Build Your Own Malware Analysis Pipeline Using New Open Source Tools

Paweł Srokosz
Jarosław Jedynak
Paweł Pawliński

CERT.PL

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

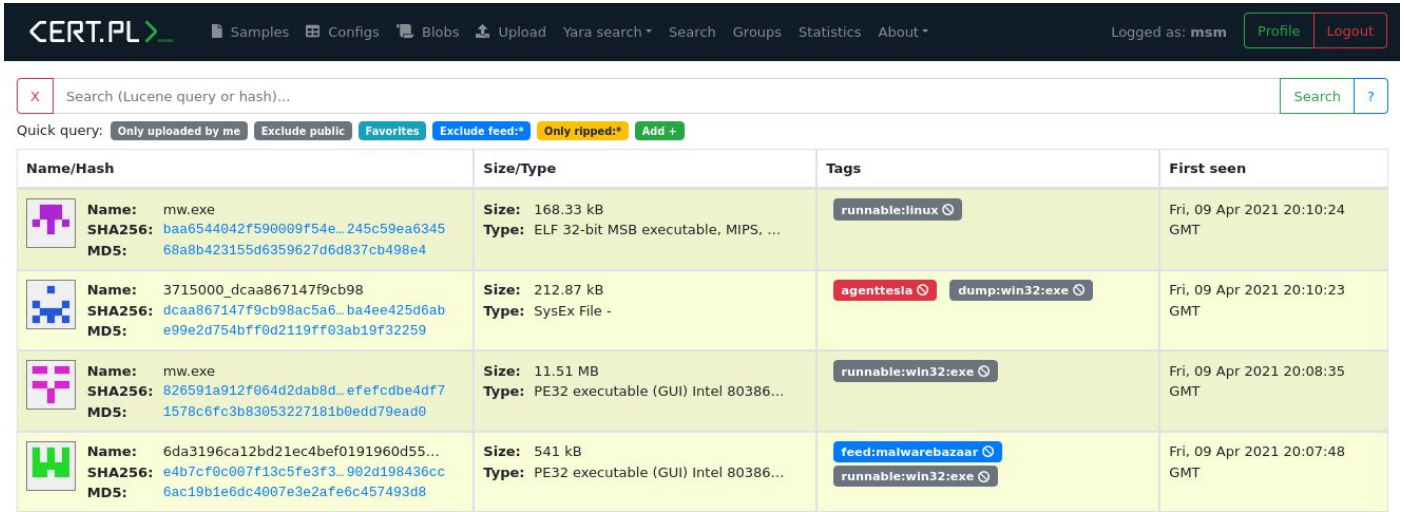
Open a terminal and check if these tools are installed:

- `$ python3 -m pip`
- `$ git`
- `$ docker-compose`
  https://docs.docker.com/engine/install/ubuntu/
  https://docs.docker.com/compose/install/

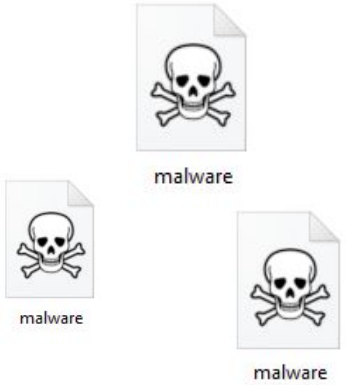
mwdb.cert.pl

What the heck is MWDB?

Introduction to **the interface**

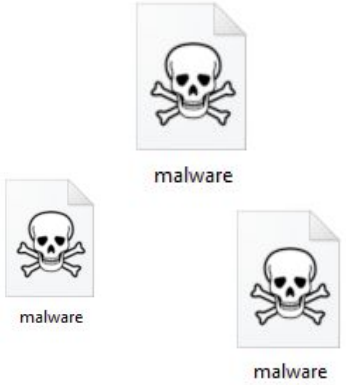


mwdb.cert.pl



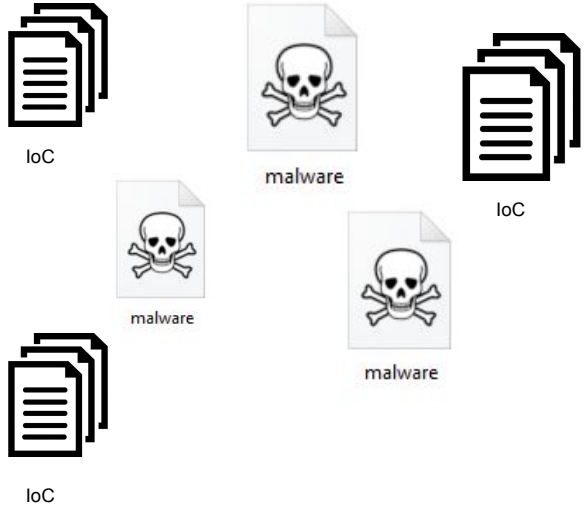
mwdb.cert.pl

Malware reverse engineer



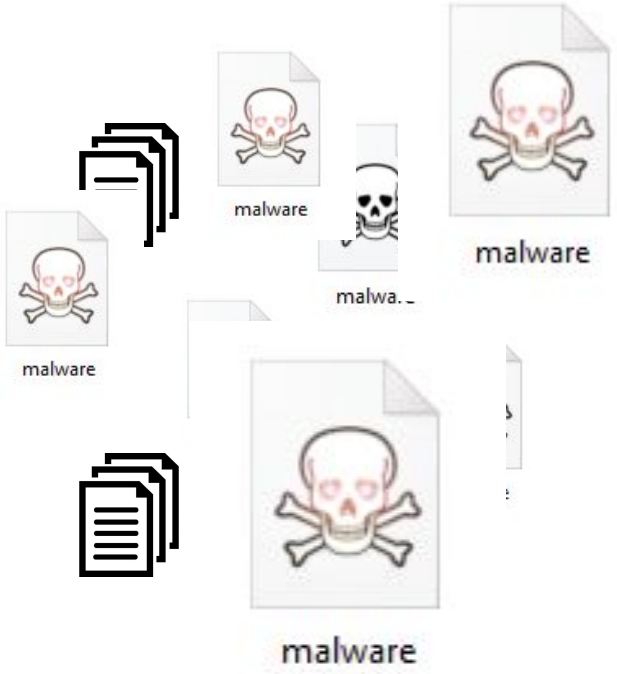
mwdb.cert.pl

Malware reverse engineer



mwdb.cert.pl

Malware reverse engineer

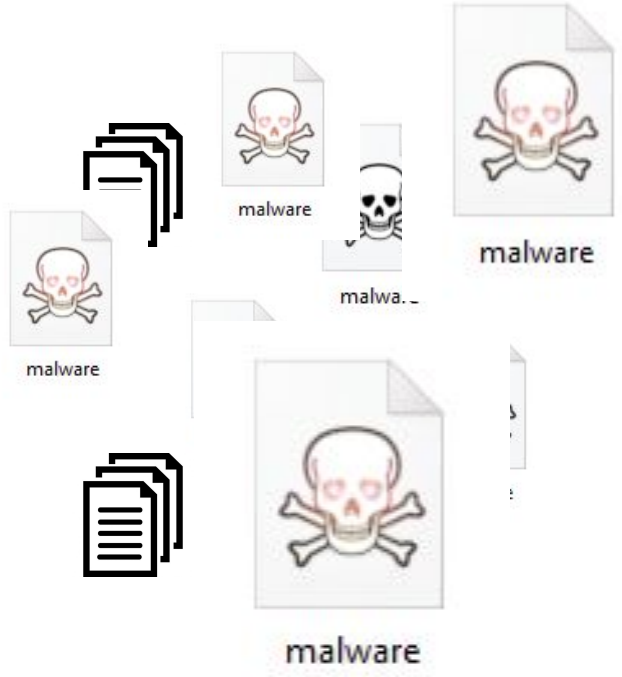


mwdb.cert.pl

researcher.py

researcher.py

researcher.py

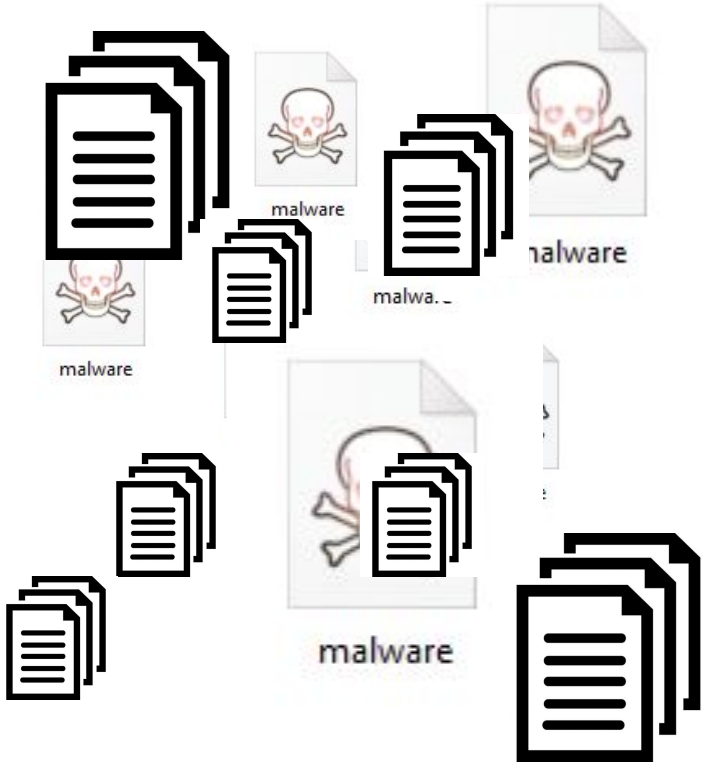


mwdb.cert.pl

researcher.py

researcher.py

researcher.py



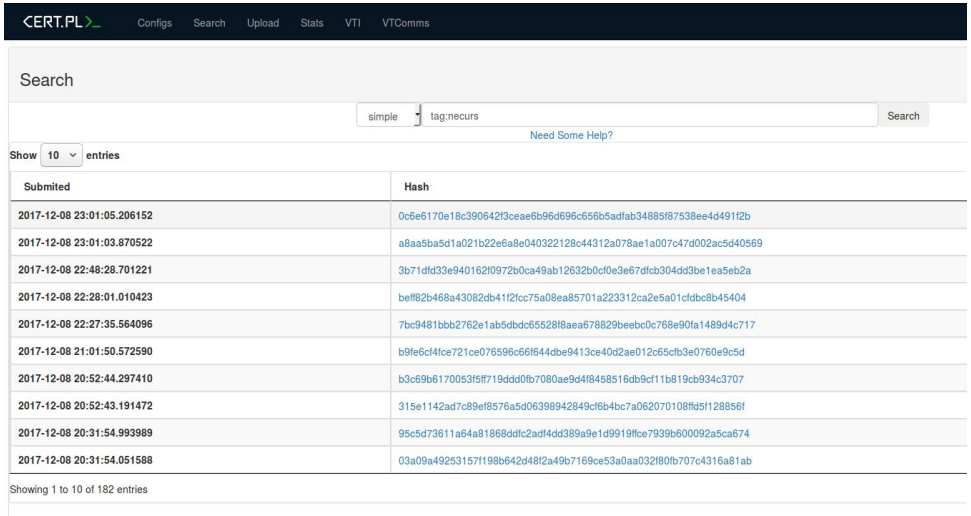
mwdb.cert.pl

researcher.py

researcher.py

researcher.py

MalWare DataBase



CERT.PL

mwdb.cert.pl

researcher.py

researcher.py

researcher.py

MalWareDB

CERT.PL Configs Search Upload Stats VTI VTComms

Search

simple tag:necurs Search

Need Some Help?

Show 10 entries

Submited	Hash
2017-12-08 23:01:05.206152	0c6e6170e18c390642f3ceae6b96d696c656b5adfab34885f87538ee4d491f2b
2017-12-08 23:01:03.870522	a8aa5ba5d1a021b22e6a8e040322128c44312a078ae1a007c47d002ac5d40569
2017-12-08 22:48:28.701221	3b71dfd33e940162f0972b0ca49ab12632b0cf0e3e67dfcb304dd3be1ea5eb2a
2017-12-08 22:28:01.010423	beff82b468a43082db41f2fcc75a08ea85701a223312ca2e5a01cfdbc8b45404
2017-12-08 22:27:35.564096	7bc9481bbb2762e1ab5dbdc65528f8aea678829beebc0c768e90fa1489d4c717
2017-12-08 21:01:50.572590	b9fe6cf4fce721ce076596c66f644dbe9413ce40d2ae012c65cfb3e0760e9c5d
2017-12-08 20:52:44.297410	b3c69b6170053f5ff719ddd0fb7080ae9d4f8458516db9cf11b819cb934c3707
2017-12-08 20:52:43.191472	315e1142ad7c89ef8576a5d06398942849cf6b4bc7a062070108ffd5f128856f
2017-12-08 20:31:54.993989	95c5d73611a64a81868ddfc2adf4dd389a9e1d9919ffce7939b600092a5ca674
2017-12-08 20:31:54.051588	03a09a49253157f198b642d48f2a49b7169ce53a0aa032f80fb707c4316a81ab

Showing 1 to 10 of 182 entries

CERT.PL

mwdb.cert.pl

researcher.py

researcher.py

researcher.py

MWDB

CERT.PL >_ Configs Search Upload Stats VTI VTComms

Search

simple tag:necurs Search

Need Some Help?

Show 10 entries

Submited	Hash
2017-12-08 23:01:05.206152	0c6e6170e18c390642f3ceae6b96d696c656b5adfab34885f87538ee4d491f2b
2017-12-08 23:01:03.870522	a8aa5ba5d1a021b22e6a8e040322128c44312a078ae1a007c47d002ac5d40569
2017-12-08 22:48:28.701221	3b71dfd33e940162f0972b0ca49ab12632b0cf0e3e67dfcb304dd3be1ea5eb2a
2017-12-08 22:28:01.010423	beff82b468a43082db41f2fcc75a08ea85701a223312ca2e5a01cfdbc8b45404
2017-12-08 22:27:35.564096	7bc9481bbb2762e1ab5dbdc65528f8aea678829beebc0c768e90fa1489d4c717
2017-12-08 21:01:50.572590	b9fe6cf4fce721ce076596c66f644dbe9413ce40d2ae012c65cfb3e0760e9c5d
2017-12-08 20:52:44.297410	b3c69b6170053f5ff719ddd0fb7080ae9d4f8458516db9cf11b819cb934c3707
2017-12-08 20:52:43.191472	315e1142ad7c89ef8576a5d06398942849cf6b4bc7a062070108ffd5f128856f
2017-12-08 20:31:54.993989	95c5d73611a64a81868ddfc2adf4dd389a9e1d9919ffce7939b600092a5ca674
2017-12-08 20:31:54.051588	03a09a49253157f198b642d48f2a49b7169ce53a0aa032f80fb707c4316a81ab

Showing 1 to 10 of 182 entries

CERT.PL >_

mwdb.cert.pl

MWDB

The Interface

CERT.PL >_ Configs Search Upload Stats VTI VTComms

Search

simple tag:necurs Search

Need Some Help?

Show 10 entries

Submited	Hash
2017-12-08 23:01:05.206152	0c6e6170e18c390642f3ceae6b96d696c656b5adfab34885f87538ee4d491f2b
2017-12-08 23:01:03.870522	a8aa5ba5d1a021b22e6a8e040322128c44312a078ae1a007c47d002ac5d40569
2017-12-08 22:48:28.701221	3b71dfd33e940162f0972b0ca49ab12632b0cf0e3e67dfcb304dd3be1ea5eb2a
2017-12-08 22:28:01.010423	beff82b468a43082db41f2fcc75a08ea85701a223312ca2e5a01cfdbc8b45404
2017-12-08 22:27:35.564096	7bc9481bbb2762e1ab5dbdc65528f8aea678829beebc0c768e90fa1489d4c717
2017-12-08 21:01:50.572590	b9fe6cf4fce721ce076596c66f644dbe9413ce40d2ae012c65cfb3e0760e9c5d
2017-12-08 20:52:44.297410	b3c69b6170053f5ff719ddd0fb7080ae9d4f8458516db9cf11b819cb934c3707
2017-12-08 20:52:43.191472	315e1142ad7c89ef8576a5d0639894 2649cf6b4bc7a062070108ffd5f128856f
2017-12-08 20:31:54.993989	95c5d73611a64a81868ddfc2adf4dd389a9e1d9919ffce7939b600092a5ca674
2017-12-08 20:31:54.051588	03a09a49253157f198b642d48f2a49b7169ce53a0aa032f80fb707c4316a81ab

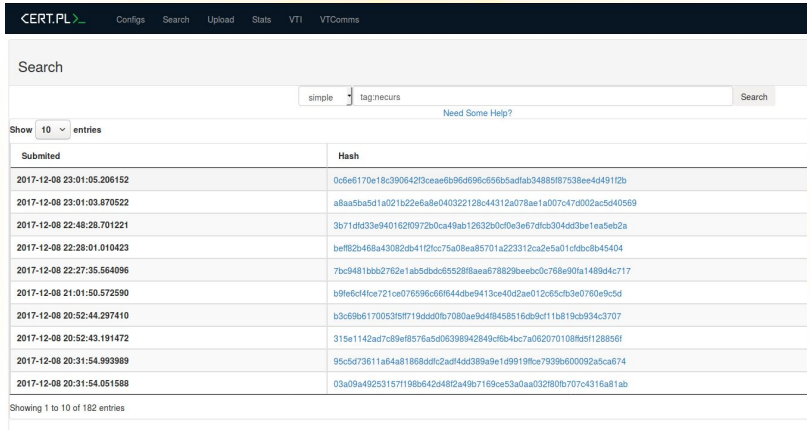
Showing 1 to 10 of 182 entries

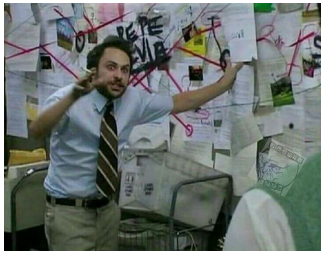
CERT.PL >_

mwdb.cert.pl

MWDB

The Interface



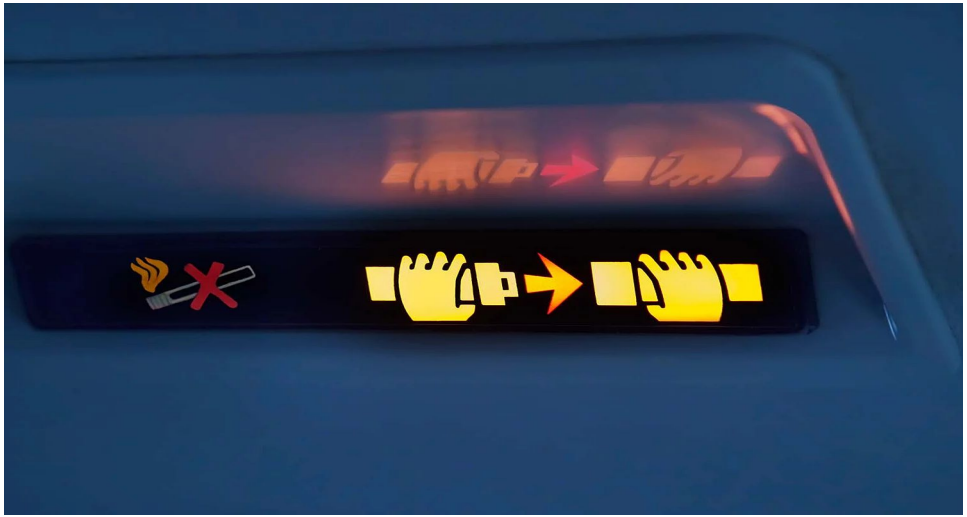


CERT.PL

mwdb.cert.pl

Tour de **mwdb.cert.pl**

https://mwdb.cert.pl



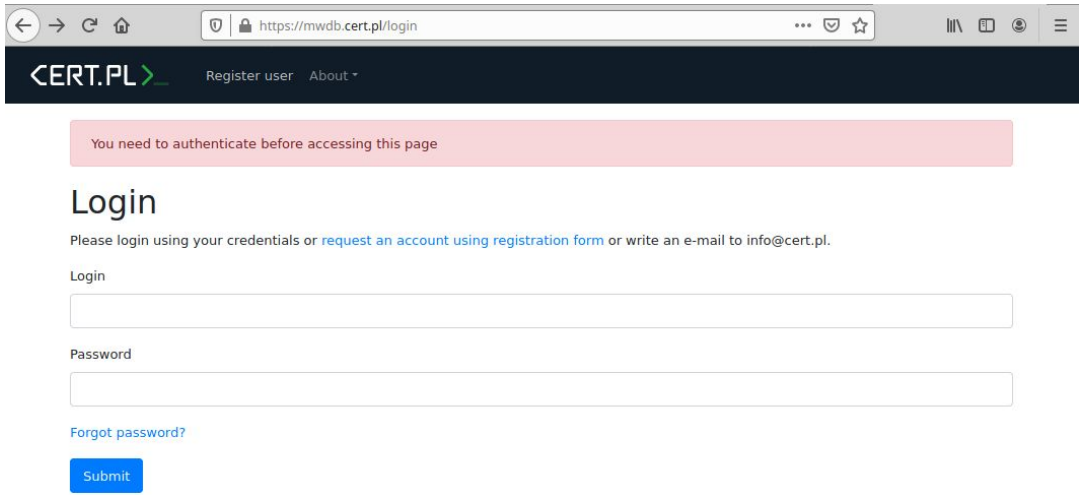
Training materials
https://github.com/CERT-Polska/training-mwdb

CERT.PL

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



Training materials
https://github.com/CERT-Polska/training-mwdb

CERT.PL

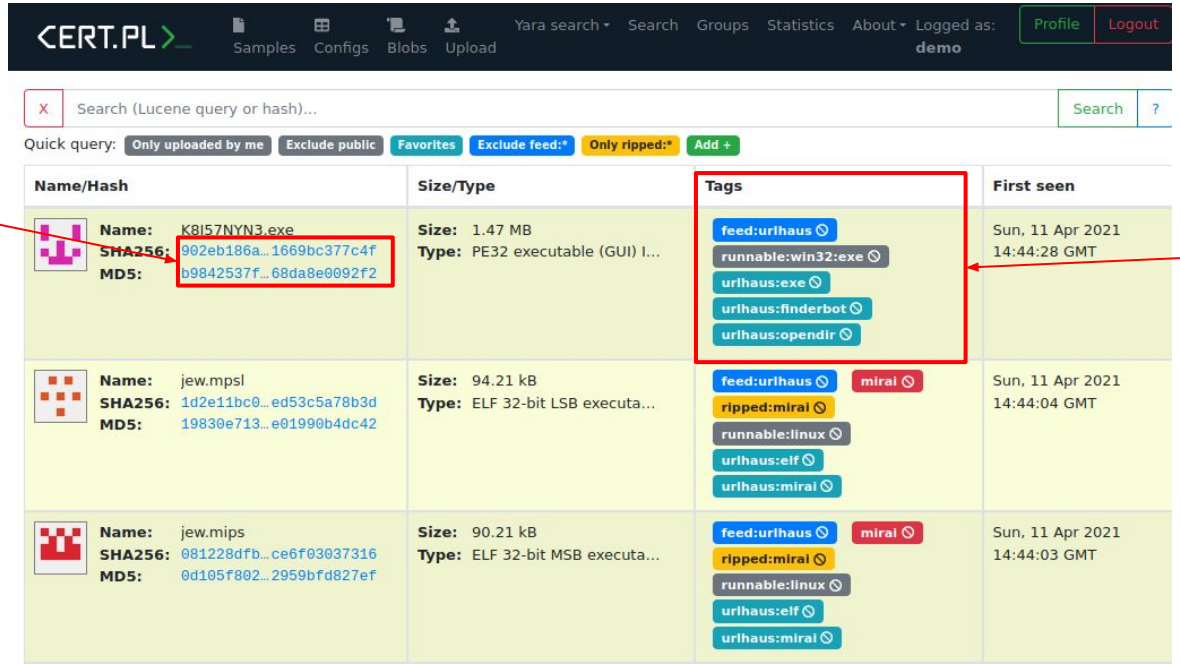
Exercise #0: Getting familiar with the interface

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

mwdb.cert.pl

Exercise #0: Getting familiar with the interface

Link to sample details



Click on tags to filter (or filter out)

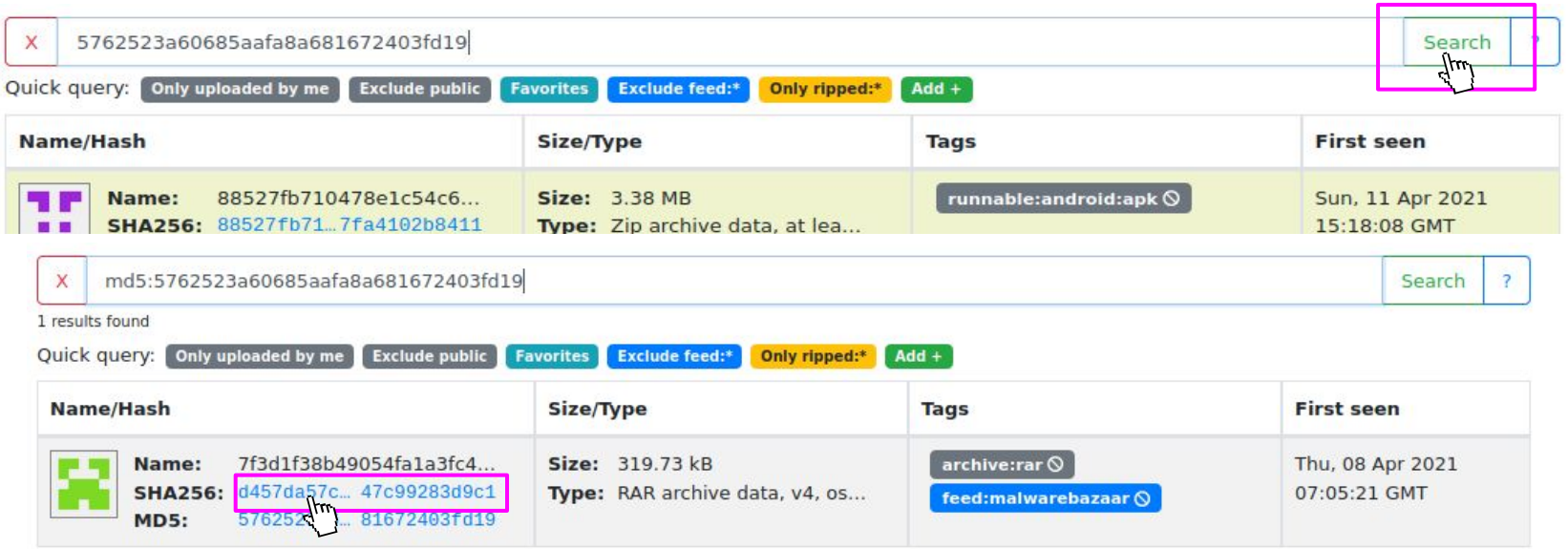
mwdb.cert.pl

Exercise #0: Getting familiar with the interface

CERT.PL

mwdb.cert.pl

Exercise #0: Getting familiar with the interface



CERT.PL

Exercise #1: Filtering samples by tags

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

mwdb.cert.pl

Exercise #1: Filtering samples by tags

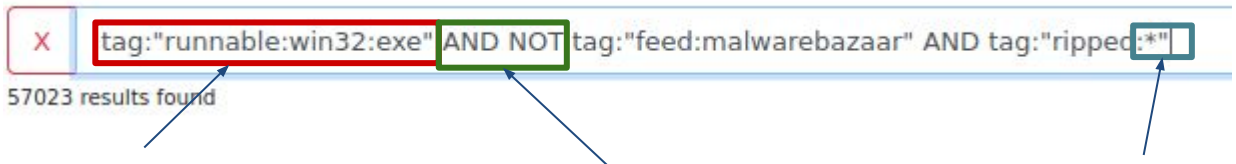
formbook	Simple tag, mostly used for marking artifacts that are associated with malware family
feed:sample	Tag describing the source of malware sample
ripped:formbook	Tag for the original sample, indicating recognized malware family
runnable:win32:exe	Tag describing the type of sample
yara:win_formbook	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

mwdb.cert.pl

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

mwdb.cert.pl

Everything related with formbook

X | tag:*formbook* | Search | ?

8472 results found

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

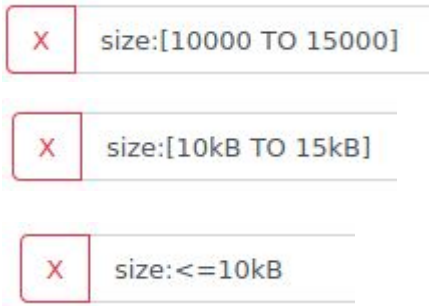
Name/Hash	Size/Type	Tags	First seen
Name: 3310000_55996a8917b... **SHA256:** 55996a891…c1b1122ca8d7 **MD5:** cedc1abf7…2d89abd544fd	**Size:** 154.75 kB **Type:** data	dump:win32:exe formbook	Sun, 11 Apr 2021 08:13:01 GMT
Name: 5f85c45be7b228140ce5... **SHA256:** b9a1fe9ef…bccca5623704 **MD5:** 430a47302…5a335d184bc4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

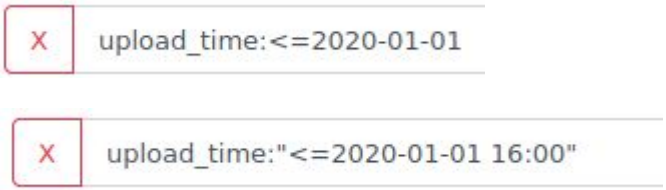
CERT.PL

mwdb.cert.pl

Exercise #1: Filtering samples by tags

Ranges



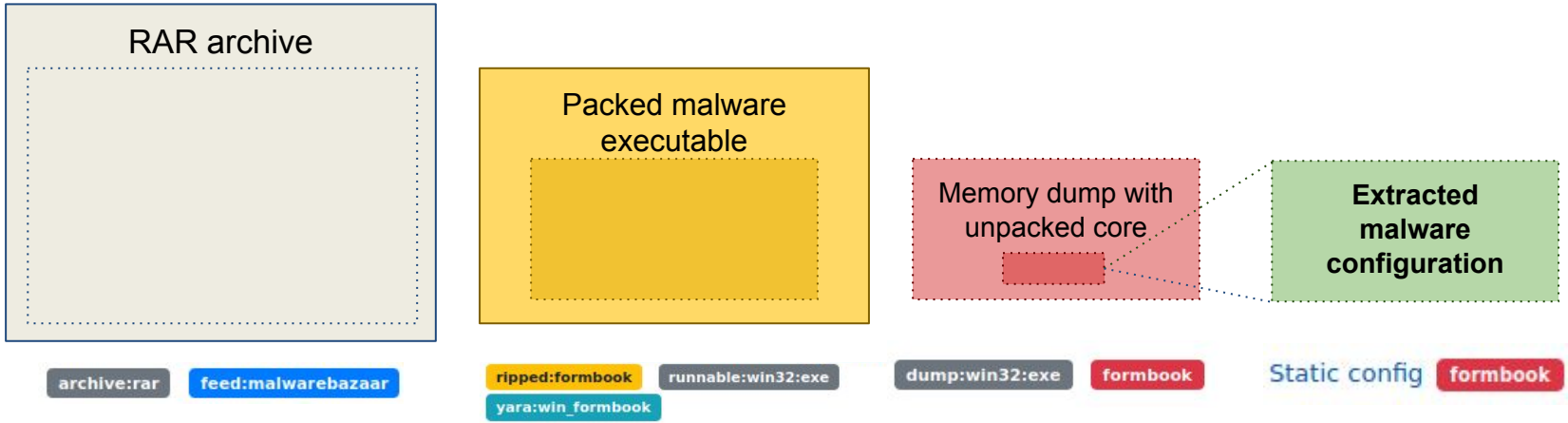


Exercise #2: Exploring sample view and hierarchy

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

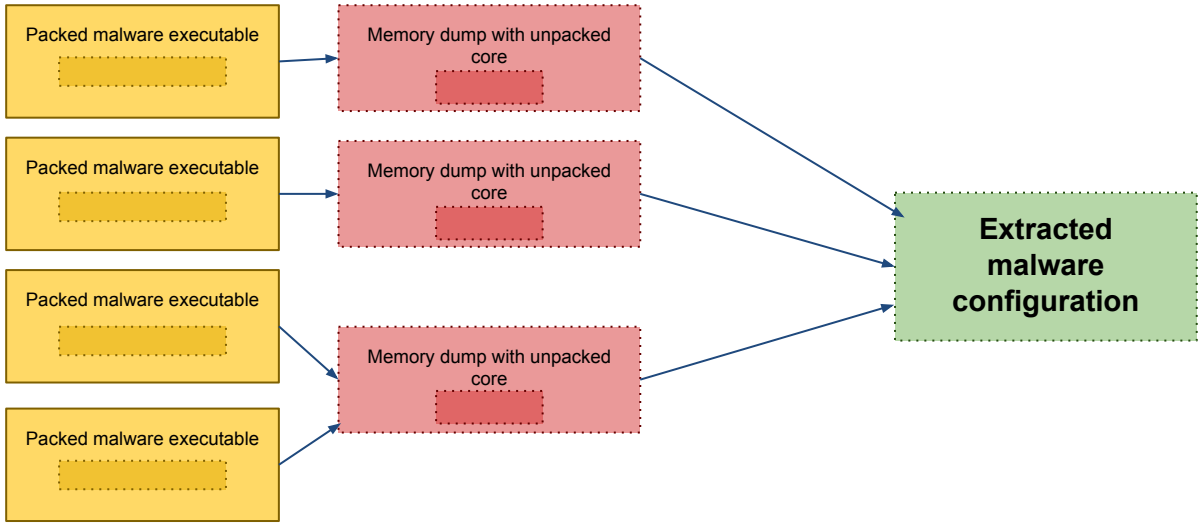
mwdb.cert.pl

Exercise #2: Exploring sample view and hierarchy



mwdb.cert.pl

Exercise #2: Exploring sample view and hierarchy



mwdb.cert.pl

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

Exercise #3: Looking for similar configurations

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

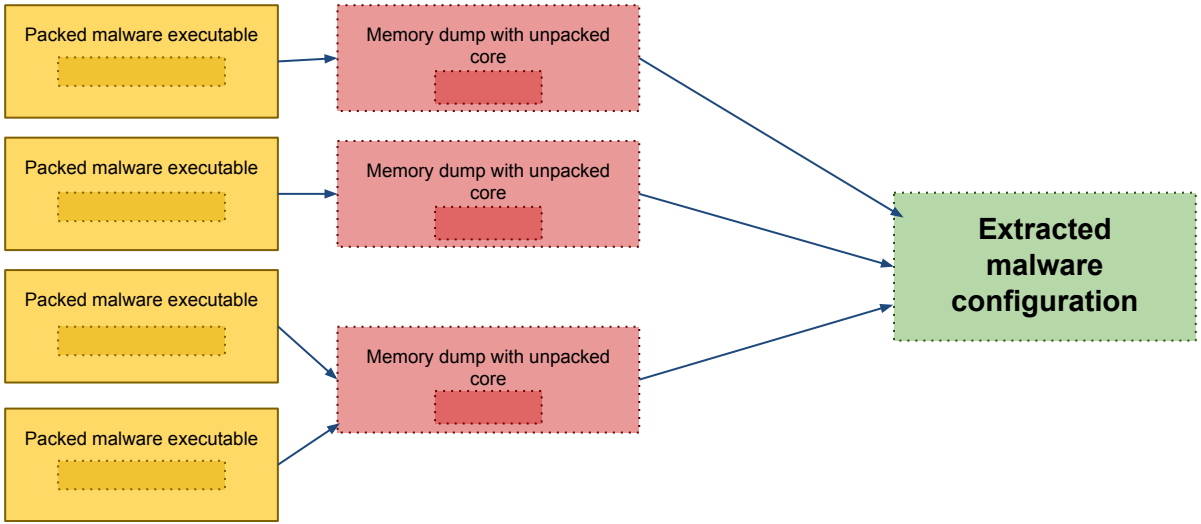
mwdb.cert.pl 

Exercise #4: Blobs and dynamic configurations

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

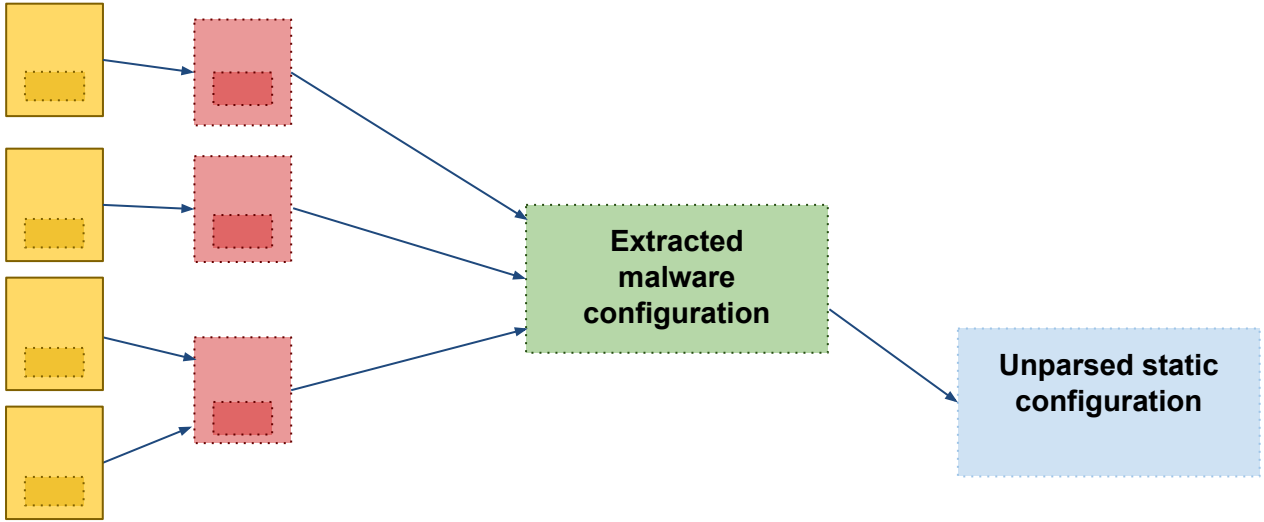
mwdb.cert.pl

Exercise #4: Blobs and dynamic configurations



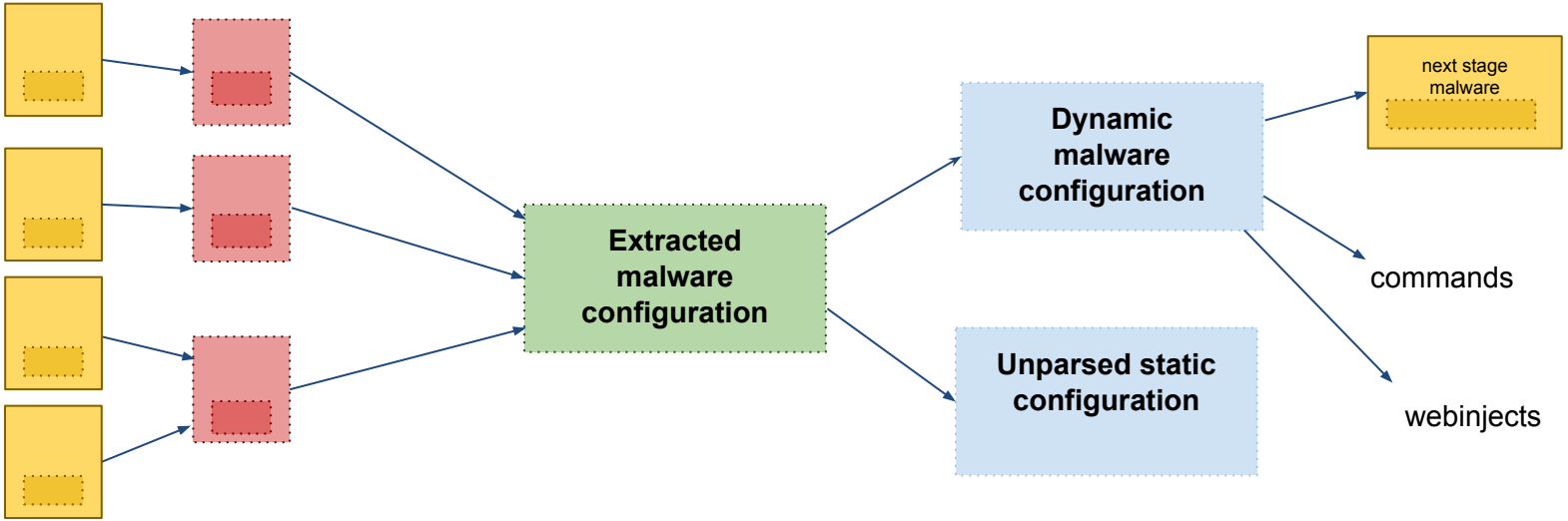
mwdb.cert.pl

Exercise #4: Blobs and dynamic configurations



mwdb.cert.pl

Exercise #4: Blobs and dynamic configurations

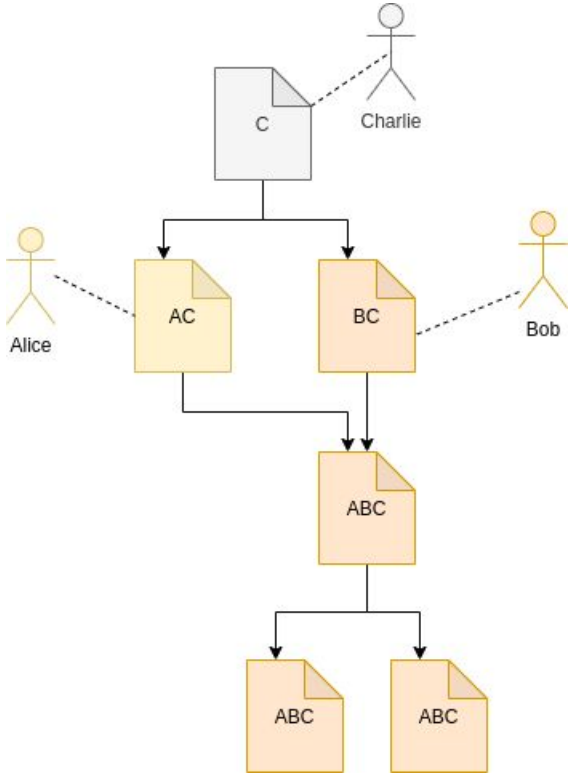


Exercise #5: Let’s upload something!

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

mwdb.cert.pl

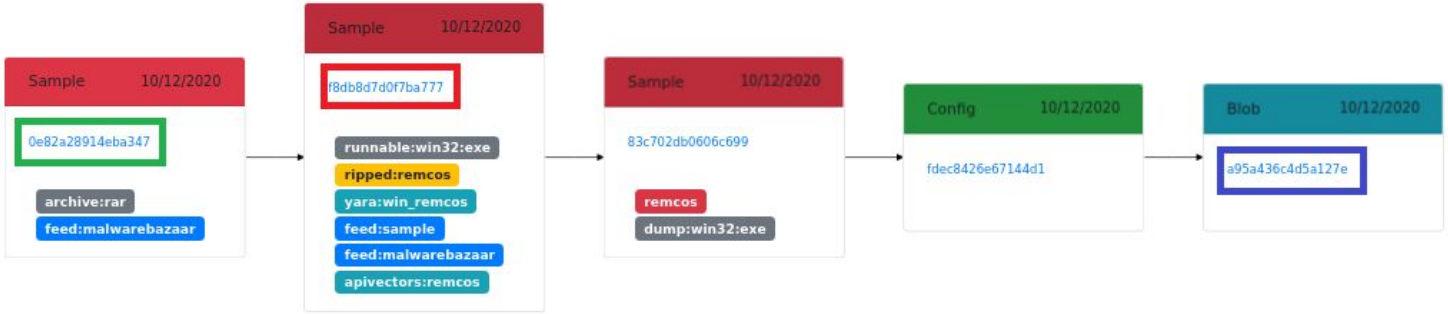
Exercise #5: Let’s upload something!



CERT.PL

mwdb.cert.pl

Exercise #5: Let's upload something!



Shares

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

mwdb.cert.pl

Scripting and automation with **mwdblib**

CERT.PL

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

Exercise #6: Get recent files

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

Exercise #7: MWDBObject properties

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

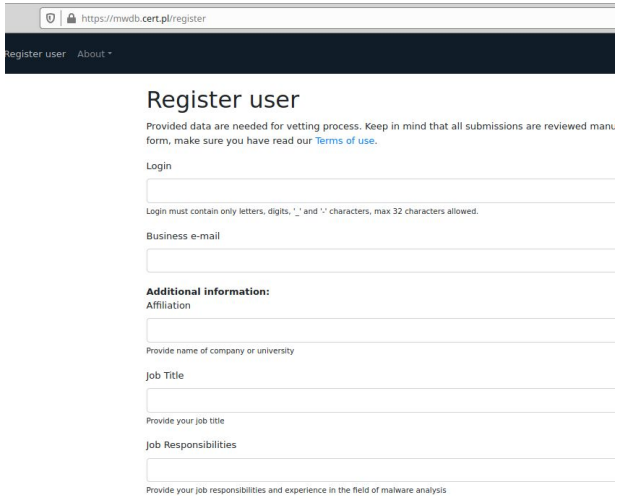
Exercise #8: Using mwdblib CLI

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

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).

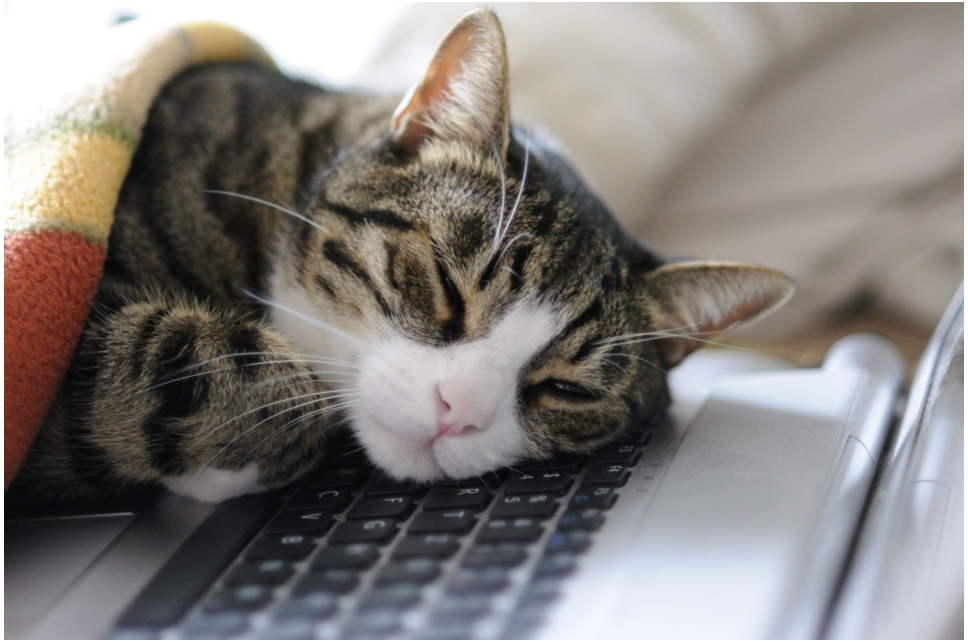
- provide some info about yourself (homepage, research blog, Twitter handle). Mention this workshop.

- don’t worry to ping us if we don’t answer for a few days

CERT.PL

mwdb.cert.pl

Coffee break



Agenda

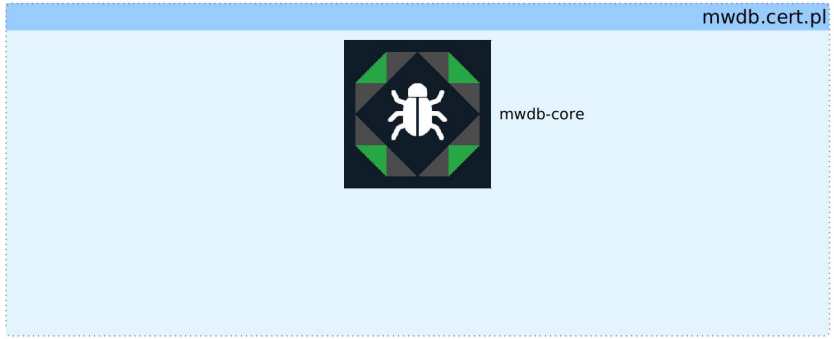
- **mwdb.cert.pl**
  - What the heck is **MWD**
  - Tour de **mwdb.cert.p**
  - Scripting and automati **mwdblib**
- **mwdb-core** and **karton**
  - Run a **self-hosted** mwdb-core and karton instances
  - Experiment with **karton-playground**
  - Distributed collaboration with mwdb **remotes**
  -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

Learn karton with the **karton-playground**

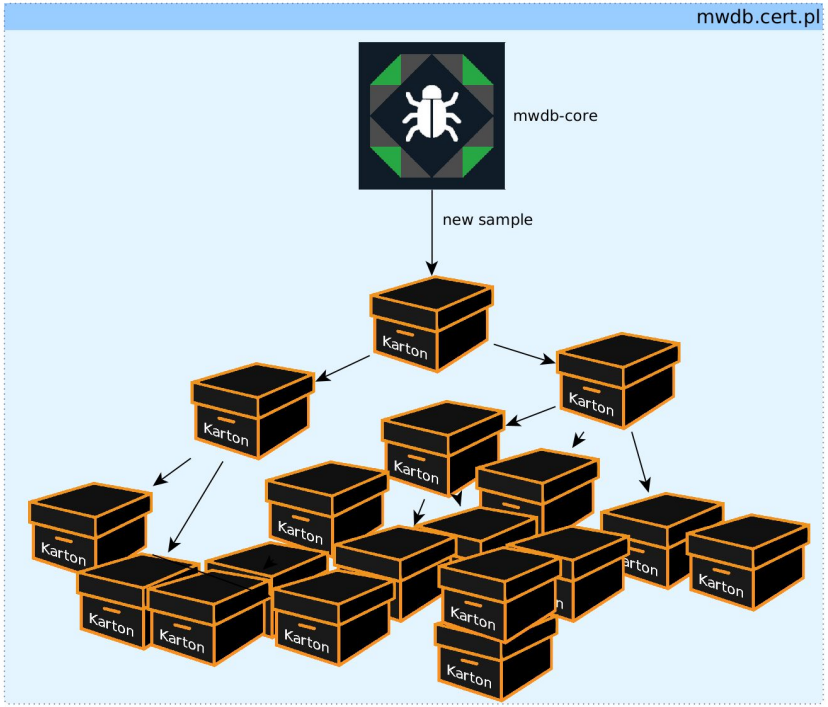
Run a **self-hosted** mwdb-core and karton instance

CERT.PL

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

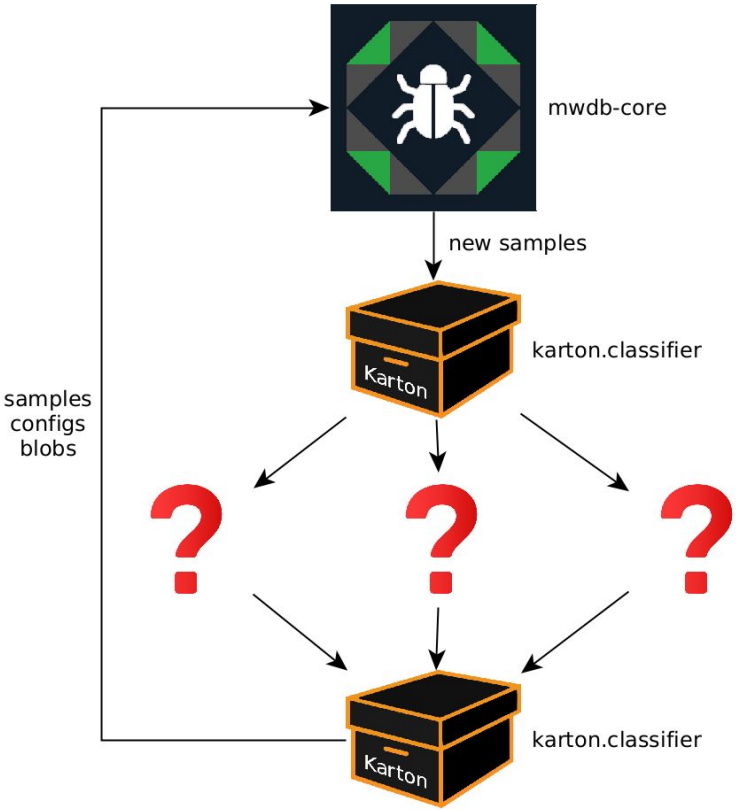
Karton Playground

```
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

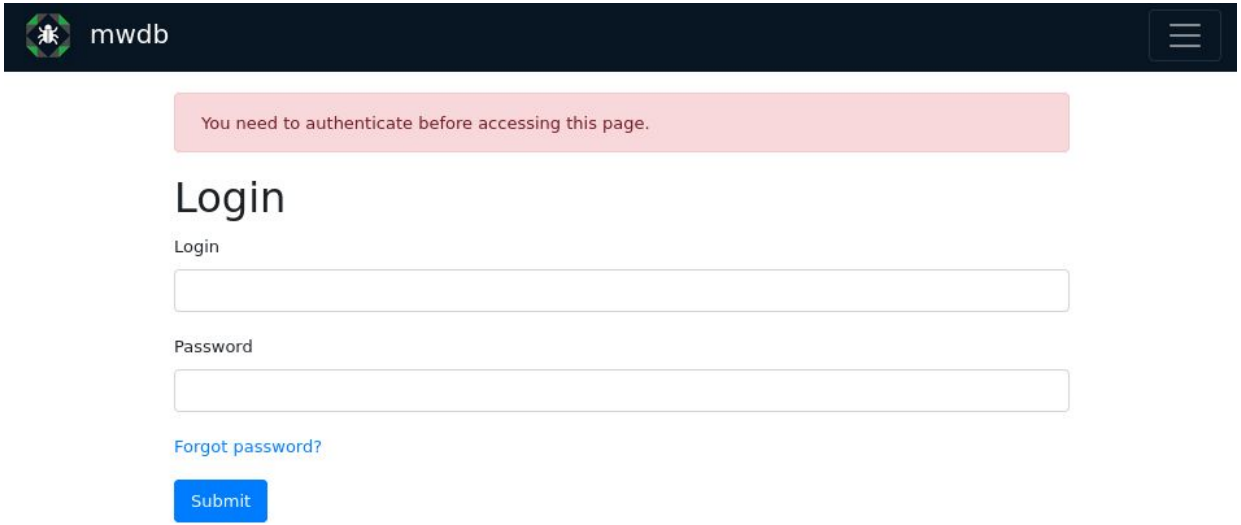
Karton Playground

While you wait...



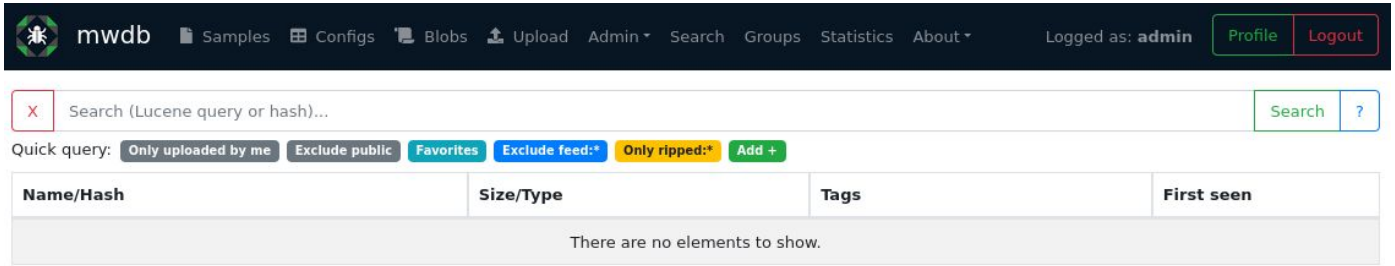
Karton Playground: click around

Navigate to http://127.0.0.1:8080. Login using `admin:admin`.



Karton Playground: click around

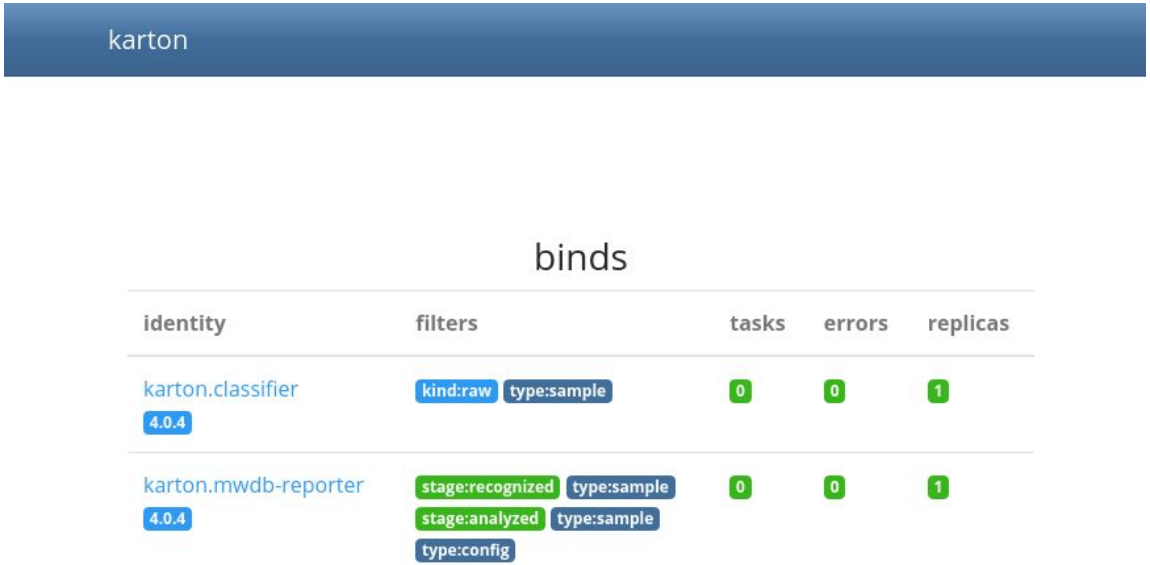
Navigate to http://127.0.0.1:8080. Login using `admin:admin`.



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

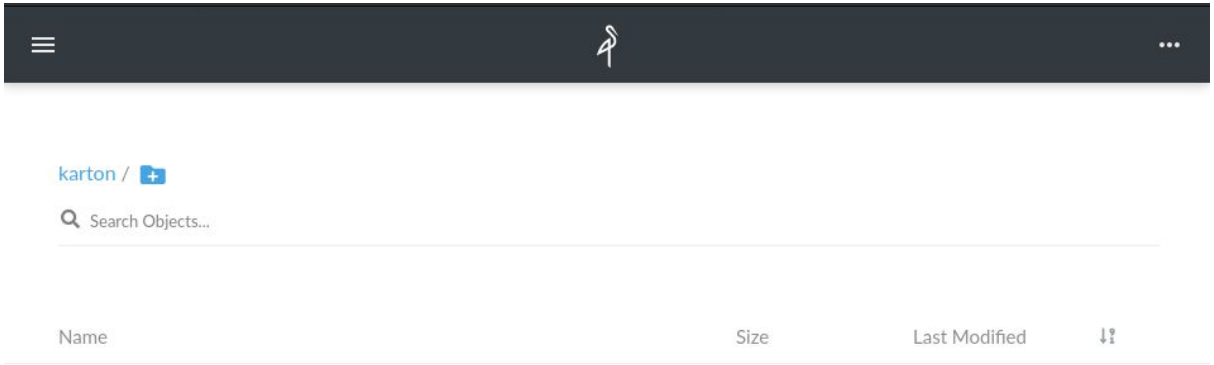
Karton Playground: click around

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



Karton 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:

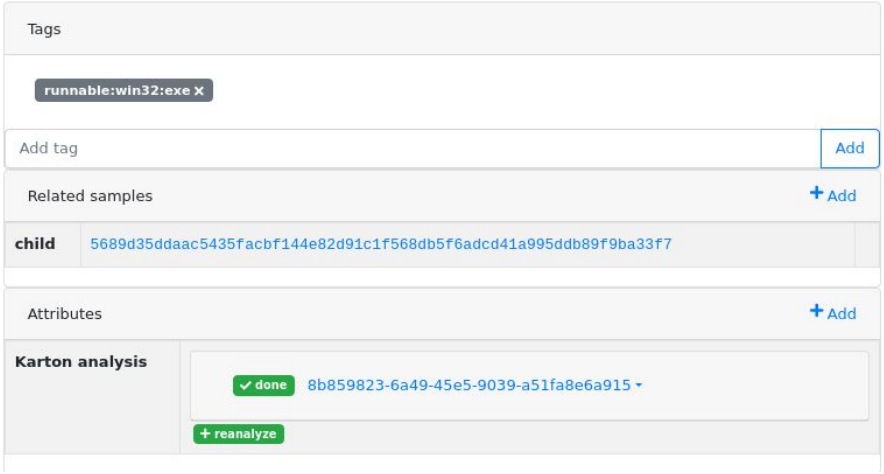
```
$ wget https://github.com/CERT-Polska/training-mwdb/blob/main/autoit-malware.bin
$ sha256sum autoit-malware.bin
a4816d4fecd6d2806d5b105c3aab55f4a1eb5deb3b126f317093a4dc4aab88a1 autoit-malware.bin
```

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!



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

```python
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
```

CERT.PL >_

Karton Playground exercise

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

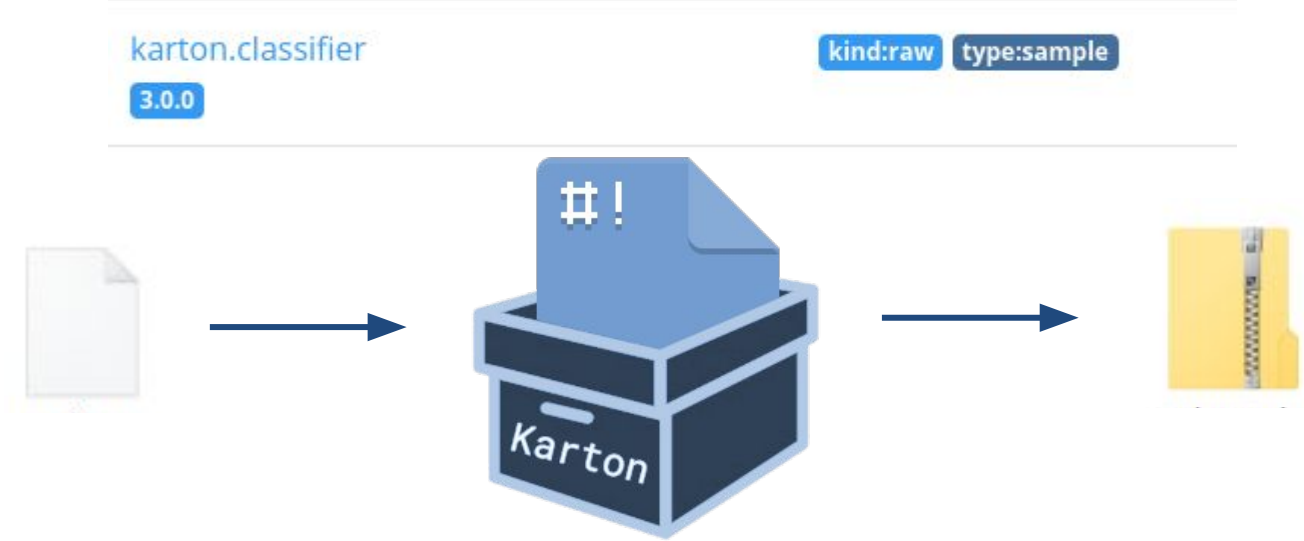
```python
class MyFirstKarton(Karton):
    identity = "karton.first"
    filters = [{"type": "sample", "stage": "recognized"}]

    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

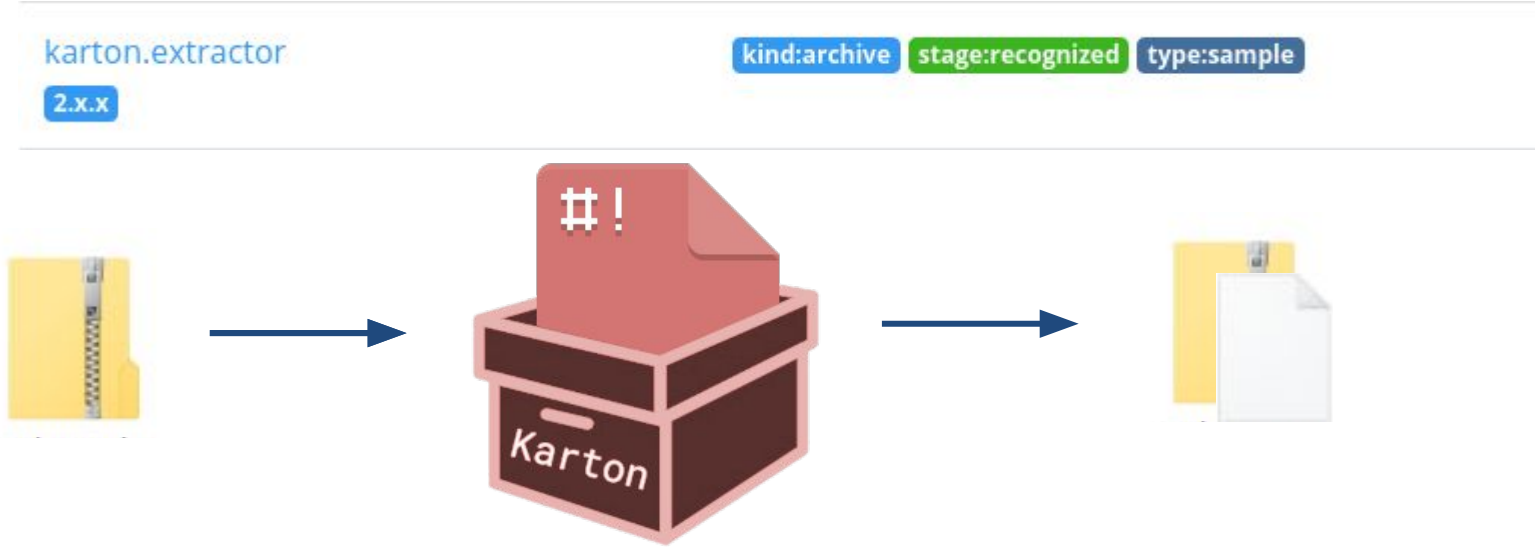
- What **are** these?

```python
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
```

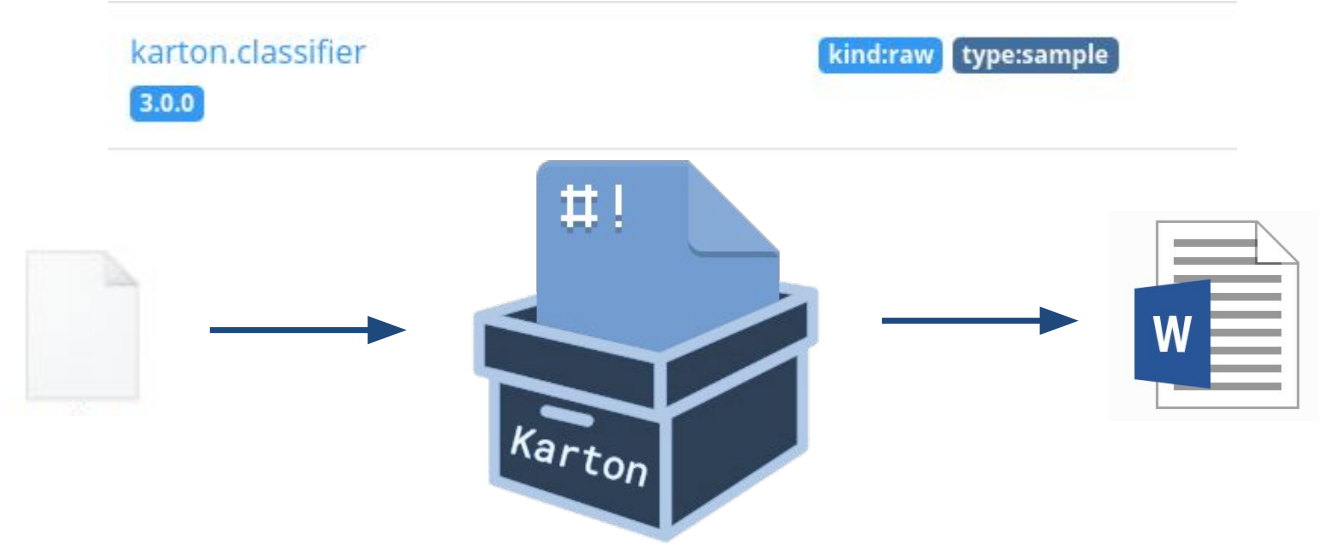
Karton pipeline



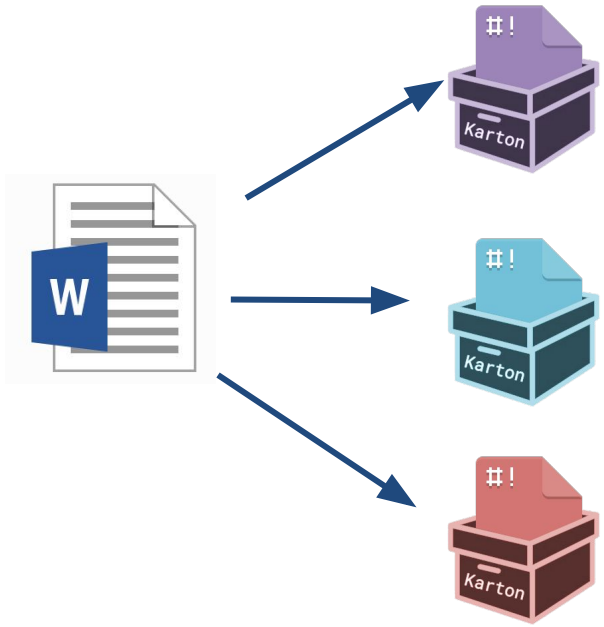
Karton pipeline



Karton pipeline



Karton pipeline



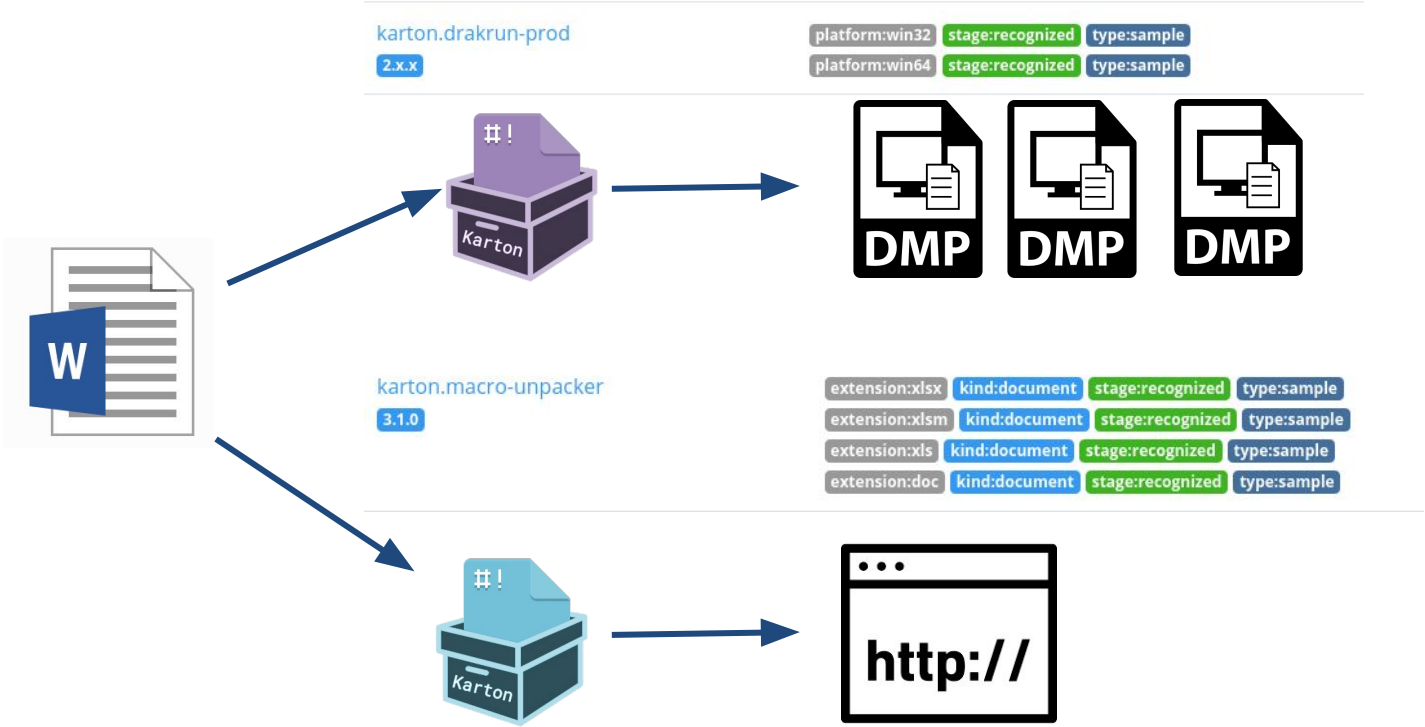
karton.cuckoo1
2.x.x
platform:win32 quality:high stage:recognized type:sample

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

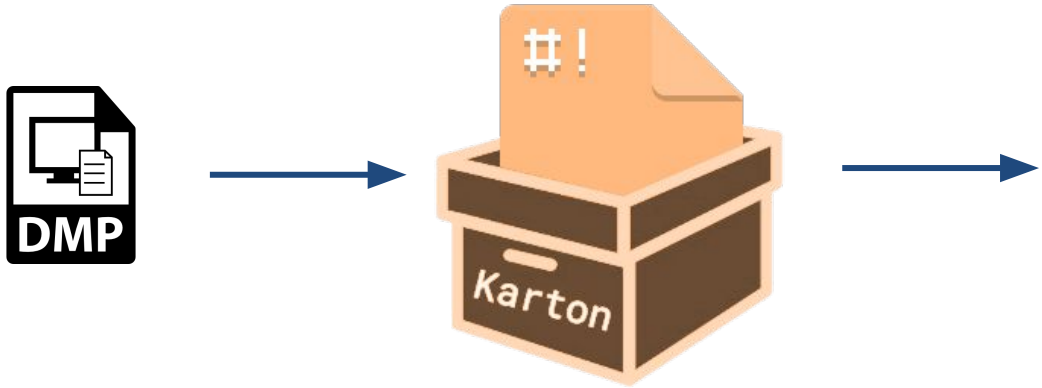
2.x.x

kind:runnable platform:win32 stage:recognized type:sample

kind:runnable platform:win64 stage:recognized type:sample

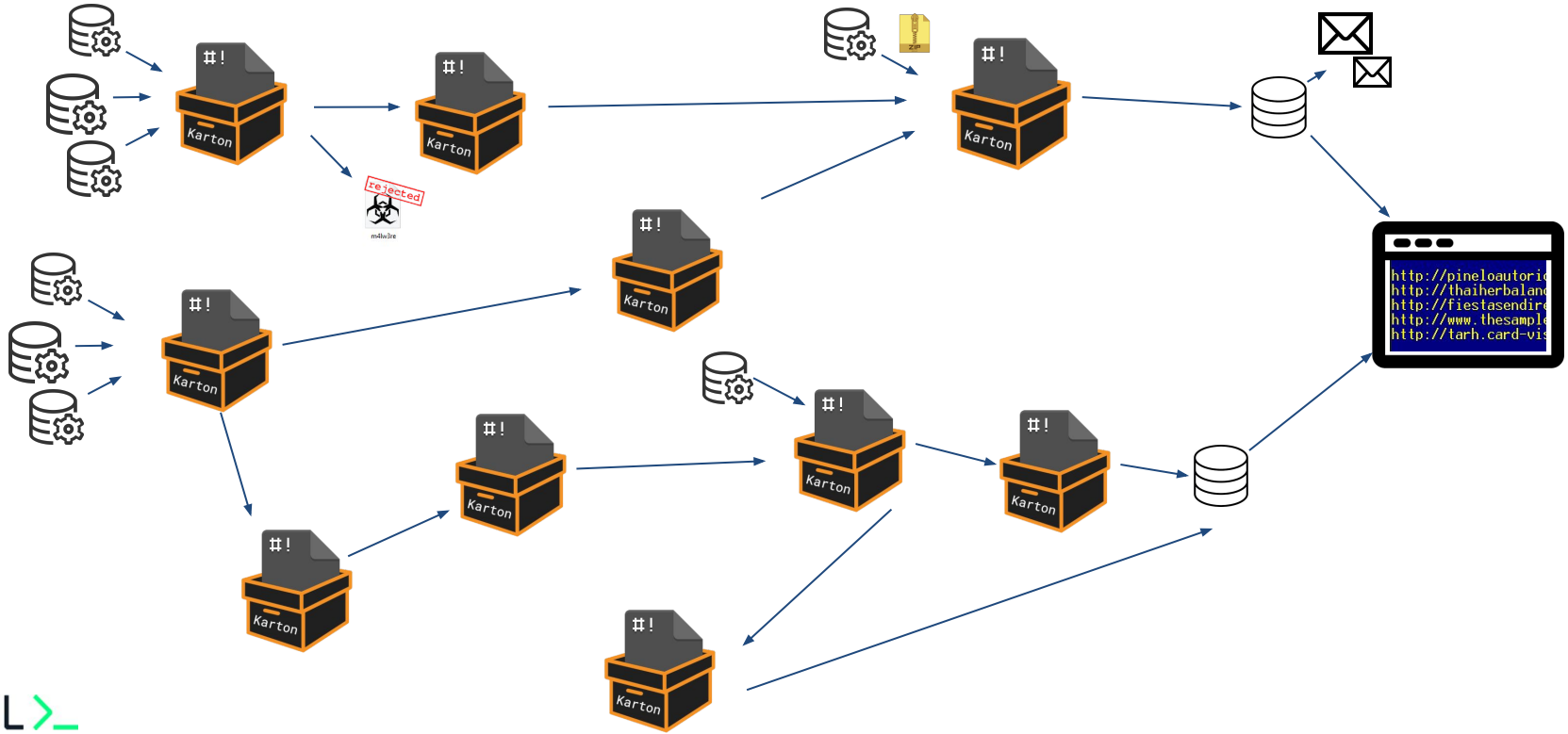
kind:runnable platform:linux stage:recognized type:sample

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



CERT.PL

Karton Playground exercise

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

```python
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
```

CERT.PL >_

Karton Playground exercise

- Karton = Consumer + Producer

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

```python
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

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

```python
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
```

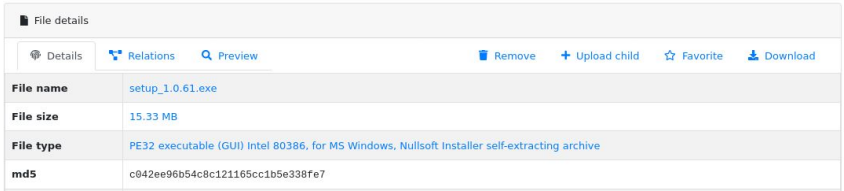
Karton Playground exercise

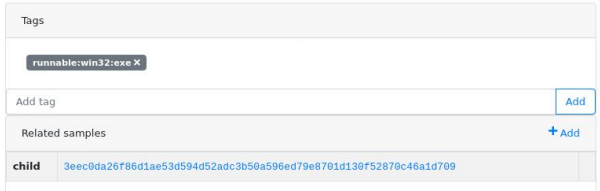
- Download a template: https://github.com/CERT-Polska/training-mwdb/blob/main/karton-template.py
- Your task: edit the template, and:
  - Run the `strings` utility on every incoming sample
  - Save the result in a variable (use `subprocess.check_output`)
  - Upload the result to mwdb (already handled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

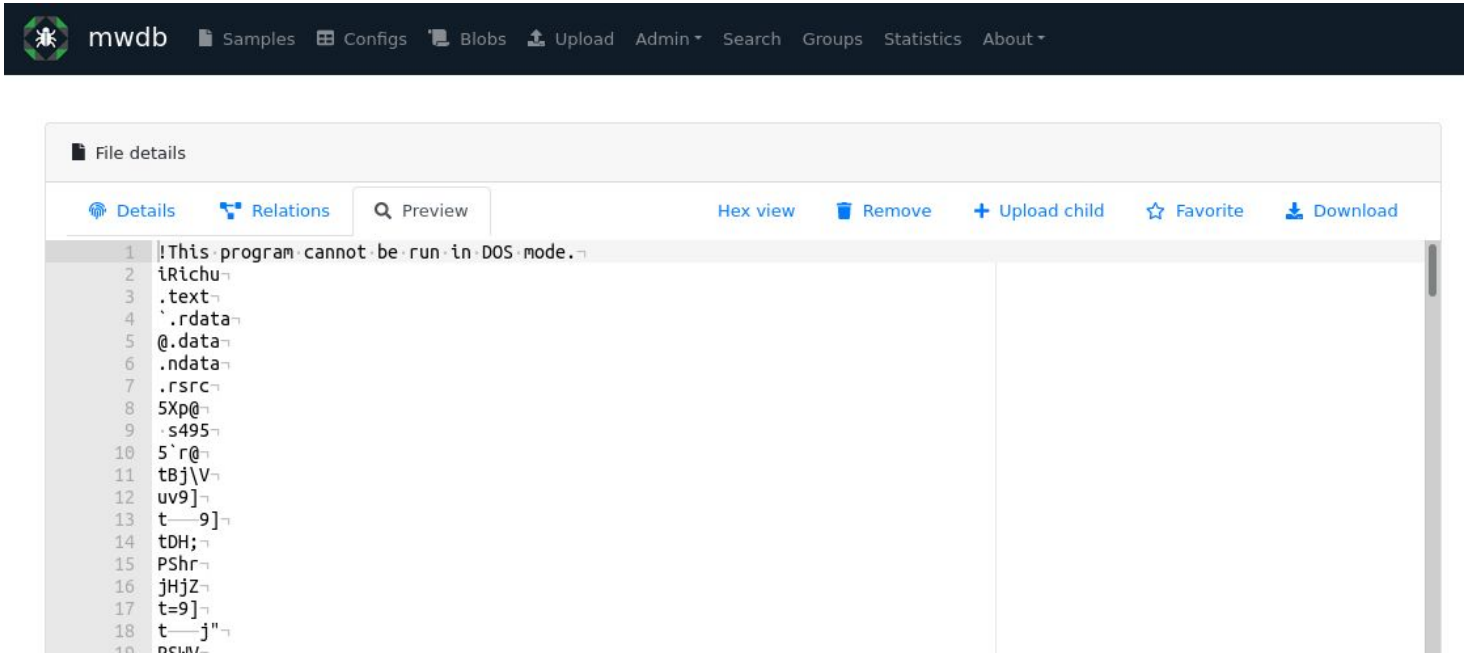
```
$ python3 karton-template.py
[2021-04-14 20:56:28,927][INFO] Service karton.first started
/home/msm/Projects/karton-playground/venv/lib/python3.8/site-packages/karton/core
  warnings.warn("There is no active log consumer to receive logged messages.")
[2021-04-14 20:56:28,928][INFO] Binding on: {'type': '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
```





CERT.PL

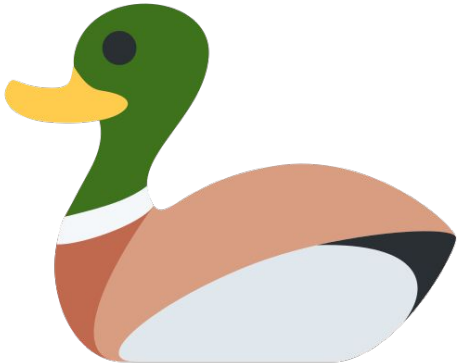
Karton Playground exercise: solution



CERT.PL >_

Learn karton with the **karton-playground**

Automated unpacking with **malduck**



CERT.PL

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

```python
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

CERT.PL