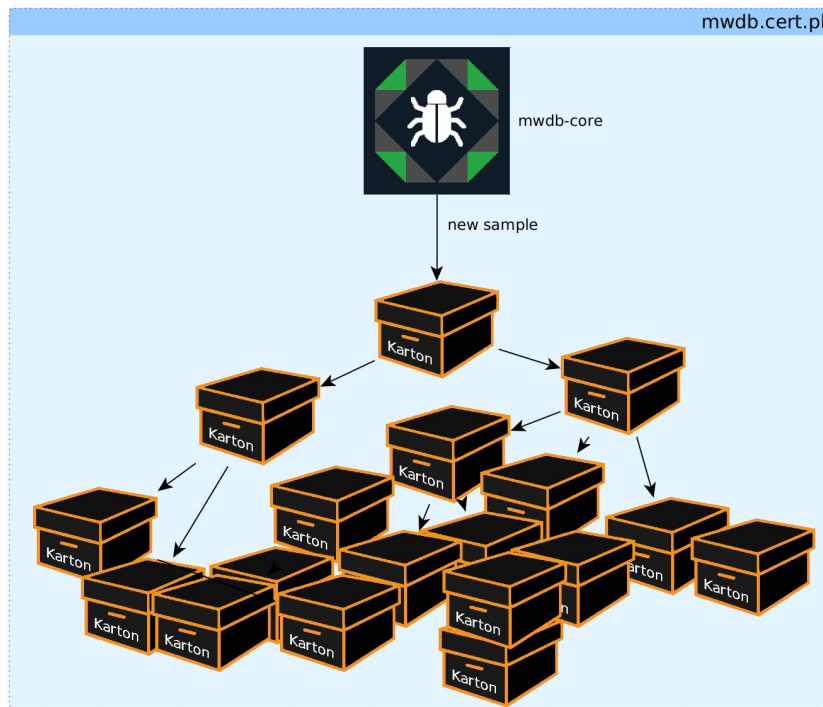


# mwdb-core





# mwdb-core



mwdb-core



# mwdb-core + karton = a usable service

MWDB is only a frontend.


To make it possible to create an environment similar to ours, we've decided to open-source the “engine” of our pipeline too.

 <https://github.com/CERT-Polska/mwdb-core/>

 <https://github.com/CERT-Polska/karton>



# Karton Playground

- [Karton Playground](#) - a project dedicated for karton learners
- An easy way to set up the environment and get to work
- Not suitable for production
-  <https://github.com/CERT-Polska/karton-playground>



*image credit: wikipedia*

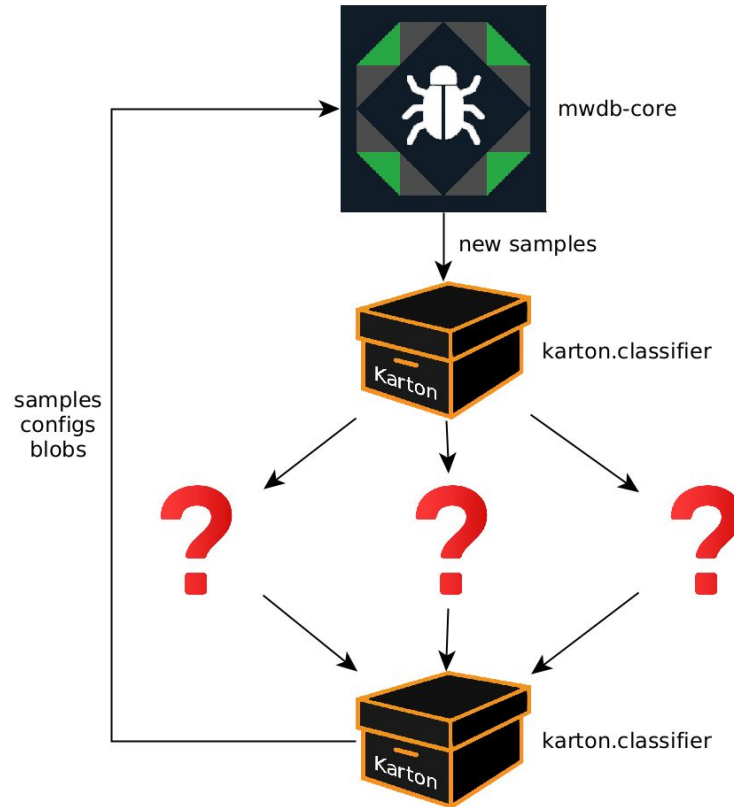
# Karton Playground

```
git clone https://github.com/CERT-Polska/karton-playground.git
cd karton-playground
sudo docker-compose up # this may take a while
```

- Go grab a cup of coffee - this will take a while
- But when it's done, you will have a working instance on your local machine

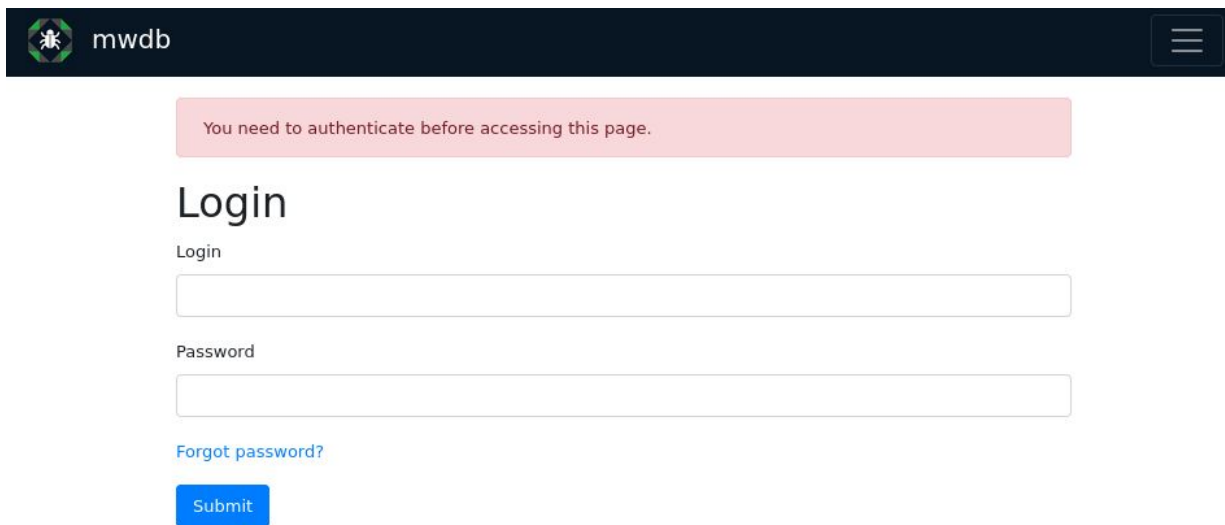
# Karton Playground



While you wait...



# Karton Playground: click around

Navigate to <http://127.0.0.1:8080>. Login using admin:admin.



 mwdb 

You need to authenticate before accessing this page.

## Login

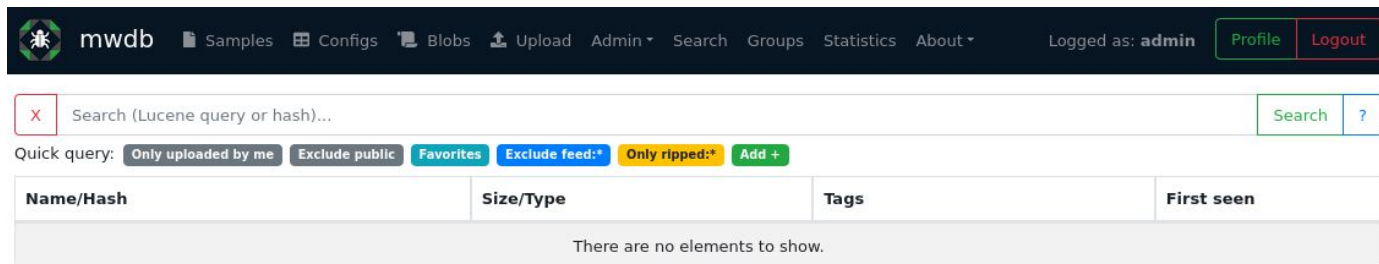
Login

Password

[Forgot password?](#)

# Karton Playground: click around

Navigate to <http://127.0.0.1:8080>. Login using admin:admin.



The screenshot shows the mwdb interface. The top navigation bar includes links for Samples, Configs, Blobs, Upload, Admin, Search, Groups, Statistics, and About. The user is logged in as 'admin' and has buttons for Profile and Logout. Below the navigation bar is a search bar with a placeholder 'Search (Lucene query or hash)...' and a 'Search' button. Underneath the search bar are several filter buttons: 'Only uploaded by me', 'Exclude public', 'Favorites', 'Exclude feed:\*', 'Only ripped:\*', and 'Add +'. Below the filters is a table with the following columns: 'Name/Hash', 'Size/Type', 'Tags', and 'First seen'. The table is currently empty, with a message 'There are no elements to show.' displayed below the header.

Name/Hash	Size/Type	Tags	First seen
There are no elements to show.			

There is no malware yet... But that's about to change!



# Karton Playground: click around

Check out the karton dashboard at <http://127.0.0.1:8030/> too:

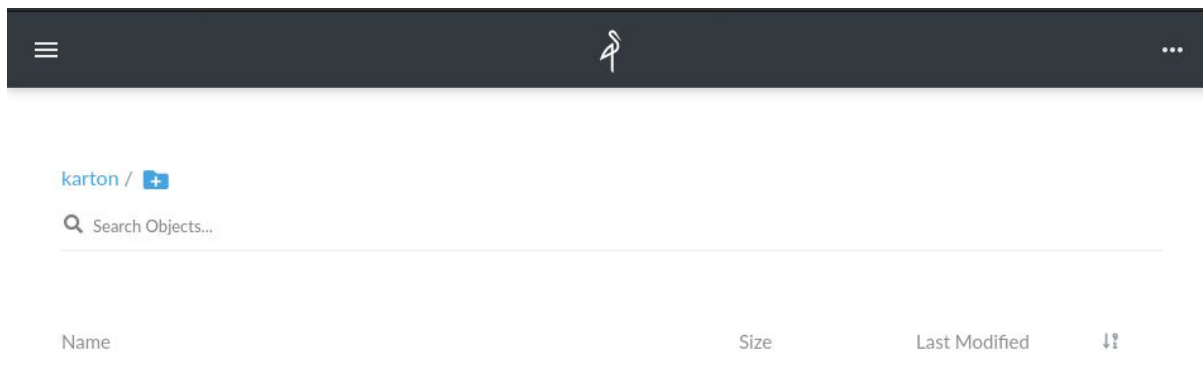
karton

## binds

identity	filters	tasks	errors	replicas
karton.classifier 4.0.4	kind:raw type:sample	0	0	1
karton.mwdb-reporter 4.0.4	stage:recognized type:sample stage:analyzed type:sample type:config	0	0	1

# Karton Playground: click around

Optional: check out the minio interface <http://127.0.0.1:8090/> (mwdb :mwdbmwdb)



# Karton Playground exercise

Integrate an existing karton service into your pipeline: karton-autoit-ripper

 <https://github.com/CERT-Polska/karton-autoit-ripper>

```
$ python3 -m venv venv
$ source ./venv/bin/activate
$ pip install karton-autoit-ripper

$ # playground-specific: copy local config to cwd
$ cp config/kartonlocal.ini karton.ini
$ karton-autoit-ripper
[2021-04-11 17:19:57,867][INFO] Service karton.autoit-ripper started
```

# Karton Playground exercise

Download a sample, and verify its hash:

```
$ wget https://github.com/CERT-Polska/training-mwdb/blob/main/autoit-malware.bin
$ sha256sum autoit-malware.bin
a4816d4fec6d6d2806d5b105c3aab55f4a1eb5deb3b126f317093a4dc4aab88a1 autoit-malware.bin
```

Finally, upload it to your local mwdb (<http://127.0.0.1:8080>, admin:admin)

```
$ karton-autoit-ripper
[2021-04-11 17:19:57,867][INFO] Service karton.autoit-ripper started
/home/msm/Projects/karton-playground/venv/lib/python3.8/site-packages/karton/core/logger.py:57: UserWarning: There is no active
warnings.warn("There is no active log consumer to receive logged messages.")
[2021-04-11 17:19:57,871][INFO] Binding on: {'type': 'sample', 'stage': 'recognized', 'kind': 'runnable', 'platform': 'win32'}
[2021-04-11 17:19:57,871][INFO] Binding on: {'type': 'sample', 'stage': 'recognized', 'kind': 'runnable', 'platform': 'win64'}
[2021-04-11 17:20:10,645][INFO] Received new task - cbe177c0-a824-47be-a1c9-fb0aa4898f75
[2021-04-11 17:20:10,661][INFO] Found a possible autoit v3.26+ binary
[2021-04-11 17:20:14,149][INFO] Found embedded data, reporting!
[2021-04-11 17:20:14,150][INFO] Sending a task with script.au3
[2021-04-11 17:20:14,261][INFO] Looking for a binary embedded in the script
[2021-04-11 17:20:14,305][INFO] Task done - cbe177c0-a824-47be-a1c9-fb0aa4898f75
```

# Karton Playground exercise

Volia!

File details

Details Relations Preview

```
graph LR; S1[Sample 4/11/2021  
a4816d4fec6d6d280  
runnable:win32:exe] --> S2[Sample 4/11/2021  
5689d35ddaac5435]
```

Tags

runnable:win32:exe x

Add tag Add

Related samples + Add

child 5689d35ddaac5435facbf144e82d91c1f568db5f6adcd41a995ddb89f9ba33f7

Attributes + Add

Karton analysis

✓ done 8b859823-6a49-45e5-9039-a51fa8e6a915 ▾

+ reanalyze

# Karton Playground exercise

- But using existing services is just half the fun
- For a real Karton experience, write your own service
- Download a template:

<https://github.com/CERT-Polska/training-mwdb/blob/main/karton-template.py>

```
class MyFirstKarton(Karton):
    identity = "karton.first"
    filters = [{"type": "sample", "stage": "recognized"}]

    def process(self, task: Task) -> None:
        sample_resource = task.get_resource("sample") # Get the incoming sample
        self.log.info(f"Hi {sample_resource.name}, let me analyse you!") # Log with self.log

        with sample_resource.download_temporary_file() as sample_file: # Download to a temporary file
            result = do_your_processing(sample_file.name) # And process it

        self.send_task(Task(
            {"type": "sample", "stage": "analyzed"},
            payload={"parent": sample_resource, "sample": Resource("result-name", result)},
        )) # Upload the result as a sample:

if __name__ == "__main__":
    MyFirstKarton().loop() # Here comes the main loop
```

# Karton Playground exercise

- Karton's "identity": `identity = "karton.first"`
- Python namespace: `import karton.first`
- Pypi package: `pip install karton-first`

```
class MyFirstKarton(Karton):
    identity = "karton.first"
    filters = [{"type": "sample", "stage": "recognized"}]

    def process(self, task: Task) -> None:
        sample_resource = task.get_resource("sample") # Get the incoming sample
        self.log.info(f"Hi {sample_resource.name}, let me analyse you!") # Log with self.log

        with sample_resource.download_temporary_file() as sample_file: # Download to a temporary file
            result = do_your_processing(sample_file.name) # And process it

        self.send_task(Task(
            {"type": "sample", "stage": "analyzed"},
            payload={"parent": sample_resource, "sample": Resource("result-name", result)},
        )) # Upload the result as a sample:

if __name__ == "__main__":
    MyFirstKarton().loop() # Here comes the main loop
```

# Karton Playground exercise

- What are these?

```
class MyFirstKarton(Karton):
    identity = "karton.first"
    filters = [{"type": "sample", "stage": "recognized"}]

    def process(self, task: Task) -> None:
        sample_resource = task.get_resource("sample") # Get the incoming sample
        self.log.info(f"Hi {sample_resource.name}, let me analyse you!") # Log with self.Log

        with sample_resource.download_temporary_file() as sample_file: # Download to a temporary file
            result = do_your_processing(sample_file.name) # And process it

        self.send task(Task(
            {"type": "sample", "stage": "analyzed"},
            payload={"parent": sample_resource, "sample": Resource("result-name", result)},
        )) # Upload the result as a sample:

if __name__ == "__main__":
    MyFirstKarton().loop() # Here comes the main loop
```



# Karton pipeline

karton.classifier

3.0.0

kind:raw

type:sample



# Karton pipeline

karton.extractor

2.x.x

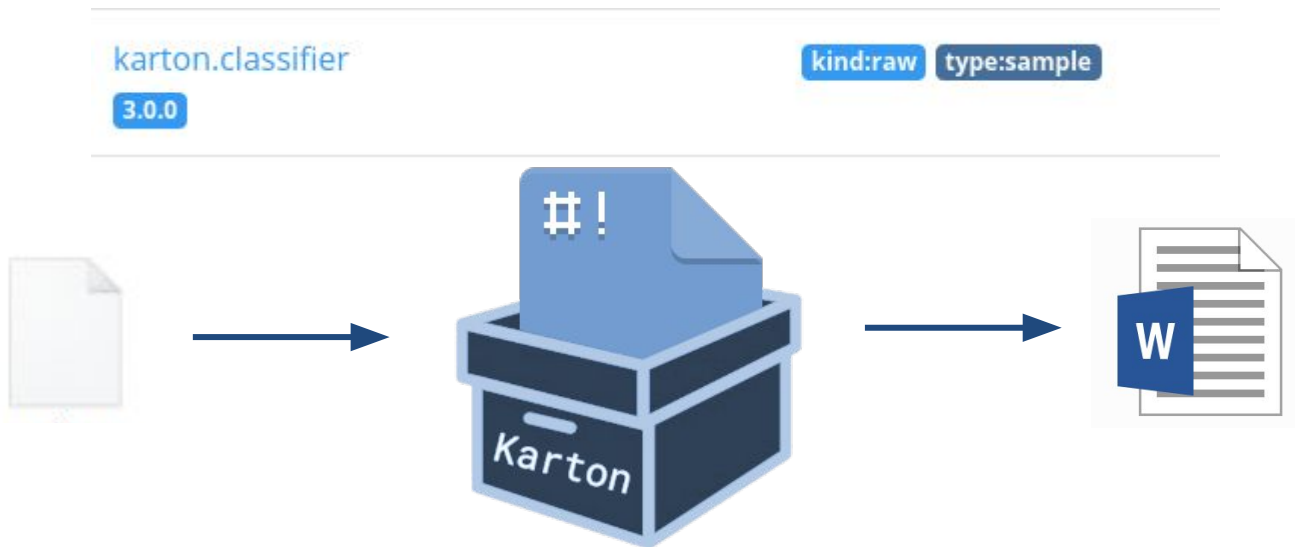
kind:archive

stage:recognized

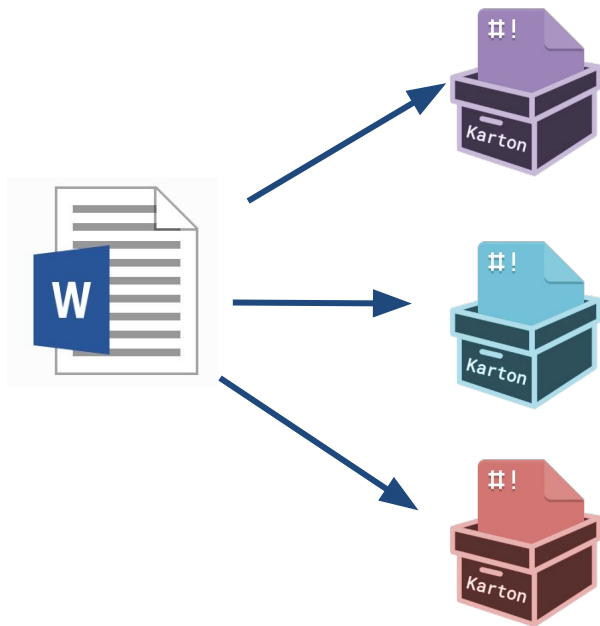
type:sample



# Karton pipeline



# Karton pipeline



karton.cuckoo1

2.x.x

platform:win32 quality:high stage:recognized type:sample

karton.drakrun-prod

2.x.x

platform:win32 stage:recognized type:sample

platform:win64 stage:recognized type:sample

karton.macro-unpacker

3.1.0

extension:xlsx kind:document stage:recognized type:sample

extension:xlsm kind:document stage:recognized type:sample

extension:xls kind:document stage:recognized type:sample

extension:doc kind:document stage:recognized type:sample

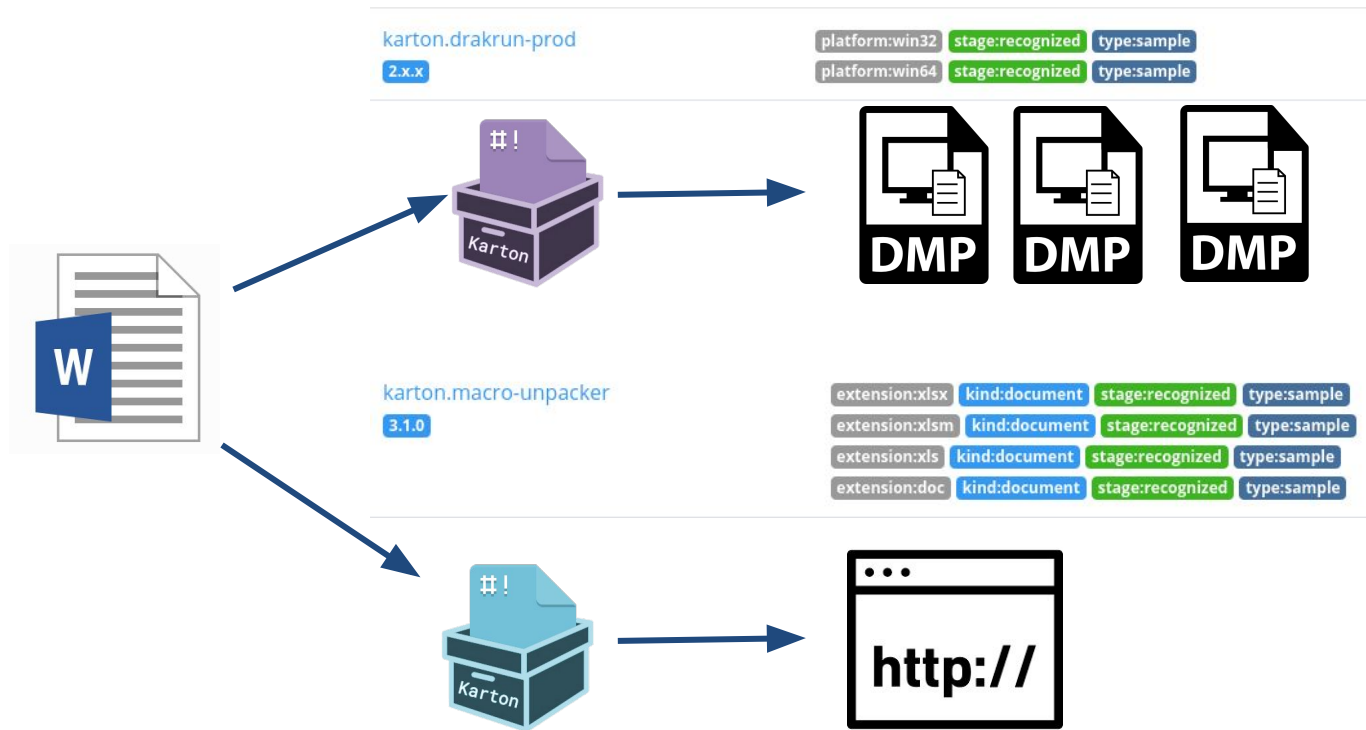
karton.emodoc

2.x.x

extension:doc kind:document stage:recognized type:sample

extension:docx kind:document stage:recognized type:sample

# Karton pipeline

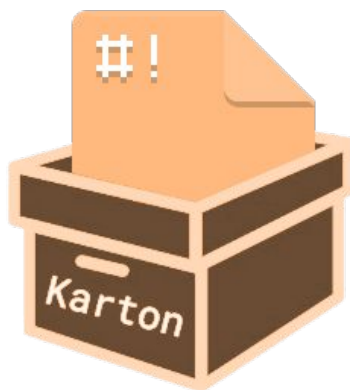


# Karton pipeline

karton.roach-ripper

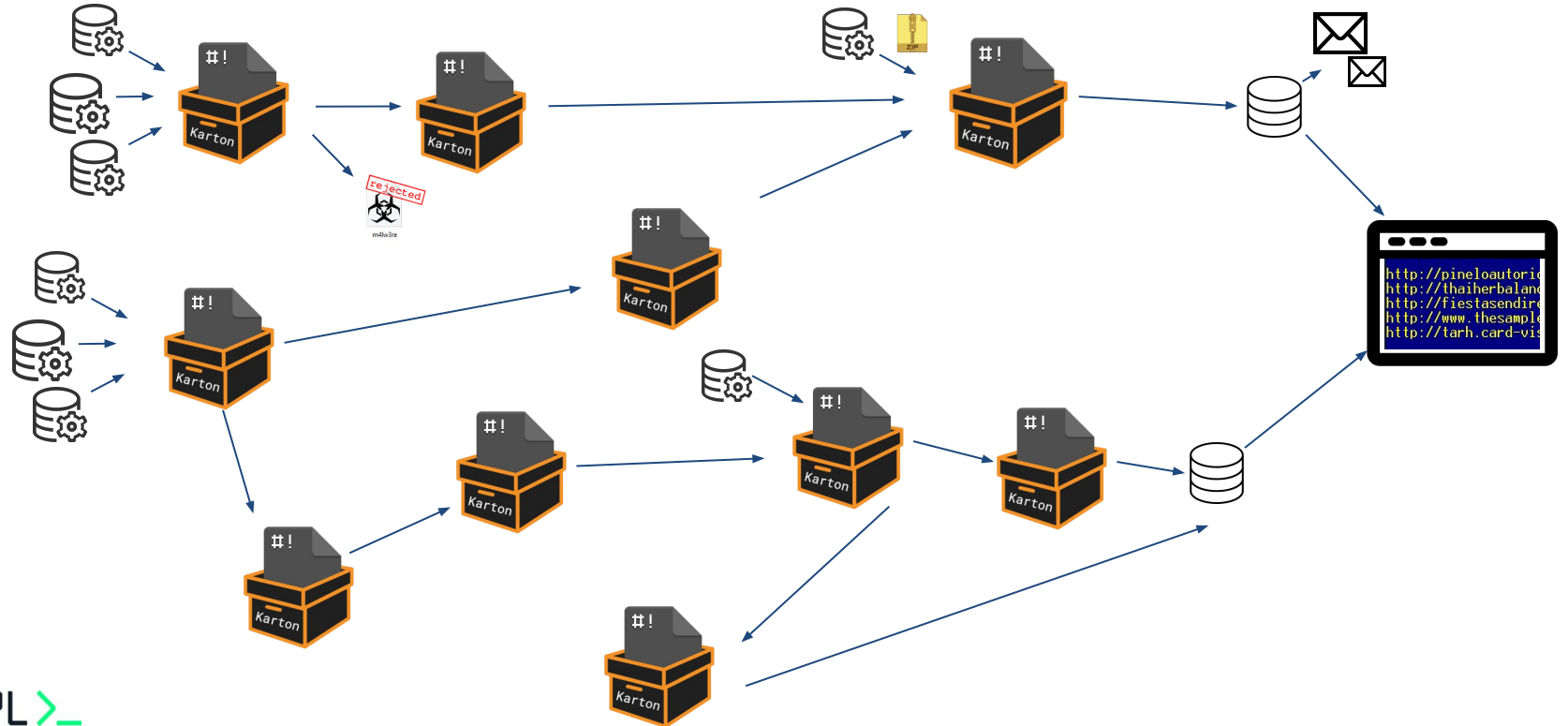
2.x.x

kind:runnable platform:win32 stage:recognized type:sample  
kind:runnable platform:win64 stage:recognized type:sample  
kind:runnable platform:linux stage:recognized type:sample  
kind:drakrun-prod type:analysis



Family	zloader
Config type	static
+ binary_id	bot7
+ rc4key	NvuVIV3kbg7
+ rsa_key	-----BEGIN PUBLIC KEY----- MIG
+ type	zloader
+ urls	[ { "url": "https://militanttr

# Karton pipeline in the real world



# Karton Playground exercise

- Karton tasks are routed in the system based on their headers
- Consumer declares what kind of tasks it is interested in
- Producer indicates the kind of produced task

```
class MyFirstKarton(Karton):
    identity = "karton.first"
    filters = [{"type": "sample", "stage": "recognized"}]

    def process(self, task: Task) -> None:
        sample_resource = task.get_resource("sample") # Get the incoming sample
        self.log.info(f"Hi {sample_resource.name}, let me analyse you!") # Log with self.Log

        with sample_resource.download_temporary_file() as sample_file: # Download to a temporary file
            result = do_your_processing(sample_file.name) # And process it

        self.send task(Task(
            {"type": "sample", "stage": "analyzed"},
            payload={"parent": sample_resource, "sample": Resource("result-name", result)},
        )) # Upload the result as a sample:

if __name__ == "__main__":
    MyFirstKarton().loop() # Here comes the main loop
```



# Karton Playground exercise

- Karton = Consumer + Producer

```
class MyFirstKarton(Karton):
    identity = "karton.first"
    filters = [{"type": "sample", "stage": "recognized"}]

    def process(self, task: Task) -> None:
        sample_resource = task.get_resource("sample") # Get the incoming sample
        self.log.info(f"Hi {sample_resource.name}, let me analyse you!") # Log with self.Log

        with sample_resource.download_temporary_file() as sample_file: # Download to a temporary file
            result = do_your_processing(sample_file.name) # And process it

        self.send_task(Task(
            {"type": "sample", "stage": "analyzed"},
            payload={"parent": sample_resource, "sample": Resource("result-name", result)},
        )) # Upload the result as a sample:

if __name__ == "__main__":
    MyFirstKarton().loop() # Here comes the main loop
```

```
class Karton(Consumer, Producer):
    """
    This glues together Consumer and Producer - which is the most common use case
    """
```

# Karton Playground exercise

- Resource - bigger files, hosted on minio (or other s3 compatible storage server)

```
class MyFirstKarton(Karton):
    identity = "karton.first"
    filters = [{"type": "sample", "stage": "recognized"}]

    def process(self, task: Task) -> None:
        sample_resource = task.get_resource("sample") # Get the incoming sample
        self.log.info(f"Hi {sample_resource.name}, let me analyse you!") # Log with self.log

        with sample_resource.download_temporary_file() as sample_file: # Download to a temporary file
            result = do_your_processing(sample_file.name) # And process it

        self.send_task(Task(
            {"type": "sample", "stage": "analyzed"},
            payload={"parent": sample_resource, "sample": Resource("result-name", result)},
        )) # Upload the result as a sample:

if __name__ == "__main__":
    MyFirstKarton().loop() # Here comes the main loop
```

# Karton Playground exercise

- Download a template:  
<https://github.com/CERT-Polska/training-mwdb/blob/main/karton-template.py>
- Your task: edit the template, and:
  - Run the `strings` utility on every incoming sample
  - Save the result in a variable (use `subprocess.check_output`)
  - Upload the result to mwdb (already handled in the template)
- Start your first karton!

```
$ python3
Python 3.8.5 (default, Jan 27 2021, 15:41:15)
>>> import subprocess
>>> s = subprocess.check_output(["strings", "/bin/ls"])
>>> print(s.decode())
/lib64/ld-linux-x86-64.so.2
.j<c~
MB#F-
Libselinux.so.1
...
```

# Karton Playground exercise: solution

```
$ python3 karton-template.py
[2021-04-14 20:56:28,927][INFO] Service karton.first started
/home/msm/Projects/karton-playground/venv/lib/python3.8/site-packages/karton/core
warnings.warn("There is no active log consumer to receive logged messages.")
[2021-04-14 20:56:28,928][INFO] Binding on: {'type': 'sample', 'stage': 'recogniz
[2021-04-15 08:45:10,546][INFO] Received new task - c17c9659-49d6-444c-b208-f00fc
[2021-04-15 08:45:10,547][INFO] Hi setup_1.0.61.exe, let me analyse you!
[2021-04-15 08:45:11,100][INFO] Task done - c17c9659-49d6-444c-b208-f00fcd36bc5b
```

File details	
<a href="#">Details</a> <a href="#">Relations</a> <a href="#">Preview</a> <a href="#">Remove</a> <a href="#">+ Upload child</a> <a href="#">Favorite</a> <a href="#">Download</a>	
<b>File name</b>	<a href="#">setup_1.0.61.exe</a>
<b>File size</b>	15.33 MB
<b>File type</b>	PE32 executable (GUI) Intel 80386, for MS Windows, Nullsoft Installer self-extracting archive
<b>md5</b>	c042ee96b54c8c121165cc1b5e338fe7

Tags	
<a href="#">runnable:win32:exe X</a>	
Add tag	<input type="button" value="Add"/>
Related samples	<a href="#">+ Add</a>
<b>child</b>	<a href="#">3eec0da26f86d1ae53d594d52adc3b59a596ed79e8701d130f52870c46a1d709</a>

# Karton Playground exercise: solution

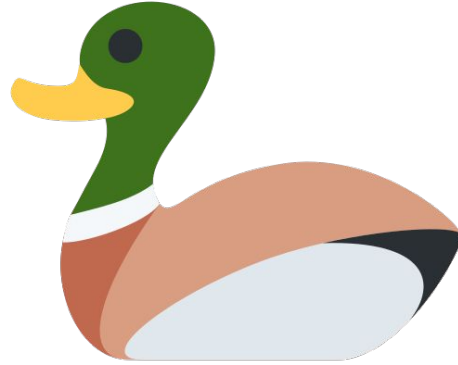
The screenshot displays the mwdb interface. At the top, a dark navigation bar contains the mwdb logo and menu items: Samples, Configs, Blobs, Upload, Admin, Search, Groups, Statistics, and About. Below this, a 'File details' window is open, showing a list of file sections. The first section is highlighted in grey and contains the text: '!This program cannot be run in DOS mode.'. The other sections are listed with their respective symbols and names.

Line	Content
1	!This program cannot be run in DOS mode.
2	iRichu
3	.text
4	.rdata
5	@.data
6	.ndata
7	.rsrc
8	5Xp@
9	.s495
10	5'r@
11	tBj\V
12	uv9]
13	t—9]
14	tDH;
15	PShr
16	jHjZ
17	t=9]
18	t—j"
19	pcw.

Learn **karton** with the **karton-playground**



Automated unpacking with **malduck**



# Malduck

- Malduck is a swiss army knife for malware researchers
  - Cryptographic functions
  - Compression algorithms
  - Hashing functions
  - Bit-twiddling hacks
  - String utilities
  - Memory model objects (unified view of PE, ELF, IDA, cuckoo or raw files, etc)
  - Binary extraction engine
- It's generic enough to be used everywhere, but developed with malware in mind

# Malduck

- Malduck is a swiss army knife for malware researchers
  - Cryptographic functions
  - Compression algorithms
  - Hashing functions
  - Bit-twiddling hacks
  - String utilities
  - **Memory model objects** (unified view of PE, ELF, IDA, cuckoo or raw files, etc)
  - **Binary extraction engine**
- It's generic enough to be used everywhere, but developed with malware in mind



# Malduck

- Module file structure:

```
$ fdfind . modules
modules/__init__.py
modules/citadel
modules/citadel/__init__.py
modules/citadel/citadel.py
modules/citadel/citadel.yar
```

```
from .citadel import Citadel
```

# Malduck Framework

```
rule citadel : zeus
{
  meta:
    author = "mak"
    module = "citadel"
  strings:
    $briankerbs = "Coded by BRIAN KREBS for personal use only. I love my job & wife."

    $cit_aes_xor = { 81 30 [4] 0F B6 50 03 0F B6 78 02 81 70 04 [4] 81 70 08 [4] 81
                    70 0C [4] C1 E2 08 0B D7 }
    $cit_salt = { 8A D1 80 E2 07 C0 E9 03 47 83 FF 04 }
    $cit_login = { 30 [1-2] 8A 8? [4] 32 }
    $cit_getpes = { 68 [2] 00 00 8D ( 84 24 | 85) [4] 50 8D ( 85 ?? ?? ?? ?? | 44 24 ?? )
                   50 E8 [4] B8 [2] 00 00 50 68 }
    $cit_base_off = { 5? 8D 85 [4] E8 [4] 6A 20 68 [4] 8D [2] 50 E8 [4] 8D 85 [4] 50 }

  condition:
    3 of them
}
```

# Malduck Framework

```
import logging

from malduck.extractor import Extractor

log = logging.getLogger()

class Citadel(Extractor):
    family = "citadel"
    yara_rules = "citadel",
    overrides = ["zeus"]

    @Extractor.extractor("briankerbs")
    def citadel_found(self, p, addr):
        log.info('[+] `Coded by Brian Krebs` str @ %X' % addr)
        return {'family': 'citadel'}

    @Extractor.extractor
    def cit_salt(self, p, addr):
        salt = p.uint32v(addr - 8)
        log.info('[+] Found salt @ %X - %X' % (addr, salt))
        return {'salt': salt}
```

# Malduck Framework

- Install malduck config extractor from pypi
- Download and extract modules
- Start the extractor

```
$ python3 -m venv venv
$ source ./venv/bin/activate.fish
$ pip install karton-config-extractor
$
$ wget https://github.com/CERT-Polska/training-mwdb/raw/main/modules.7z
$ 7z x modules.7z
$ karton-config-extractor --modules modules/
```

# Malduck Framework

- Install malduck config extractor from pypi
- Download and extract modules
- Start the extractor

```
$ python3 -m venv venv
$ source ./venv/bin/activate.fish
$ pip install karton-config-extractor
$
$ wget https://github.com/CERT-Polska/training-mwdb/raw/main/modules.7z
$ 7z x modules.7z
$ karton-config-extractor --modules modules/
```

```
$ karton-config-extractor --modules modules/
[2021-04-15 16:30:14,998][INFO] Service karton.config-extractor started
/home/msm/Projects/karton-playground/venv/lib/python3.8/site-packages/karton/core/logger.py:57:
  warnings.warn("There is no active log consumer to receive logged messages.")
[2021-04-15 16:30:15,001][INFO] Binding on: {'type': 'sample', 'stage': 'recognized', 'kind': '
[2021-04-15 16:30:15,001][INFO] Binding on: {'type': 'sample', 'stage': 'recognized', 'kind': '
[2021-04-15 16:30:15,001][INFO] Binding on: {'type': 'sample', 'stage': 'recognized', 'kind': '
[2021-04-15 16:30:15,001][INFO] Binding on: {'type': 'analysis', 'kind': 'drakrun-prod'}
[2021-04-15 16:30:15,001][INFO] Binding on: {'type': 'analysis', 'kind': 'drakrun'}
[2021-04-15 16:30:22,000][INFO] Received new task - cecf2494-07b7-45a0-97c7-012bab4fe74f
[2021-04-15 16:30:22,002][INFO] Analyzing original binary
[2021-04-15 16:30:22,213][INFO] Got config: {"salt": 4073311727, "family": "citadel"}
[2021-04-15 16:30:22,215][INFO] Task done - cecf2494-07b7-45a0-97c7-012bab4fe74f
```

# Malduck Framework

- Stretch challenge: extend the module with extractor for **login\_key**
  - It's very similar to the `cit_salt`
  - But instead of **`addr - 8`** use **`addr + 5`**
  - And read a string from that address (with `p.asciiz(addr)`)
  - Why? Ask a reverse-engineer :).

# Malduck Framework

- Stretch challenge: extend the module with extractor for **login\_key**
  - It's very similar to the `cit_salt`
  - But instead of **`addr - 8`** use **`addr + 5`**
  - And read a string from that address (with `p.asciiz(addr)`)
  - Why? Ask a reverse-engineer :).
- Intended solution:

```
@Extractor.extractor
def cit_login(self, p, addr):
    log.info('[+] Found login_key xor @ %X' % addr)

    hit = p.uint32v(addr + 5)
    if p.is_addr(hit):
        return {'login_key': p.asciiz(hit)}
```

# Q & A

<https://github.com/CERT-Polska/>  
<https://mwdb.readthedocs.io/>  
<https://karton-core.readthedocs.io/en/latest/>  
<https://malduck.readthedocs.io/>  
<https://mwdb.cert.pl/>  
<https://www.cert.pl/en/>

**pawel.srokosz@cert.pl**  
**jaroslaw.jedynak@cert.pl**  
**pawel.pawlinski@cert.pl**  
**info@cert.pl**





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