

# AIL Framework for Analysis of Information Leaks

Workshop - A generic analysis open source software



**CIRCL**  
Computer Incident  
Response Center  
Luxembourg

Sami Mokaddem

[info@circl.lu](mailto:info@circl.lu)

August 3, 2017

## Objectives of the workshop

---

- Learn how to install and start AIL
- Learn how to manage current modules/features
- Learn how to create new modules/features
- Discover new open source software \o/

# Planning

---

- Introduction to AIL-Framework
  - Why? and what?
  - Capabilities and screenshots demo
- How to use AIL-Framework
  - Installation
  - Running your own instance
  - Using the web interface
  - Managing modules
- How to feed data to AIL-Framework
  - Feeding your own data
- Writing your own module
- Try it out!

# 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."*



## A source of leaks: Paste monitoring (1)

---

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

## A source of leaks: Paste monitoring (1)

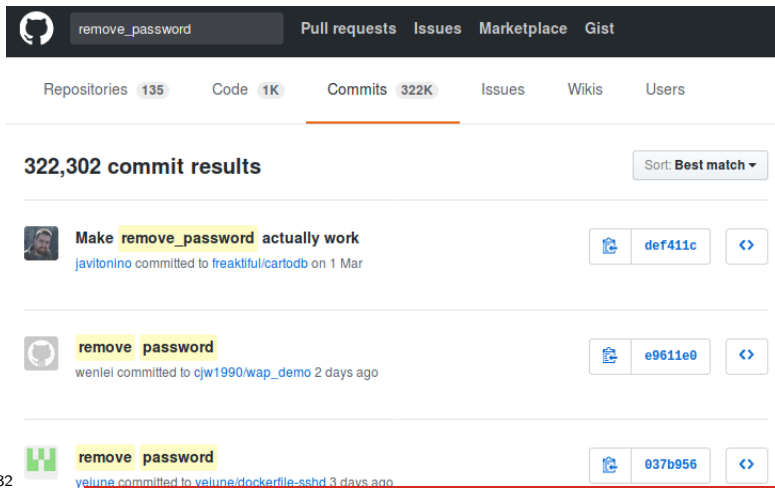
---

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

# A source of leaks: Paste monitoring (2)

- Mistakes from users

- [https://github.com/search?q=remove\\_password&type=Commits&ref=searchresults](https://github.com/search?q=remove_password&type=Commits&ref=searchresults)



The screenshot shows the GitHub search interface for the query "remove\_password". The search results are filtered to "Commits" and show 322,302 results. The top three results are:

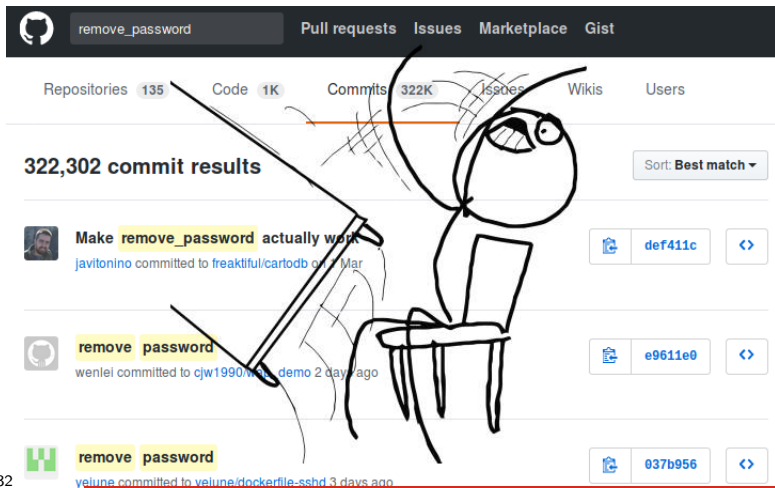
- Make remove\_password actually work**  
javitonino committed to freakiful/cartodb on 1 Mar
- remove password**  
wenlei committed to cjlw1990/wap\_demo 2 days ago
- remove password**  
yelune committed to yelune/dockerfile-sshd 3 days ago

Each result includes a commit hash (def411c, e9611e0, 037b956) and a link to view the commit details.

# A source of leaks: Paste monitoring (2)

- Mistakes from users

- [https://github.com/search?q=remove\\_password&type=Commits&ref=searchresults](https://github.com/search?q=remove_password&type=Commits&ref=searchresults)



The screenshot shows the GitHub search interface for the query "remove\_password". The search results are filtered to "Commits" and show 322,302 results. A hand-drawn stick figure is overlaid on the page, sitting on a chair and pointing a long, thin object (resembling a pen or pencil) at the search results. The figure has a large, expressive face with wide eyes and a slightly open mouth, suggesting surprise or concern. The search results list several commits, including one by javitonino, wenlei, and yelune, all containing the text "remove\_password".


remove\_password


Pull requests Issues Marketplace Gist


Repositories 135 Code 1K Commits 322K Issues Wikis Users

322,302 commit results

Sort: Best match ▾

 Make **remove\_password** actually work  
javitonino committed to freaktiful/cartodb on 7 Mar

 **remove password**  
wenlei committed to cju1990/wenlei-demo 2 day ago

 **remove password**  
yelune committed to yelune/dockerfile-ssh 3 days ago

def411c

e9611e0

037b956



# Examples of pastes

---

text 4.41 KB

```
1. - - - - - Tool by Y3t1y3t ( u  
2.  
3.
```

text 4.57 KB

```
1. #include "wejwyj.h"  
2.  
3. int zapisz (FILE *plik_  
4.     int i, j;  
5. if (obr->KOLOR==0) {  
6.  
7.     fprintf (plik_wy, "P2  
8.     fprintf (plik_wy, "%d  
9.     fprintf (plik_wy, "%d  
10. for (i=0; i<obr->wymy  
11.     for (j=0; j<obr->wymx; j++  
12.         fprintf (plik_wy, "%d ",  
13.     }
```

text 2.02 KB

```
1. KillerGram - Yuffie - Smoke The Big Dick [smkwhr] (Upload  
2.  
3.
```

text 2.66 KB

```
1. <item name="%the_component_to_be_disabled%" xsi:type="array">  
2.     <item name="config" xsi:type="array">  
3.         <item name="componentDisabled" xsi:type="boolean">true</item>  
4.     </item>  
5. </item>  
6.  
7. <?xml version="1.0"?>  
8.  
9. <page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespace  
/etc/page_configuration.xsd">  
10.     <body>  
11.         <referenceBlock name="checkout.root">  
12.             <arguments>  
13.                 <argument name="jsLayout" xsi:type="array">
```

# Examples of pastes

~\ (ツ) \_/

The image shows a screenshot of a code editor with four overlapping text files:

- Top-left file (4.41 KB):** Contains a single line of text: `----- Tool by Y3t1y3t ( u`
- Top-right file (2.02 KB):** Contains a URL: `KillerGram - Yuffie - Smoke The Big Dick [smkwhr] (Upload`, which is circled in red.
- Bottom-left file (4.57 KB):** Contains C code for a program that reads a file and prints its contents. The code includes `#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 "`, and `}`.
- Bottom-right file (2.66 KB):** Contains XML code for a page configuration. The code includes `<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>`, and `<argument name="jsLayout" xsi:type="array">`.

## Paste monitoring at CIRCL: Statistics

---

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

Table: Statistics for 2016

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

## 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 2017, AIL framework is an open source software in Python. The software is actively used (and maintained) by CIRCL.

## 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 2017, AIL framework is an open source software in Python. The software is actively used (and maintained) by CIRCL.
- 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.

## Current capabilities

---

- **Multiple** concurrent **data input**
- Extracting **credit cards numbers, credentials, phone numbers, ...**
- Extracting and validating potential **hostnames**
- Keep track of **duplicates**
- **Full-text indexer** to index unstructured information
- Terms, sets and regex **tracking and occurrences**
- **Sentiment/Mood analyser** for incoming data
- Modules manager
- And many more



# AIL: Following a notification (1) - Searching

---

Q 1 Results for "B35nGGBp"

Show  entries Search:

#	Path	Date	Size (Kb)	Action
1	<a href="/home/adulau/git/AIL-framework/PASTES/archive/pastebin.com_pro/2017/01/20/B35nGGBp.gz">/home/adulau/git/AIL-framework/PASTES/archive/pastebin.com_pro/2017/01/20/B35nGGBp.gz</a>	2017/01/20	5.8	<a href="#">i</a> <a href="#">Q</a>

Showing 1 to 1 of 1 entries Previous **1** Next

**Totalling 0 results related to paste content**



## AIL: 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:

Show  entries Search:

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

# ALL: Following a notification (3) - Browsing content

---

## Content:

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

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

```
MoFos IKnowThatGir1 PublicPickUps  
http://members2.mofos.com  
ChriSmagg40884:loganm40  
brando1:zzbrando1  
aacoen:1q2w3e4r  
1rstunkle23:my8self
```

```
BraZZers  
http://ma.brazzers.com  
gcjensen:gcj21pva  
skycsc17:rbcndnd
```

```
#####
```

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

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

# AIL: 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:R0bluJ8XB  
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  
Domi80:harkonnen  
GaggedUK:a1k0chan  
  
http: [REDACTED]
```

**UNCLE SAM  
WANTS YOU**



[imgflip.com](http://imgflip.com)

# Installation

---

## Setting up AIL-Framework

```
1 git clone https://github.com/CIRCL/AIL-framework.git
2 cd AIL-framework
3 ./installing_deps.sh
4 cd var/www/
5 ./update_thirdparty.sh
```

# Running your own instance

---

## Accessing the environment and Starting AIL

```
1 cd ~/AIL-Framework/  
2 . ./AILENV/bin/activate  
3 cd bin/  
4 ./LAUNCH
```

# Running your own instance

---

## Accessing the environment and Starting AIL

```
1 cd ~/AIL-Framework/  
2 . ./AILENV/bin/activate  
3 cd bin/  
4 ./LAUNCH
```

## Starting the web interface

```
1 cd $AILENV  
2 cd var/www/  
3 ./Flask_server.py  
4 # -> Browse http://localhost:7000/
```

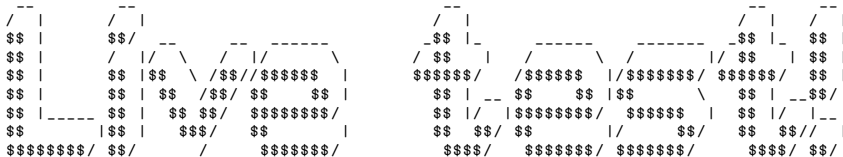
## Using the web interface

---



# Using the web interface

---



## Managing your modules: Old fashion way

---

### Access the script screen

```
1 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)

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 G3rBPVqV 3.10% 1.56% 3.60%
<K> BrowseWarningPaste 31952 2 2017-08-03 00:23:55 0:00:09 yP3DaL03 0.00% 1.43% 0.00%
<K> Categ 31766 30 2017-08-03 00:23:58 0:00:06 Hs13zr6Y 6.70% 1.64% 17.40%
<K> Credential 31822 7 2017-08-03 00:24:04 0:00:00 yP3DaL03 3.50% 1.63% 3.50%
<K> CreditCards 31783 11 2017-08-03 00:24:04 0:00:00 q9qssLnd 4.80% 1.66% 4.80%
<K> DomClassifier 31755 71 2017-08-03 00:23:52 0:00:12 Wz0FFBX 1.70% 1.64% 5.73%
<K> Indexer 31870 10 2017-08-03 00:24:03 0:00:01 0255zMLU 67.60% 1.93% 61.47%
<K> Lines 31744 5 2017-08-03 00:24:03 0:00:01 zLEpJfB 5.20% 1.57% 3.37%
<K> Mlxer 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 ghqFECHA 3.40% 1.59% 3.85%
<K> Release 31899 30 2017-08-03 00:23:57 0:00:07 3PwHXVtJ 1.80% 1.64% 0.55%
<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 WTSF5hgL 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%

Idle Queues
Action Queue Idle Time Last paste hash
<K> Global 31717 0:00:00 n0DwHkX
<K> Keys 31880 0:00:00 yCWJXRlp
<K> Mail 31805 0:00:01 rhn2F3Yt

Queues not running
Action Queue State
<S> Curve Stuck or idle, restarting disabled
<S> CurveManagementSets Not running by default
<S> Cve Stuck or idle, restarting disabled
<S> DumpValidOntion Not running by default
<S> Duplicates Stuck or idle, restarting disabled
<S> Ontion 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
TTime 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 *id pid in MODULE_TYPE_SetForTermsFrequency
00:23:11 * - Cleared redis module info

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

## Feeding AIL

---

There are different ways to feed AIL with data:

1. Be a collaborator of CIRCL and ask to access our feed
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

## Feeding ALL with your own data - `import_dir.py`

---

1. Change your local configuration `bin/package/config.cfg`
  - change address of `ZMQ_Global` to `127.0.0.1:5556`

## Feeding ALL with your own data - `import_dir.py`

---

1. Change your local configuration `bin/package/config.cfg`
  - change address of `ZMQ_Global` to `127.0.0.1:5556`
2. launch `import_dir.py` with de directory you want to import
  - `import_dir.py -d dir_path`

## Feeding ALL with your own data - `import_dir.py`

---

1. Change your local configuration `bin/package/config.cfg`
  - change address of `ZMQ_Global` to `127.0.0.1:5556`
2. launch `import_dir.py` with the directory you want to import
  - `import_dir.py -d dir_path`
3. Watch your data being feed to ALL



## Feeding AIL

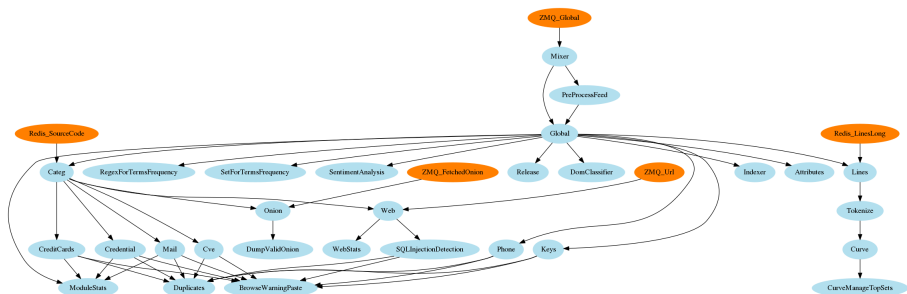
---

- Access to the workshop feed @ xxx.xxx.xxx.xxx:5556
  - change address of ZMQ\_Global to xxx.xxx.xxx.xxx:5556 of the file bin/package/config.cfg



# AIL - Add your own module

Choose where to locate your module in the data flow:



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

## Writing your own modules - /bin/template.py

---

```
1 import time
2 from pubsublogger import publisher
3 from Helper import Process
4 if __name__ == '__main__':
5     # Port of the redis instance used by pubsublogger
6     publisher.port = 6380
7     # Script is the default channel used for the modules.
8     publisher.channel = 'Script'
9     # Section name in bin/packages/modules.cfg
10    config_section = '<section name>'
11    # Setup the I/O queues
12    p = Process(config_section)
13    # Sent to the logging a description of the module
14    publisher.info("<description of the module>")
15    # Endless loop getting messages from the input queue
16    while True:
17        # Get one message from the input queue
18        message = p.get_from_set()
19        if message is None:
20            publisher.debug("{} queue is empty, waiting".format(config_section))
21            time.sleep(1)
22            continue
23        # Do something with the message from the queue
24        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!**

## 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.
- To contribute your module, feel free to pull your contribution.

## How to contribute

---

- Feel free to fork the code, play with it, make some patches or add additional analysis modules.
- To contribute your module, feel free to pull your contribution.
- That's it!

< ( ^ . ^ ) >

## Conclusion

---

- 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.