AIL Framework for Analysis of Information Leaks

Workshop - A generic analysis information leak open source software

CIRCL
Computer Incident
Response Center
Luxembourg

Sami Mokaddem
sami.mokaddem@circl.lu
Aurélien Thirion

info@circl.lu

July 2, 2018

Objectives of the workshop

Our objectives of the workshop

- Demonstrate why data-analysis is critical in information security
- Explain challenges and the design of the AIL framework
- Learn how to install and start AIL
- Learn how to properly feed AIL with custom data
- Learn how to manage current modules
- Learn how to create new modules
- Practical part

Sources of leaks

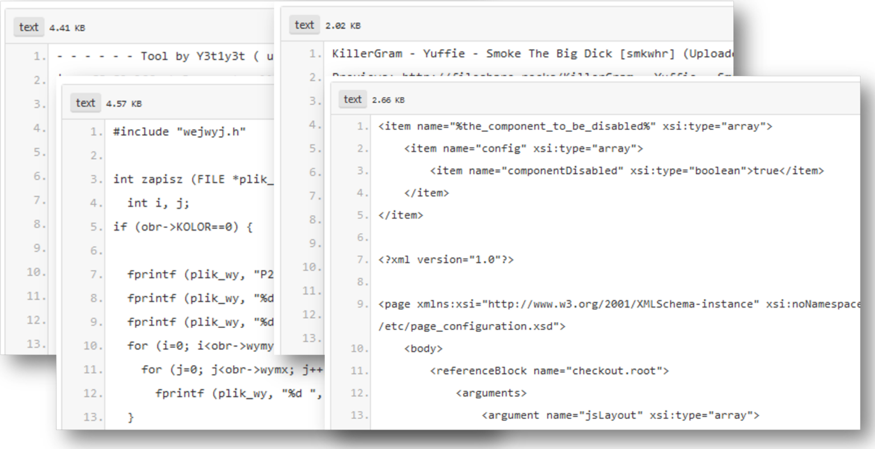
Sources of leaks: Paste monitoring

- Example: `http://pastebin.com/`
  - Easily storing and sharing text online
  - Used by programmers and legitimate users
    → Source code & information about configurations

Sources of leaks: Paste monitoring

- Example: `http://pastebin.com/`
    - Easily storing and sharing text online
    - Used by programmers and legitimate users
      → Source code & information about configurations
- Abused by attackers to store:
    - List of vulnerable/compromised sites
    - Software vulnerabilities (e.g. exploits)
    - Database dumps
      → User data
      → Credentials
      → Credit card details
    - More and more ...

Examples of pastes



```
- - - - - - Tool by Y3t1y3t ( u
```

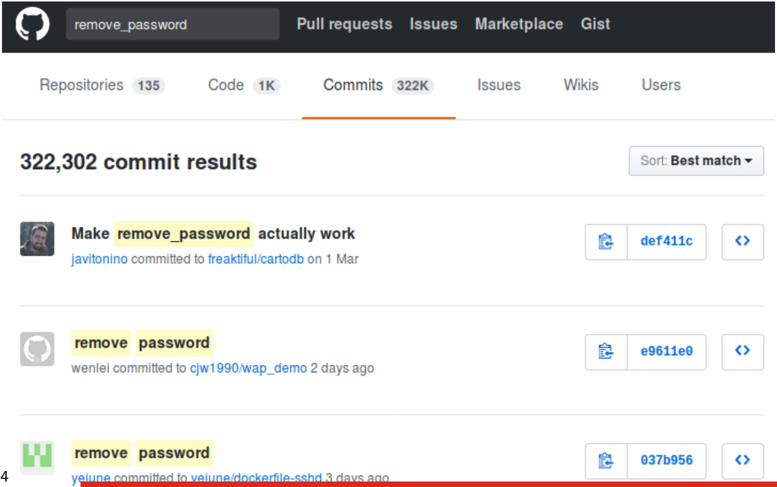
```
#include "wejwyj.h"

int zapisz (FILE *plik_
  int i, j;
if (obr->KOLOR==0) {

  fprintf (plik_wy, "P2
  fprintf (plik_wy, "%d
  fprintf (plik_wy, "%d
  for (i=0; i<obr->wymy
    for (j=0; j<obr->wymx; j++
      fprintf (plik_wy, "%d ",
  }
```

```
KillerGram - Yuffie - Smoke The Big Dick [smkwhr] (Upload
Previews: http://fileshare.rocks/KillerGram - Yuffie - Sm
```

```
<item name="%the_component_to_be_disabled%" xsi:type="array">
    <item name="config" xsi:type="array">
        <item name="componentDisabled" xsi:type="boolean">true</item>
    </item>
</item>

<?xml version="1.0"?>

<page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespace
/etc/page_configuration.xsd">
    <body>
        <referenceBlock name="checkout.root">
            <arguments>
                <argument name="jsLayout" xsi:type="array">
```

Sources of leaks: Others

- Mistakes from users
  - https://github.com/search?q=remove_password&type=Commits&ref=searchresults



322,302 commit results

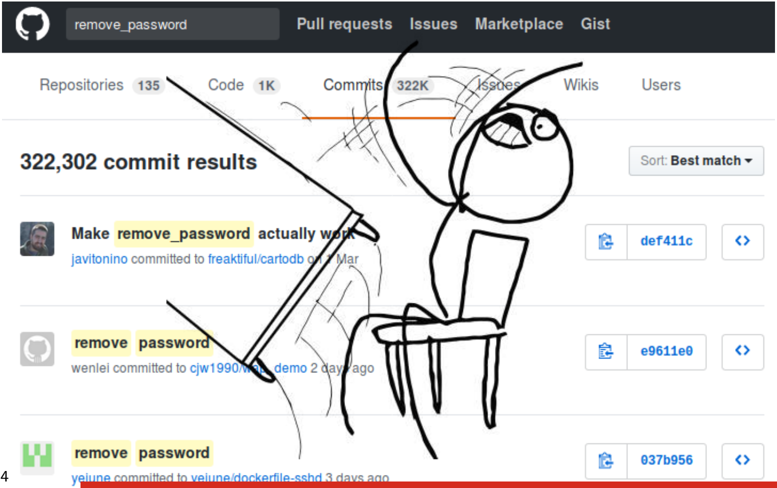
Make remove_password actually work
javitonino committed to freaktiful/cartodb on 1 Mar
def411c

remove password
wenlei committed to cjw1990/wap_demo 2 days ago
e9611e0

remove password
yejune committed to yejune/dockerfile-sshd 3 days ago
037b956

Sources of leaks: Others

- Mistakes from users
  - https://github.com/search?q=remove_password&type=Commits&ref=searchresults



322,302 commit results

Make remove_password actually work
javitonino committed to freaktiful/cartodb on 1 Mar — def411c

remove password
wenlei committed to cjw1990/wap_demo 2 days ago — e9611e0

remove password
yejune committed to yejune/dockerfile-sshd 3 days ago — 037b956

Are leaks frequent?

Yes!

And it’s important to detect them.

Paste monitoring at CIRCL: Statistics

- Monitored paste sites: 27
  - *pastebin.com*
  - *ideone.com*
  - *...*

Table: Statistics for 2016

Pastes 2016	Monthly average	Total
Fetched pastes	1 547 094	18 565 124
Security related (TR-46)	21	252
Incidents & investigations	54	649

AIL Framework

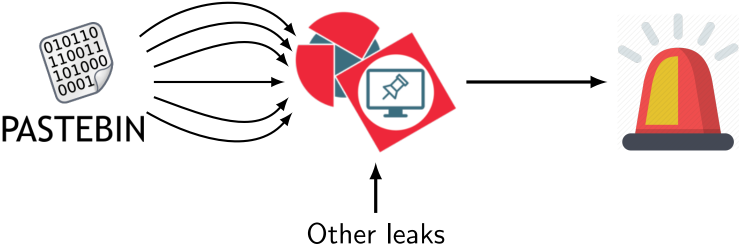
From a requirement to a solution: AIL Framework

History:

- AIL initially started as an internship project (2014) to evaluate the feasibility to automate the analysis of (un)structured information to find leaks.
- In 2018, AIL framework is an open source software in Python. The software is actively used (and maintained) by CIRCL.

AIL Framework: A framework for Analysis of Information Leaks

"AIL is a modular framework to analyse potential information leaks from unstructured data sources like pastes from Pastebin."



AIL Framework: Current capabilities

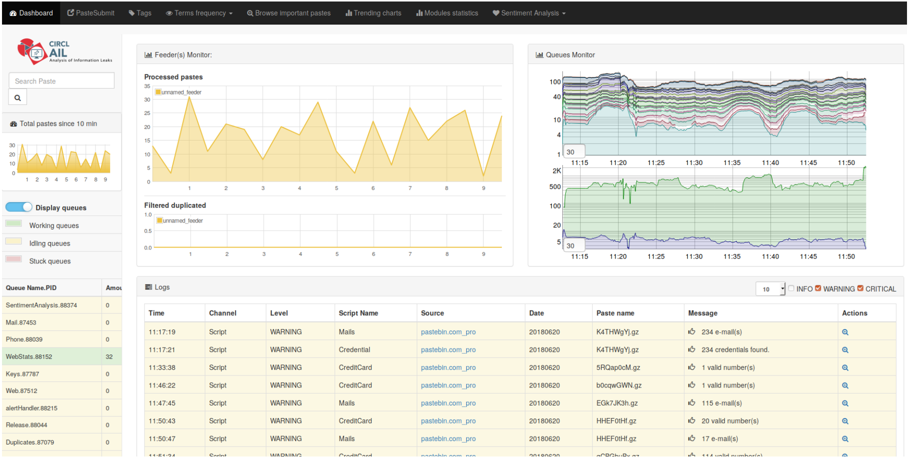
- Extending AIL to add a new **analysis module** can be done in 50 lines of Python
- The framework **supports multi-processors/cores by default**. Any analysis module can be started multiple times to support faster processing during peak times or bulk import
- **Multiple** concurrent **data input**

AIL Framework: Current features

- Extracting **credit cards numbers, credentials, phone numbers, ...**
- Extracting and validating potential **hostnames**
- Keeps track of **duplicates**
- Submission to threat sharing and incident response platform (**MISP** and **TheHive**)
- **Full-text indexer** to index unstructured information
- **Tagging** for classification and searches
- Terms, sets and regex **tracking and occurences**
- Archives, files and raw **submission** from the UI
- **Sentiment/Mood analyser** for incoming data
- And many more

Live demo!

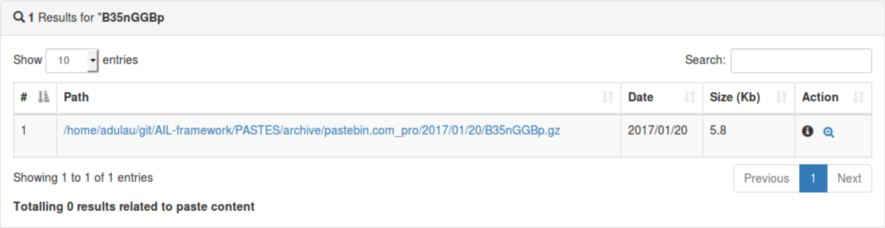
Example: Following a notification (0) - Dashboard



Queue Name.PID	Amou
SentimentAnalysis.88374	0
Mail.87453	0
Phone.88039	0
WebStats.88152	32
Keys.87787	0
Web.87512	0
alertHandler.88215	0
Release.88044	0
Duplicates.87079	0

Time	Channel	Level	Script Name	Source	Date	Paste name	Message	Actions
11:17:19	Script	WARNING	Mails	pastebin.com_pro	20180620	K4THWgYj.gz	234 e-mail(s)	
11:17:21	Script	WARNING	Credential	pastebin.com_pro	20180620	K4THWgYj.gz	234 credentials found.	
11:33:38	Script	WARNING	CreditCard	pastebin.com_pro	20180620	5RQap0cM.gz	1 valid number(s)	
11:46:22	Script	WARNING	CreditCard	pastebin.com_pro	20180620	b0cqwGWN.gz	1 valid number(s)	
11:47:45	Script	WARNING	Mails	pastebin.com_pro	20180620	EGk7JK3h.gz	115 e-mail(s)	
11:50:43	Script	WARNING	CreditCard	pastebin.com_pro	20180620	HHEF0tHf.gz	20 valid number(s)	
11:50:47	Script	WARNING	Mails	pastebin.com_pro	20180620	HHEF0tHf.gz	17 e-mail(s)	
11:51:34	Script	WARNING	CreditCard	pastebin.com_pro	20180620	gCPGhuPx.gz	114 valid number(s)	

Example: Following a notification (1) - Searching



1 Results for "B35nGGBp

Show 10 entries

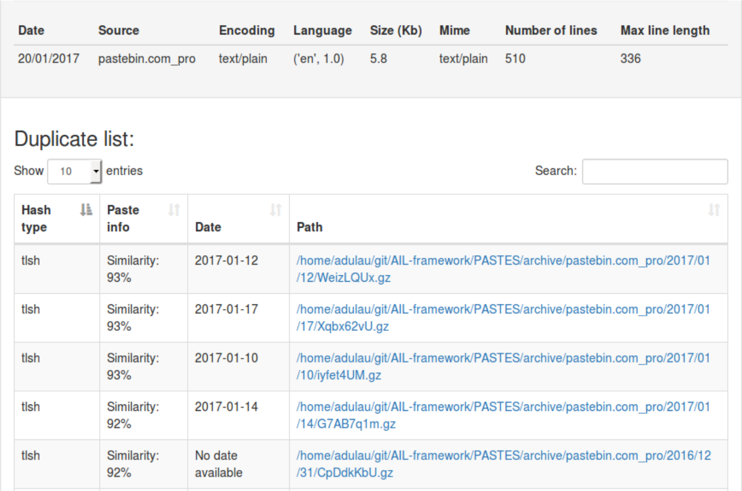
Search:

#	Path	Date	Size (Kb)	Action
1	/home/adulau/git/AIL-framework/PASTES/archive/pastebin.com_pro/2017/01/20/B35nGGBp.gz	2017/01/20	5.8	

Showing 1 to 1 of 1 entries

Totalling 0 results related to paste content

Example: Following a notification (2) - Metadata

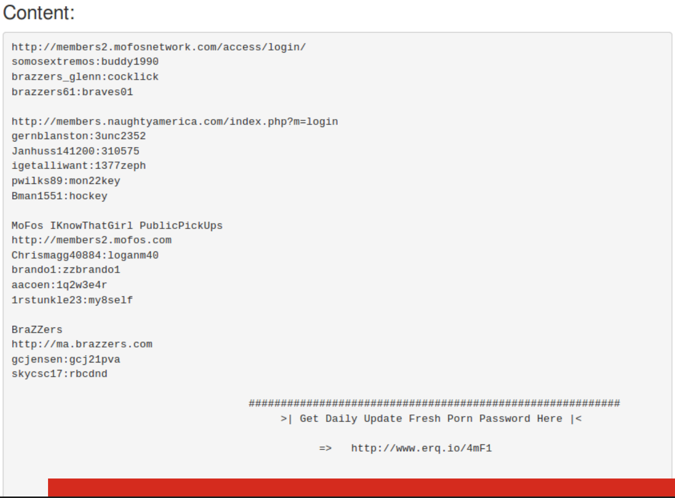


Date	Source	Encoding	Language	Size (Kb)	Mime	Number of lines	Max line length
20/01/2017	pastebin.com_pro	text/plain	('en', 1.0)	5.8	text/plain	510	336

Duplicate list:

Hash type	Paste info	Date	Path
tlsh	Similarity: 93%	2017-01-12	/home/adulau/git/AIL-framework/PASTES/archive/pastebin.com_pro/2017/01/12/WeizLQUx.gz
tlsh	Similarity: 93%	2017-01-17	/home/adulau/git/AIL-framework/PASTES/archive/pastebin.com_pro/2017/01/17/Xqbx62vU.gz
tlsh	Similarity: 93%	2017-01-10	/home/adulau/git/AIL-framework/PASTES/archive/pastebin.com_pro/2017/01/10/iyfet4UM.gz
tlsh	Similarity: 92%	2017-01-14	/home/adulau/git/AIL-framework/PASTES/archive/pastebin.com_pro/2017/01/14/G7AB7q1m.gz
tlsh	Similarity: 92%	No date available	/home/adulau/git/AIL-framework/PASTES/archive/pastebin.com_pro/2016/12/31/CpDdkKbU.gz

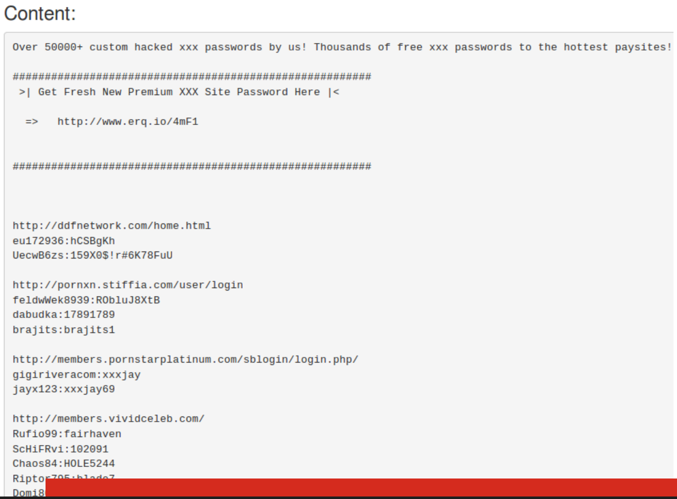
Example: Following a notification (3) - Browsing content



Content:

```
http://members2.mofosnetwork.com/access/login/
somosextremos:buddy1990
brazzers_glenn:cocklick
brazzers61:braves01

http://members.naughtyamerica.com/index.php?m=login
gernblanston:3unc2352
Janhuss141200:310575
igetalliwant:1377zeph
pwilks89:mon22key
Bman1551:hockey

MoFos IKnowThatGirl PublicPickUps
http://members2.mofos.com
Chrismagg40884:loganm40
brando1:zzbrando1
aacoen:1q2w3e4r
1rstunkle23:my8self

BraZZers
http://ma.brazzers.com
gcjensen:gcj21pva
skycsc17:rbcdnd

                                     ##########################################################
                                          >| Get Daily Update Fresh Porn Password Here |<

                                                =>   http://www.erq.io/4mF1
```

Example: Following a notification (3) - Browsing content



Content:

```
Over 50000+ custom hacked xxx passwords by us! Thousands of free xxx passwords to the hottest paysites!

########################################################
 >| Get Fresh New Premium XXX Site Password Here |<

  =>   http://www.erq.io/4mF1


########################################################




http://ddfnetwork.com/home.html
eu172936:hCSBgKh
UecwB6zs:159X0$!r#6K78FuU

http://pornxn.stiffia.com/user/login
feldwWek8939:RObluJ8XtB
dabudka:17891789
brajits:brajits1

http://members.pornstarplatinum.com/sblogin/login.php/
gigiriveracom:xxxjay
jayx123:xxxjay69

http://members.vividceleb.com/
Rufio99:fairhaven
ScHiFRvi:102091
Chaos84:HOLE5244
Riptor795:blade7
Domi8
```

Setting up the framework

Setting up AIL-Framework from source or virtual machine

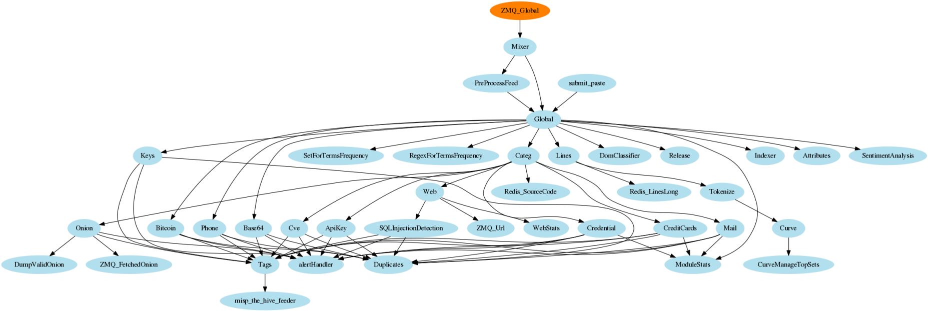
Setting up AIL-Framework from source

```
git clone https://github.com/CIRCL/AIL-framework.git
cd AIL-framework
./installing_deps.sh
cd var/www/
./update_thirdparty.sh
```

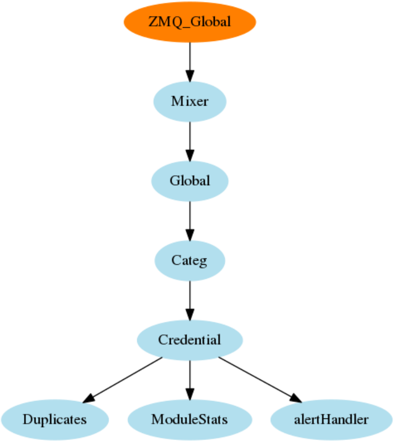
Using the virtual machine:

1. Download `https://www.circl.lu/assets/files/ail-training/AIL_v@4986352.ova`
2. Start virtualbox
3. File → import appliance → select AIL_v@4986352.ova
4. (for now) Prevent the automatic launch and `git pull` the changes

AIL global architecture: Data streaming between module



AIL global architecture: Data streaming between module (Credential example)



Starting the framework

Running your own instance from source

Make sure that ZMQ_Global→address =

`tcp://crf.circl.lu:5556,tcp://127.0.0.1:5556` in bin/package/config.cfg

Accessing the environment and starting AIL

```
# Activate the virtualenv
. ./AILENV/bin/activate

# Launch the system
cd bin/
./LAUNCH
    # check options 1->5

# Start web interface
cd var/www/
./Flask_server.py
    # -> Browse http://localhost:7000/
```

Running your own instance using the virtual machine

Login and passwords:

```
Web interface (default network settings):
    http://192.168.56.51:7000/
Shell/SSH:
    ail/Password1234

```

Feeding the framework

Feeding AIL

There are differents way to feed AIL with data:

1. Be a partner with CIRCL and ask to get access to our feed info@circl.lu
2. Setup *pystemon* and use the custom feeder
   - *pystemon* will collect pastes for you
3. Feed your own data using the `import_dir.py` script
4. Feed your own file/text using the UI (`/PasteSubmit/`)

Feeding AIL

There are differents way to feed AIL with data:

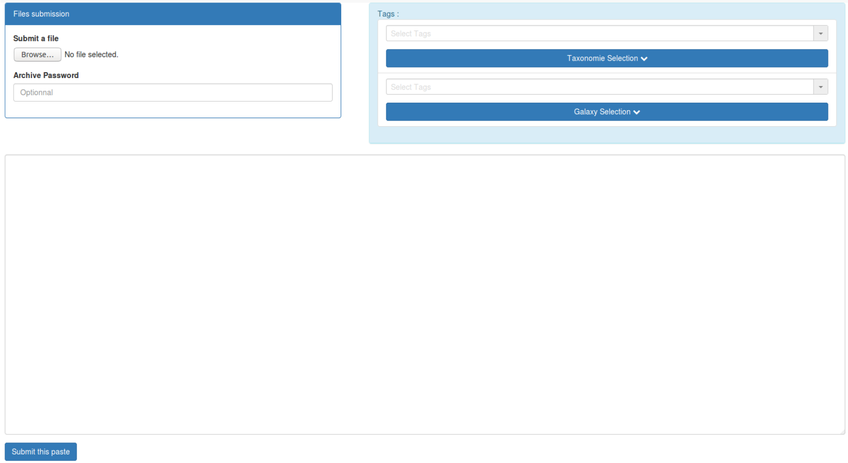
1. CIRCL partners and ask to access our feed info@circl.lu
   - ▷ You already have access
2. ~~Setup *pystemon* and use the custom feeder~~
   - ○ ~~*pystemon* will collect pastes for you~~
3. Feed your own file/text using the UI (`/PasteSubmit/`)
4. Feed your own data using the `import_dir.py` script

Plug-in AIL to the CIRCL feed

You can freely access the CIRCL feed during this workshop!

- In the file `bin/package/config.cfg`,
- Set `ZMQ_Global->address` to `tcp://crf.circl.lu:5556`

Via the UI (1)



Files submission

Submit a file

Browse... No file selected.

Archive Password

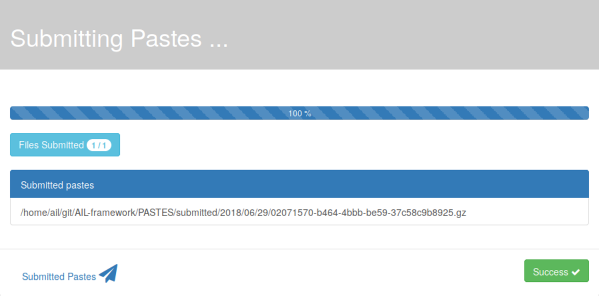
Optionnal

Tags :

Select Tags

Select Tags

Via the UI (2)



/home/ail/git/AIL-framework/PASTES/submitted/2018/06/29/02071570-b464-4bbb-be59-37c58c9b8925.gz

Feeding AIL with your own data - `import_dir.py` (1)

/!\ 2 requirements:

1. Data to be fed must have the path hierarchy as the following:
   1.1 `year/month/day/(textfile/gzfile)`
   1.2 This is due to the inner representation of paste in AIL
2. Each file to be fed must be of a raisonable size:
   2.1 $\sim$ `3 Mb` is already large
   2.2 This is because some modules are doing regex matching
   2.3 If you want to feed a large file, better split it in multiple ones

Feeding AIL with your own data - import_dir.py (2)

1. Check your local configuration bin/package/config.cfg
   - In the file bin/package/config.cfg,
   - Add 127.0.0.1:5556 in ZMQ_Global
   - (should already be set by default)

Feeding AIL with your own data - `import_dir.py` (2)

1. Check your local configuration `bin/package/config.cfg`
   - In the file `bin/package/config.cfg`,
   - Add `127.0.0.1:5556` in `ZMQ_Global`
   - (should already be set by default)
2. Launch `import_dir.py` with de directory you want to import
   - `import_dir.py -d dir_path`

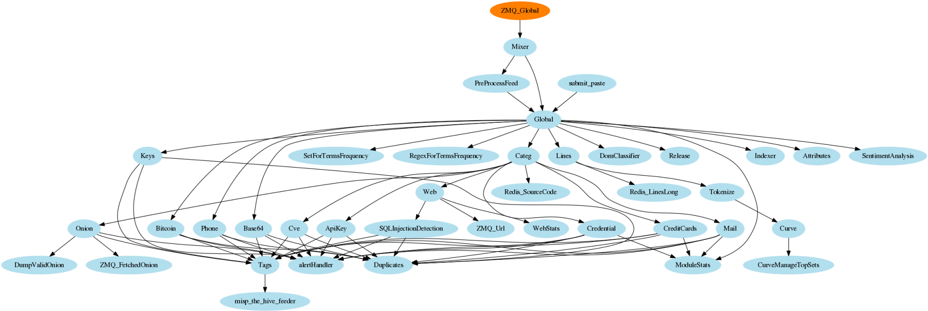
Feeding AIL with your own data - `import_dir.py` (2)

1. Check your local configuration `bin/package/config.cfg`
   - In the file `bin/package/config.cfg`,
   - Add `127.0.0.1:5556` in `ZMQ_Global`
   - (should already be set by default)
2. Launch `import_dir.py` with de directory you want to import
   - `import_dir.py -d dir_path`
3. Watch your data being feed to AIL

Creating new features

Developping new features: Plug-in a module in the system

Choose where to locate your module in the data flow:



Then, modify `bin/package/modules.cfg` accordingly

Writing your own modules - /bin/template.py

```python
import time
from pubsublogger import publisher
from Helper import Process
if __name__ == ’__main__’:
    # Port of the redis instance used by pubsublogger
    publisher.port = 6380
    # Script is the default channel used for the modules.
    publisher.channel = ’Script’
    # Section name in bin/packages/modules.cfg
    config_section = ’<section name>’
    # Setup the I/O queues
    p = Process(config_section)
    # Sent to the logging a description of the module
    publisher.info("<description of the module>")
    # Endless loop getting messages from the input queue
    while True:
        # Get one message from the input queue
        message = p.get_from_set()
        if message is None:
            publisher.debug("{} queue is empty, waiting".format(config_section))
            time.sleep(1)
            continue
        # Do something with the message from the queue
        something_has_been_done = do_something(message)

```

AIL - Add your own web interface

1. Launch `var/www/create_new_web_module.py`
2. Enter the module's name
3. A template and flask skeleton has been created for your new webpage in `var/www/modules/`
4. You can start **coding** server-side in:

   var/www/modules/*your_module_name*/Flask_*your_module_name.py*

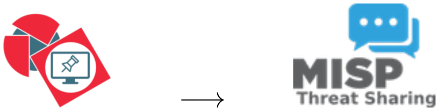
5. You can start **coding** client-side in:

   var/www/modules/*your_module_name*/templates/*your_module_name.html*

   var/www/modules/*your_module_name*/templates/header_*your_module_name.html*

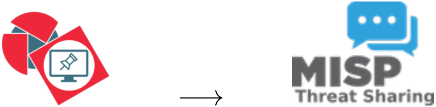
Case study: Push alert to MISP

Push alert to MISP



Goal: push tags to MISP.

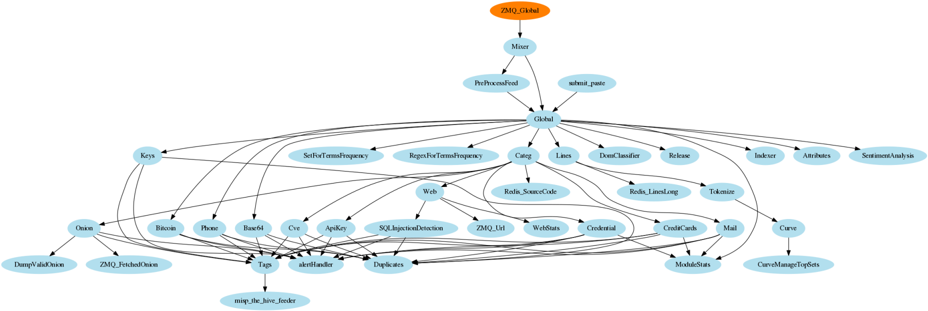
Push alert to MISP



1. Use infoleak taxonomie
2. Add your own tags
3. Create a event on a paste

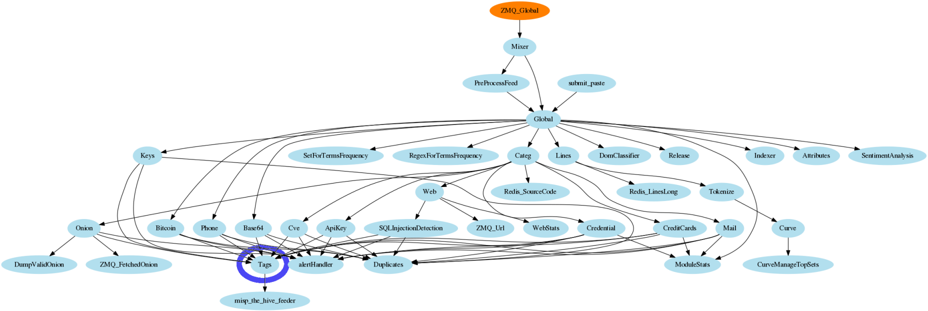
Case study: Finding the best place in the system

Best place to put it?



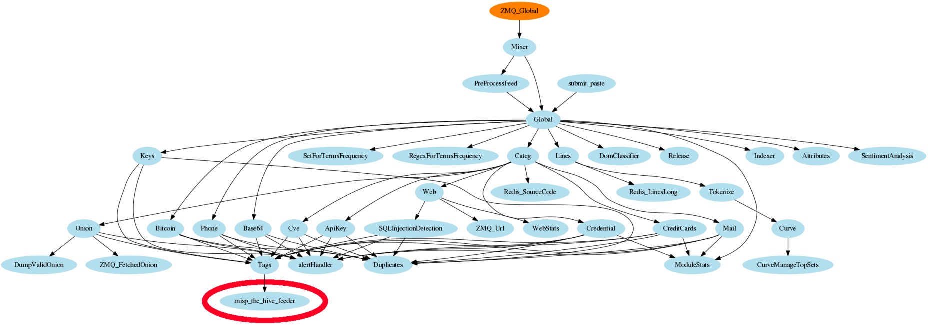
Case study: Finding the best place in the system

Best place to put it?



Case study: Finding the best place in the system

Best place to put it?

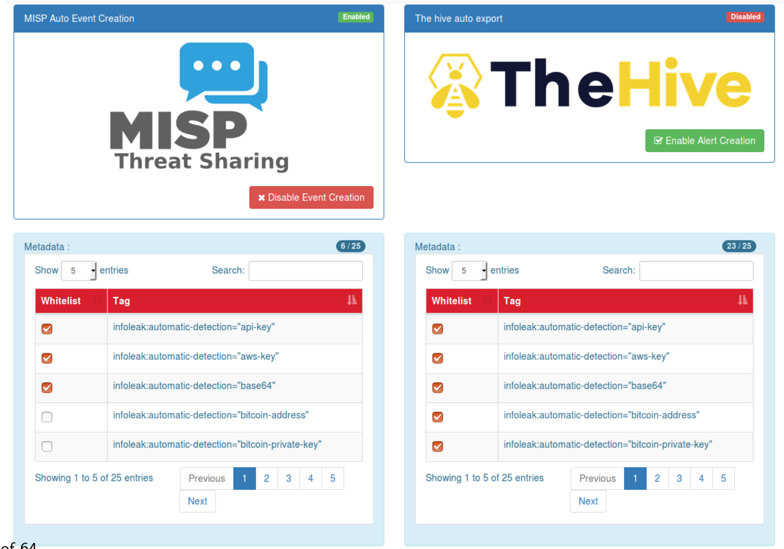


Case study: Updating `Flask server.py`

`Flask server.py`

```
[...]
# ========== INITIAL tags auto export ============
r_serv_db = redis.StrictRedis(
    host=cfg.get("ARDB_DB", "host"),
    port=cfg.getint("ARDB_DB", "port"),
    db=cfg.getint("ARDB_DB", "db"),
    decode_responses=True)
infoleak_tags = taxonomies.get('infoleak').machinetags()
infoleak_automatic_tags = []
for tag in taxonomies.get('infoleak').machinetags():
    if tag.split('=')[0][:] == 'infoleak:automatic-detection':
        r_serv_db.sadd('list_export_tags', tag)

r_serv_db.sadd('list_export_tags', 'infoleak:submission="manual"')
r_serv_db.sadd('list_export_tags', '<your_tag>')

```

Auto Push Tags



MISP Auto Event Creation — Enabled

✖ Disable Event Creation

The hive auto export — Disabled

☑ Enable Alert Creation

Metadata : 6 / 25

Show 5 entries Search:

Whitelist	Tag
☑	infoleak:automatic-detection="api-key"
☑	infoleak:automatic-detection="aws-key"
☑	infoleak:automatic-detection="base64"
□	infoleak:automatic-detection="bitcoin-address"
□	infoleak:automatic-detection="bitcoin-private-key"

Showing 1 to 5 of 25 entries

Previous 1 2 3 4 5 Next

Metadata : 23 / 25

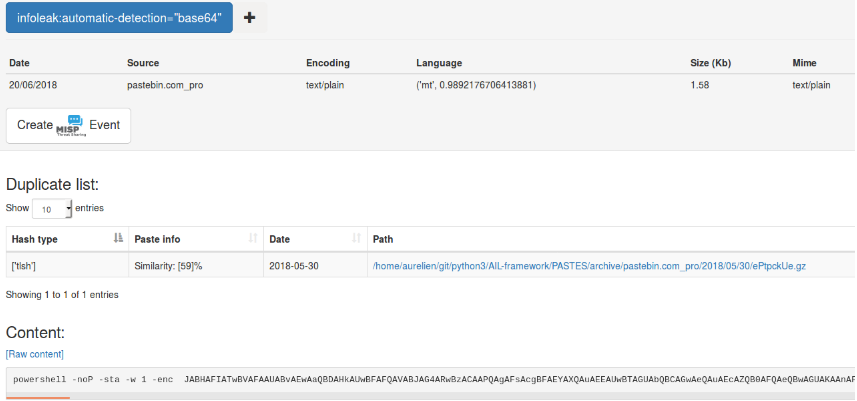
Show 5 entries Search:

Whitelist	Tag
☑	infoleak:automatic-detection="api-key"
☑	infoleak:automatic-detection="aws-key"
☑	infoleak:automatic-detection="base64"
☑	infoleak:automatic-detection="bitcoin-address"
☑	infoleak:automatic-detection="bitcoin-private-key"

Showing 1 to 5 of 25 entries

Previous 1 2 3 4 5 Next

Create a event



infoleak:automatic-detection="base64"
+

Date	Source	Encoding	Language	Size (Kb)	Mime
20/06/2018	pastebin.com_pro	text/plain	('mt', 0.9892176706413881)	1.58	text/plain

Create MISP Threat Sharing Event

Duplicate list:

Show 10 entries

Hash type	Paste info	Date	Path
['tlsh']	Similarity: [59]%	2018-05-30	/home/aurelien/git/python3/AIL-framework/PASTES/archive/pastebin.com_pro/2018/05/30/ePtpckUe.gz

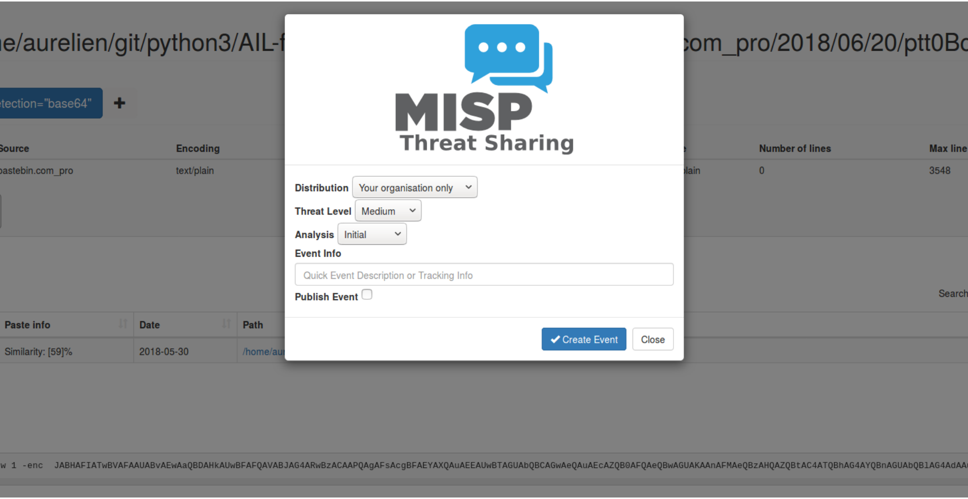
Showing 1 to 1 of 1 entries

Content:

[Raw content]

```
powershell -noP -sta -w 1 -enc  JABHAFIATwBVAFAAUABvAEwAaQBDAHkAUwBFAFQAVABJAG4ARwBzACAAPQAgAFsAcgBFAEYAXQAuAEEAUwBTAGUAbQBCAGwAeQAuAEcAZQB0AFQAeQBwAGUAKAAnAF
```

Create a event



e/aurelien/git/python3/AIL-f

com_pro/2018/06/20/ptt0Bc

tection="base64" +

Source	Encoding	e	Number of lines	Max line
pastebin.com_pro	text/plain	lain	0	3548

Distribution: Your organisation only

Threat Level: Medium

Analysis: Initial

Event Info

Quick Event Description or Tracking Info

Publish Event □

Paste info	Date	Path
Similarity: [59]%	2018-05-30	/home/au

```
w 1 -enc  JABHAFIATwBVAFAAUABvAEwAaQBDAHkAUwBFAFQAVABJAG4ARwBzACAAPQAgAFsAcgBFAEYAXQAuAEEAUwBTAGUAbQBCAGwAeQAuAEcAZQB0AFQAeQBwAGUAKAAnAFMAeQBzAHQAZQBtAC4ATQBhAG4AYQBnAGUAbQBlAG4AdAA
```

Practical part

Practical part: Pick your choice

1. Improve module `keys.py` to support other type of keys (ssh, ...)
   - `https://github.com/veorq/blueflower/blob/master/blueflower/constants.py`
2. Graph database on `Credential.py`
   - Top used passwords, most compromised user, ...
3. Webpage scrapper
   - Download html from URL found in pastes
   - Re-inject html as paste in AIL
4. Improvement of `Phone.py`
   - Way to much false positive as of now. Exploring new ways to validate phone numbers could be interesting
5. Your custom feature

Contribution rules

How to contribute

How to contribute

- Feel free to fork the code, play with it, make some patches or add additional analysis modules.

How to contribute

- Feel free to fork the code, play with it, make some patches or add additional analysis modules.
- Feel free to make a pull request for your contribution

How to contribute

- Feel free to fork the code, play with it, make some patches or add additional analysis modules.
- Feel free to make a pull request for your contribution
- That's it!

く(^.^)ノ

Final words

- Building AIL helped us to find additional leaks which cannot be found using manual analysis and **improve the time to detect duplicate/recycled leaks**.

  → Therefore quicker response time to assist and/or inform proactively affected constituents.

Annexes

Managing the framework

Managing AIL: Old fashion way

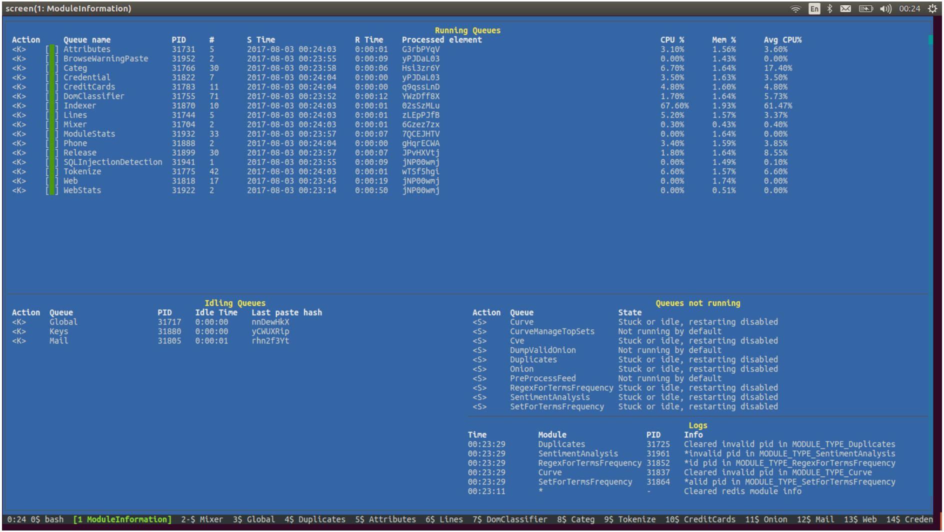
Access the script screen

```
screen -r Script
```

Table: GNU screen shortcuts

Shortcut	Action
C-a d	detach screen
C-a c	Create new window
C-a n	next window screen
C-a p	previous window screen

Managing your modules: Using the helper



```
screen(1: ModuleInformation)                                                                 00:24

                                                              Running Queues
Action    Queue name              PID    #      S Time                R Time    Processed element                       CPU %     Mem %     Avg CPU%
<K>     [█] Attributes            31731  5      2017-08-03 00:24:03   0:00:01   G3rbPYqV                                3.10%     1.56%     3.60%
<K>     [█] BrowseWarningPaste    31952  2      2017-08-03 00:23:55   0:00:09   yPJDaL03                                0.00%     1.43%     0.00%
<K>     [█] Categ                 31766  30     2017-08-03 00:23:58   0:00:06   Hsi3zr6Y                                6.70%     1.64%     17.40%
<K>     [█] Credential            31822  7      2017-08-03 00:24:04   0:00:00   yPJDaL03                                3.50%     1.63%     3.50%
<K>     [█] CreditCards           31783  11     2017-08-03 00:24:04   0:00:00   q9qssLnD                                4.80%     1.60%     4.80%
<K>     [█] DomClassifier         31755  71     2017-08-03 00:23:52   0:00:12   YWzDff8X                                1.70%     1.64%     5.73%
<K>     [█] Indexer               31870  10     2017-08-03 00:24:03   0:00:01   02sSzMLu                                67.60%    1.93%     61.47%
<K>     [█] Lines                 31744  5      2017-08-03 00:24:03   0:00:01   zLEpPJfB                                5.20%     1.57%     3.37%
<K>     [█] Mixer                 31704  2      2017-08-03 00:24:03   0:00:01   6Gzez7zx                                0.30%     0.43%     0.40%
<K>     [█] ModuleStats           31932  33     2017-08-03 00:23:57   0:00:07   7QCEJHTV                                0.00%     1.64%     0.00%
<K>     [█] Phone                 31888  2      2017-08-03 00:24:04   0:00:00   gHqrECWA                                3.40%     1.59%     3.85%
<K>     [█] Release               31899  30     2017-08-03 00:23:57   0:00:07   JPvHXVtj                                1.80%     1.64%     8.55%
<K>     [█] SQLInjectionDetection 31941  1      2017-08-03 00:23:55   0:00:09   jNP00wmj                                0.00%     1.49%     0.10%
<K>     [█] Tokenize              31775  42     2017-08-03 00:24:03   0:00:01   wTSf5hgi                                6.60%     1.57%     6.60%
<K>     [█] Web                   31818  17     2017-08-03 00:23:45   0:00:19   jNP00wmj                                0.00%     1.74%     0.00%
<K>     [█] WebStats              31922  2      2017-08-03 00:23:14   0:00:50   jNP00wmj                                0.00%     0.51%     0.00%
```

```
                   Idling Queues
Action   Queue               PID     Idle Time   Last paste hash
<K>      Global              31717   0:00:00     nnDewHkX
<K>      Keys                31880   0:00:00     yCWUXRip
<K>      Mail                31805   0:00:01     rhn2f3Yt
```

```
                                       Queues not running
Action   Queue                  State
<S>      Curve                  Stuck or idle, restarting disabled
<S>      CurveManageTopSets     Not running by default
<S>      Cve                    Stuck or idle, restarting disabled
<S>      DumpValidOnion         Not running by default
<S>      Duplicates             Stuck or idle, restarting disabled
<S>      Onion                  Stuck or idle, restarting disabled
<S>      PreProcessFeed         Not running by default
<S>      RegexForTermsFrequency Stuck or idle, restarting disabled
<S>      SentimentAnalysis      Stuck or idle, restarting disabled
<S>      SetForTermsFrequency   Stuck or idle, restarting disabled

                                      Logs
Time          Module                 PID      Info
00:23:29      Duplicates             31725    Cleared invalid pid in MODULE_TYPE_Duplicates
00:23:29      SentimentAnalysis      31961    *invalid pid in MODULE_TYPE_SentimentAnalysis
00:23:29      RegexForTermsFrequency 31852    *id pid in MODULE_TYPE_RegexForTermsFrequency
00:23:29      Curve                  31837    Cleared invalid pid in MODULE_TYPE_Curve
00:23:29      SetForTermsFrequency   31864    *alid pid in MODULE_TYPE_SetForTermsFrequency
00:23:11      *                      -        Cleared redis module info
```

```
0:24 0$ bash  [1 ModuleInformation]  2-$ Mixer  3$ Global  4$ Duplicates  5$ Attributes  6$ Lines  7$ DomClassifier  8$ Categ  9$ Tokenize  10$ CreditCards  11$ Onion  12$ Mail  13$ Web  14$ Creden
```

AIL ecosystem: Technologies used

Programing language: python3

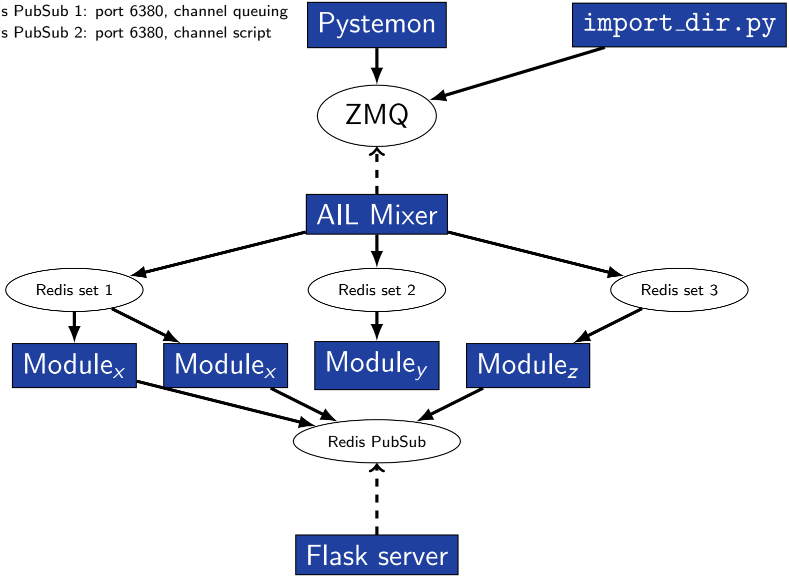
Databases: Redis and ARDB

Server: Flask

Data message passing: ZMQ and Redis Publisher/Subscriber

AIL global architecture

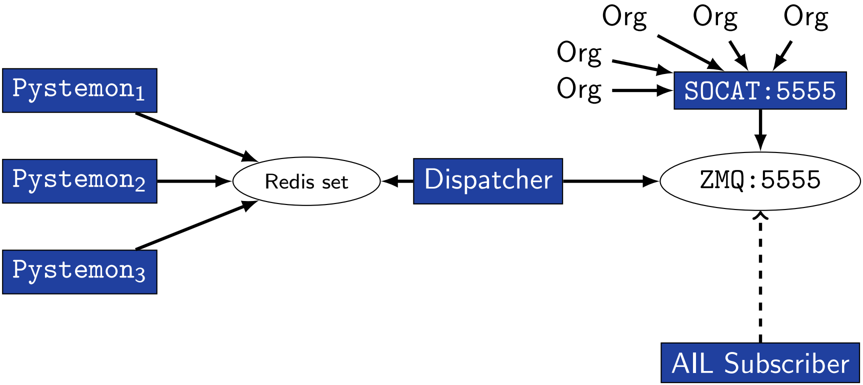
Redis PubSub 1: port 6380, channel queuing
Redis PubSub 2: port 6380, channel script



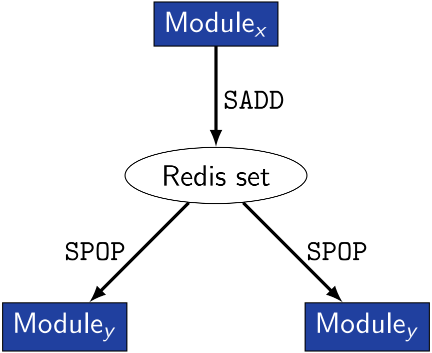
Data feeder: Gathering pastes with pystemon

Pystemon global architecture

Redis PubSub 1: port 6380, channel queuing
Redis PubSub 2: port 6380, channel script



Message consuming



→ No message lost nor double processing

→ Multiprocessing!