

# AIL Framework for Analysis of Information Leaks

Practical and Efficient Data-Mining of Suspicious Websites, Forums, Chats and  
Tor Hidden-Services

 <https://ail-project.org/>

---

Aurelien Thirion

Alexandre Dulaunoy - [info@circl.lu](mailto:info@circl.lu)

July 16, 2025

CIRCL <https://www.circl.lu>

## Links

- AIL project website <https://ail-project.org>
- AIL project open source framework <https://github.com/ail-project>
- AIL framework <https://github.com/ail-project/ail-framework>
- Training materials <https://github.com/ail-project/ail-training>
- Mastodon [https://infosec.exchange/@ail\\_project](https://infosec.exchange/@ail_project)
- Online chat <https://gitter.im/ail-project/community>



## Legal and Ethics

---

- Many modules in AIL can process personal data and even special categories of data as defined in GDPR (Art. 9).
- The data controller is often the operator of the AIL framework (limited to the organisation) and has to define **legal grounds for processing personal data**.
- To help users of AIL framework, a document is available which describe points of AIL in regards to the regulation<sup>1</sup>.

---

<sup>1</sup><https://www.circl.lu/assets/files/information-leaks-analysis-and-gdpr.pdf>

## Potential legal grounds

- **Consent of the data subject** is in many cases not feasible in practice and often impossible or illogical to obtain (Art. 6(1)(a)).
- Legal obligation (Art. 6(1)(c)) - This legal ground applies mostly to CSIRTs, in accordance with the powers and responsibilities set out in CSIRTs mandate and with their constituency, as they may have the legal obligation to collect, analyse and share information leaks without having a prior consent of the data subject.
- Art. 6(1)(f) - Legitimate interest - Recital 49 explicitly refers to CSIRTs' right to process personal data provided that they have a legitimate interest but not colliding with fundamental rights and freedoms of data subject.

# Ethics in Information Security and Cybersecurity

- The materials and tools presented can open a significant numbers of questions regarding ethics;
- Our researches and tools are there for education, supporting the public good and improve incident response;
- We ask all users and participants to **follow ethical principles and act professionally**<sup>2</sup>.

---

<sup>2</sup><https://www.acm.org/code-of-ethics> <https://www.first.org/global/sigs/ethics/ethics-first>

## Introduction

---

## Concepts - Deep Web

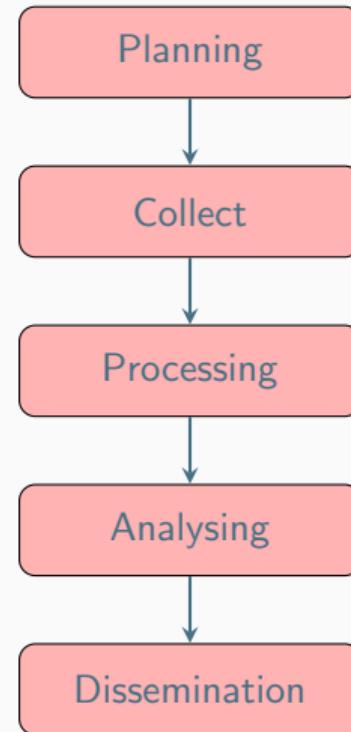
- **Deep Web** is the part of World Wide Web not indexed or directly accessible by standard web search-engines;
- This can be content hidden from **crawlers** by requiring a specific access and this can includes private social media, password-protected forums or content protected by different measures such as paywalls or specific security interface to access the information;
- A large portion of content accessible via Internet is part of the deep web<sup>3</sup>.

---

<sup>3</sup>also called invisible web, hidden web or non-indexed web

- **Darknet** is an overlay network running on top of Internet requiring specific software to access the network and its services;
- Tor, I2P and Freenet are the most commonly used ones. Many are used for hidden services access and some for proxy access to the Internet;
- There are **legitimate use-cases** for such network but also many **illegal or criminal usage**.

# Lifecycle of collection and analysis



- Building a search engine on the web is a challenging task because:
  - it has to crawl webpages,
  - it has to make sense of **unstructured data**,
  - it has to **index** these data,
  - it has to provide a way to retrieve data and structure data (e.g. correlation).
- Doing so on Tor is even more challenging because:
  - services don't always want to be found,
  - parts of the dataset have to be discarded.
- in each case, it requires a lot of bandwidth, storage and computing power.

## Collecting, processing and analysing content - structured data

- Some data are structured and are easy to process:
  - metadata!
  - API responses.
- Some even provide cryptographic evidences:
  - authentication mechanisms between peers,
  - OpenGPG can leak a lot of metadata
    - key ids,
    - subject of email in thunderbird,
  - Bitcoin's Blockchain is public,
  - pivoting on these data with external sources yields interesting results.

## AIL design Objectives

---

## Objectives of the session

- Show how to use and extend an open source tool to monitor web pages, pastes, chats, forums and hidden services
- Explain challenges and the design of the AIL open source framework
- Review different **collection mechanisms** and **sources**
- Learn how to create new modules
- Learn how to use, install and start AIL
- **Supporting investigation using the AIL framework** and including it in cyber threat intelligence life cycle

## 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 2019, AIL framework is an **open source software** in Python. The software is actively used (and maintained) by CIRCL and many organisations.
- In 2020, AIL framework is now a complete project called **ail project**<sup>4</sup>.
- In 2023, AIL framework version 5.0 released with new datastorage back-end.

---

<sup>4</sup><https://github.com/ail-project/>

## Capabilities Overview

---

## Common usage

---

- **Check** if mail/password/other sensitive information (terms tracked) leaked
- **Detect** reconnaissance of your infrastructure
- **Search** for leaks inside an archive
- **Monitor** and crawl chats/websites

## Support CERT and Law Enforcement activities

---

- Proactive investigation: leaks detection
  - List of emails and passwords
  - Leaked database
  - AWS Keys
  - Credit-cards
  - PGP private keys
  - Certificate private keys
- Feed Passive DNS or any passive collection system
- CVE and PoC of vulnerabilities most used by attackers

# Support CERT and Law Enforcement activities

- Website monitoring
  - monitor booters
  - Detect encoded exploits (WebShell, malware encoded in Base64, ...)
  - SQL injections
- Chat/Channel monitoring
  - Monitor Threat Actor Chat and Community Activities
- Automatic and manual submission to threat sharing and incident response platforms
  - MISP
  - *flowIntel (Coming soon)*
- Term/Regex/YARA monitoring for local companies/government
- YARA Retro Hunt

## Sources of leaks

---

# Mistakes from users:

The screenshot shows a GitHub search interface with the query "remove\_password". The results page displays 322,302 commit results. The commits are listed in a grid format, each with a user icon, commit message, author, date, and a copy/paste button.

Commit Message	Author	Date	Commit Hash
Make remove_password actually work	javitonino	committed to freakiful/cartodb on 1 Mar	def411c
remove password	wenlei	committed to cjh1990/wap_demo 2 days ago	e9611e0
remove password	yejune	committed to yejune/dockerfile-sshd 3 days ago	037b956
Removed Passwords	Graham Beechum	committed to gbeeckum/teethyme 2 days ago	42279ee

Sort: Best match ▾

TLP:CLEAR

16/95

# Sources of Leaks: Paste Monitoring

- Example: <https://gist.github.com/>
  - Easily stores and shares text online
  - Used by programmers and legitimate users
    - Source code & configuration information
- Abused by attackers to store:
  - Lists of vulnerable/compromised sites
  - Software vulnerabilities (e.g., exploits)
  - Database dumps
    - User data
    - Credentials
    - Credit card details
  - Increasingly more sensitive information

## Examples of pastes (items)

The image shows a screenshot of a web-based file manager or paste service. It displays two separate windows, each containing a text file.

**Left Window:**

```
text 4.41 KB
1. - - - - - Tool by Y3t1y3t ( u
2.
3.
4.     text 4.57 KB
5.
6.
7.     1. #include "wejwyj.h"
8.
9.
10.    2.
11.
12.    3. int zapisz (FILE *plik_
13.        4.     int i, j;
14.
15.        5. if (obr->KOLOR==0) {
16.
17.            6.
18.            7. fprintf (plik_wy, "P2
19.            8. fprintf (plik_wy, "%d
20.            9. fprintf (plik_wy, "%d
21.            10. for (i=0; i<obr->wymy
22.                11.     for (j=0; j<obr->wymx; j++
23.                    12.         fprintf (plik_wy, "%d ,
24.            13.     }
25.
```

**Right Window:**

```
text 2.02 KB
1. KillerGram - Yuffie - Smoke The Big Dick [smkwhr] (Upload)
2.
3.
4.     text 2.66 KB
5.
6.
7.     1. <item name="%the_component_to_be_disabled%" xsi:type="array">
8.
9.         2.     <item name="config" xsi:type="array">
10.
11.            3.         <item name="componentDisabled" xsi:type="boolean">true</item>
12.
13.             4.     </item>
14.
15.             5. </item>
16.
17.             6.
18.             7. <?xml version="1.0"?>
19.
20.             8.
21.             9. <page xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xsi:noNamespace
22.                 /etc/page_configuration.xsd">
23.
24.                 10. <body>
25.
26.                     11.             <referenceBlock name="checkout.root">
27.
28.                         12.                 <arguments>
29.
30.                             13.                                 <argument name="jsLayout" xsi:type="array">
```

# Why So Many Leaks?

- **Economic Interests:**
  - Adversaries promoting services
- **Ransom Model:**
  - To publicly pressure the victims
- **Political Motives:**
  - Adversaries showing off
- **Collaboration:**
  - Criminals need to collaborate
- **Operational Infrastructure:**
  - Malware exfiltrating information on a paste website
- **Mistakes and Errors from Users**

## What's your role while discovering such leak?

As a CSIRT, we must address this issue.

- **Contacting companies or organizations** responsible for accidental leaks
- **Engaging with the media** about specific leak cases to ensure practical and factual reporting
- **Evaluating the economic landscape** for cyber criminals (e.g., DDoS booters<sup>5</sup> or the resale of personal information - comparing reality versus media coverage)
- **Analyzing collateral effects** of malware, software vulnerabilities, or data exfiltration

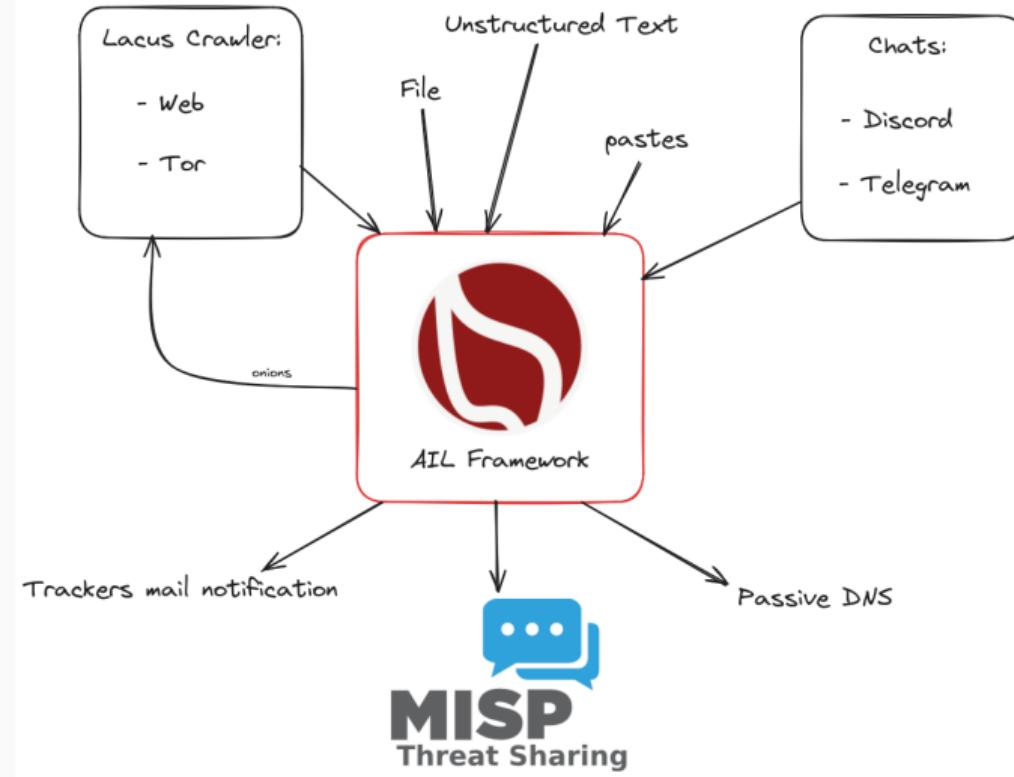
---

<sup>5</sup><https://github.com/D4-project/>

## Current capabilities

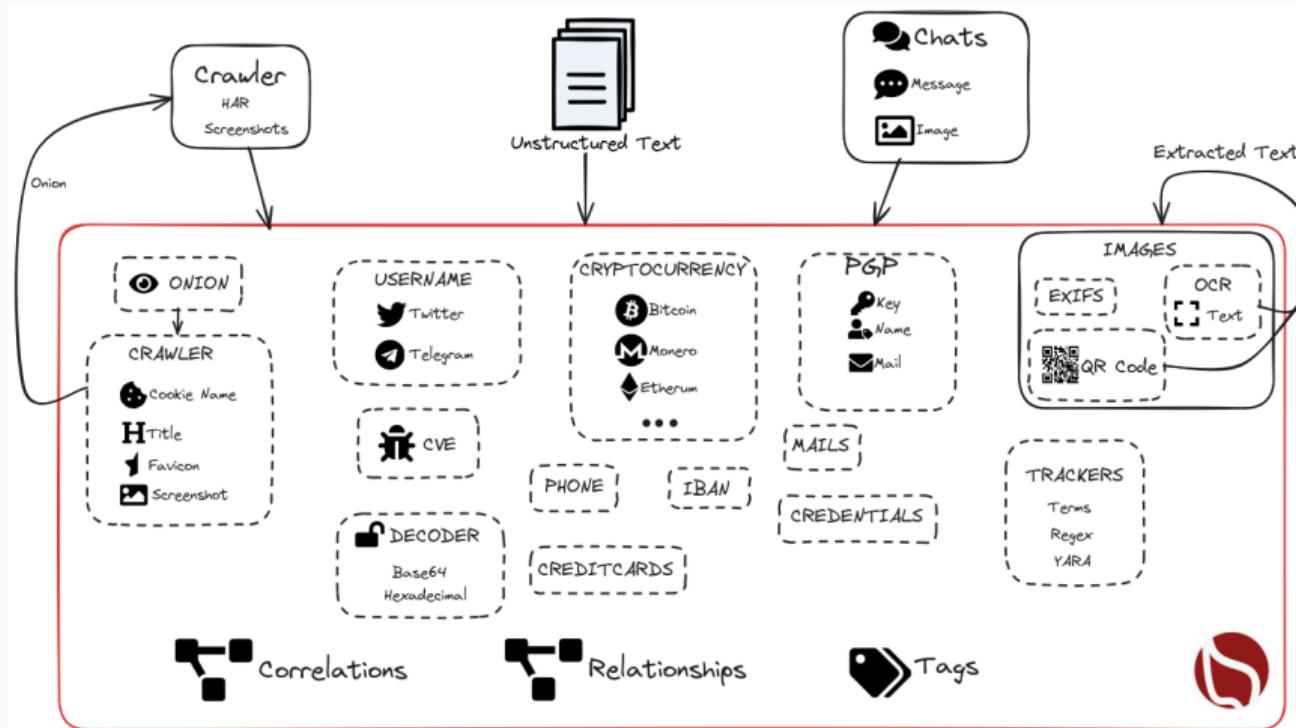
---

# High Level Overview



- 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 inputs**
- Tor Crawler (handle cookies authentication)
- Feeders: Discord, Telegram, ...

# Analysis of unstructured information



- Extracting **credit card numbers, credentials, phone numbers, ...**
- Extracting and validating potential **hostnames**
- Keeps track of **duplicates**
- Submission to threat sharing and incident response platforms (**MISP**)
- **Full-text indexer** to index unstructured information
- **Tagging** for classification and searches
- Terms, sets, regex and YARA **tracking, occurrences, and history**
- Archives, files, and raw **submission** from the UI
- Correlation engine based on PGP ID, cryptocurrencies, decoded (Base64, ...), usernames, cookie names, and many selectors to find relationships
- And many more

- Search and monitor specific keywords/patterns
  - Automatic Tagging
  - Email Notifications
- Track Word
  - ddos
- Track Set
  - booter,ddos,stresser;2
- Track Regex
  - circl\.lu
- YARA rules
  - <https://github.com/ail-project/ail-yara-rules>

# YARA Tracker

Certificate

Type: yara

Tracked: all-yara-rules/rules/crypto/certificate.yar

Date: 2023/05/12

Level: Global

Creator: admin@admin.test

First Seen: 2023 / 05 / 12

Last Seen: 2023 / 05 / 31

Tags

Mails

Webhook

Filters: No filters

Objects Match: decoded 6 Item 88

Edit Tracker

Yara Rule:

```
rule certificates
{
    meta:
        author = "@KevTheHermit"
        info = "Part of PasteHunter"
        reference = "https://github.com/kevthehermit/PasteHunter"

    strings:
        $ssh_priv = "BEGIN RSA PRIVATE KEY" wide ascii nocase
        $openssh_priv = "BEGIN OPENSSH PRIVATE KEY" wide ascii nocase
        $dsa_priv = "BEGIN DSA PRIVATE KEY" wide ascii nocase
        $sec_priv = "BEGIN EC PRIVATE KEY" wide ascii nocase
        $pgp_priv = "BEGIN PGP PRIVATE KEY" wide ascii nocase
        $pem_cert = "BEGIN CERTIFICATE" wide ascii nocase
        $pkcs7 = "BEGIN PKCS7"

    condition:
        any of them
}
```

2023-05-12 | 2023-05-31

Tracked Objects

TLP:CLEAR

all-yara-rules/rules/c

# Trackers - Practical part

- Create and test your own tracker

Create a new Tracker

E-Mails Notification (optional, space separated)

Show tracker to all Users

Webhook URL

Tracker Description (optional)

Objects to Track:

Decoded

Item

Filter Item by sources

New Decodes to track (ALL IF EMPTY)

PGP

Filter PGP by subtype:

name

mail

Tags

Custom Tags (optional, space separated)

Select Tags

Taxonomic Selected

Select Tags

Galaxy Selected

Tracker Type:

**TLP:CLEAR**

# Retro Hunt

test completed

Date 2023/05/10

Description None

Tags

Creator admin@admin.test

Filters {  
 "item": {  
 "date\_from": "20230304",  
 "date\_to": "20230601"  
 }  
}

Objects Match item 3

Show Objects

```
rule certificates
{
    meta:
        author = "@KevTheHermit"
        info = "Part of PasteHunter"
        reference = "https://github.com/kevthehermit/PasteHunter"

    strings:
        $ssh_priv = "BEGIN RSA PRIVATE KEY" wide ascii nocase
        $openssh_priv = "BEGIN OPENSSH PRIVATE KEY" wide ascii nocase
        $dsa_priv = "BEGIN DSA PRIVATE KEY" wide ascii nocase
        $ec_priv = "BEGIN EC PRIVATE KEY" wide ascii nocase
        $pgp_priv = "BEGIN PGP PRIVATE KEY" wide ascii nocase
        $pem_cert = "BEGIN CERTIFICATE" wide ascii nocase
        $pkcs7 = "BEGIN PKCS7"

    condition:
        any of them
}
```

Type	Id	Tags
●	archive/gist.github.com/2023/04/14/luzmiranda_3b3d1133a3d3842092c5fc5fb39e84f2.gz	infoleak:automatic-detection="private-key" test12 test12 infoleak:automatic-detection="certificate"
●	submitted/2023/04/20/submitted_cc9190ab-80d2-4d2b-9c0e-97c51e69a855.gz	infoleak:submission="manual" test12 infoleak:automatic-detection="rsa-private-key" infoleak:automatic-detection="vpn-static-key" test12 infoleak:automatic-detection="certificate" infoleak:automatic-detection="onion"
●	archive/gist.github.com/2023/04/13/chipzoller_d8d6d2d737d02ad4fe9d30a897170761.gz	test12 test12 infoleak:automatic-detection="certificate"

Showing 1 to 3 of 3 entries

Previous 1 Next

TLP:CLEAR

28/95

# Recon and intelligence gathering tools

- Attacker also share informations
- Recon tools detected: 94
  - sqlmap
  - dnscan
  - whois
  - msfconsole (metasploit)
  - dnmap
  - nmap
  - ...

# Recon and intelligence gathering tools

```
#####
=====
Hostname      www.pabloquintanilla.cl           ISP      Wix.com Ltd.
Continent     North America          Flag
US
Country       United States        Country Code   US
Region        Unknown            Local time    19 Nov 2019 07:59 CST
City          Unknown            Postal Code   Unknown
IP Address    185.230.60.195      Latitude     37.751
                  Longitude    -97.822
=====
```

```
#####
> www.pabloquintanilla.cl
Server:      38.132.106.139
Address:     38.132.106.139#53
```

Non-authoritative answer:

```
www.pabloquintanilla.cl canonical name = www192.wixdns.net.
www192.wixdns.net      canonical name = balancer.wixdns.net.
Name:      balancer.wixdns.net
Address:   185.230.60.211
>
```

```
#####
Domain name: pabloquintanilla.cl
```

```
Registrant name: SERGIO TORO
```

```
Registrant organisation:
```

```
Registrar name: NIC Chile
```

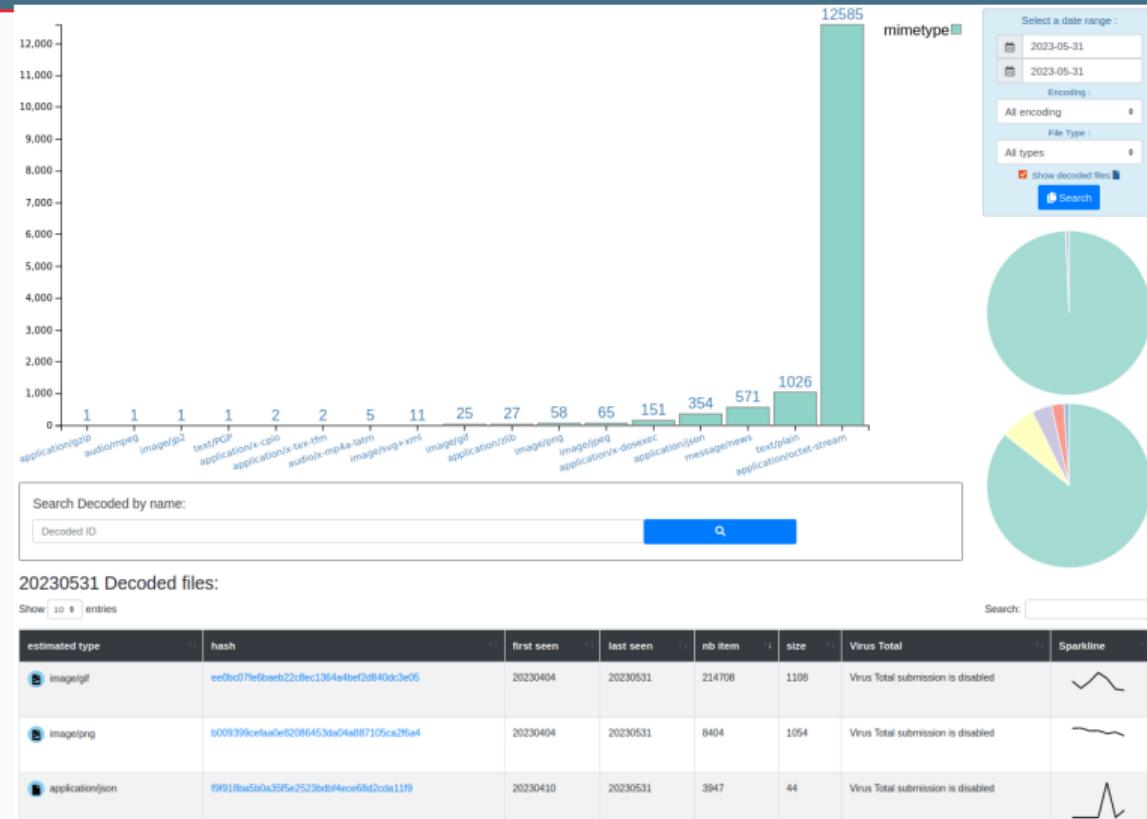
TLP:CLEAR

```
Registrar URL: https://www.nic.cl
```

```
Creation date: 2018-11-21 14:34:34 CLST
```

- Search for encoded strings
  - Base64
  - Hexadecimal
  - Binary
- Guess Mime-type
- Items/Domains Correlation

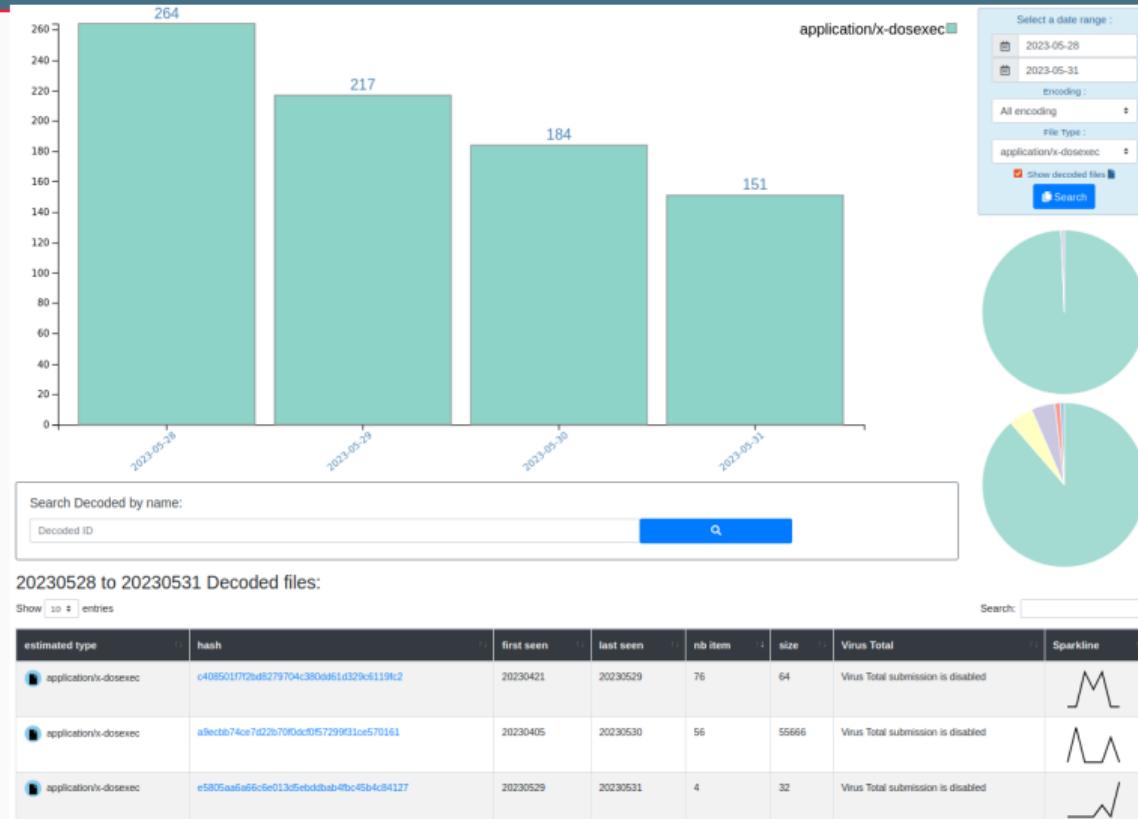
# Decoder:



TLP:CLEAR

32/95

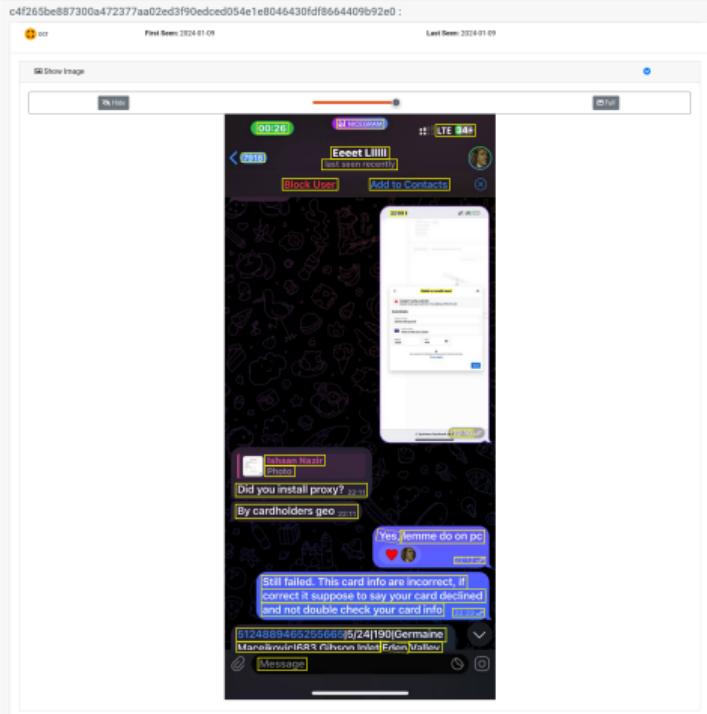
# Decoder:



TLP:CLEAR

33/95

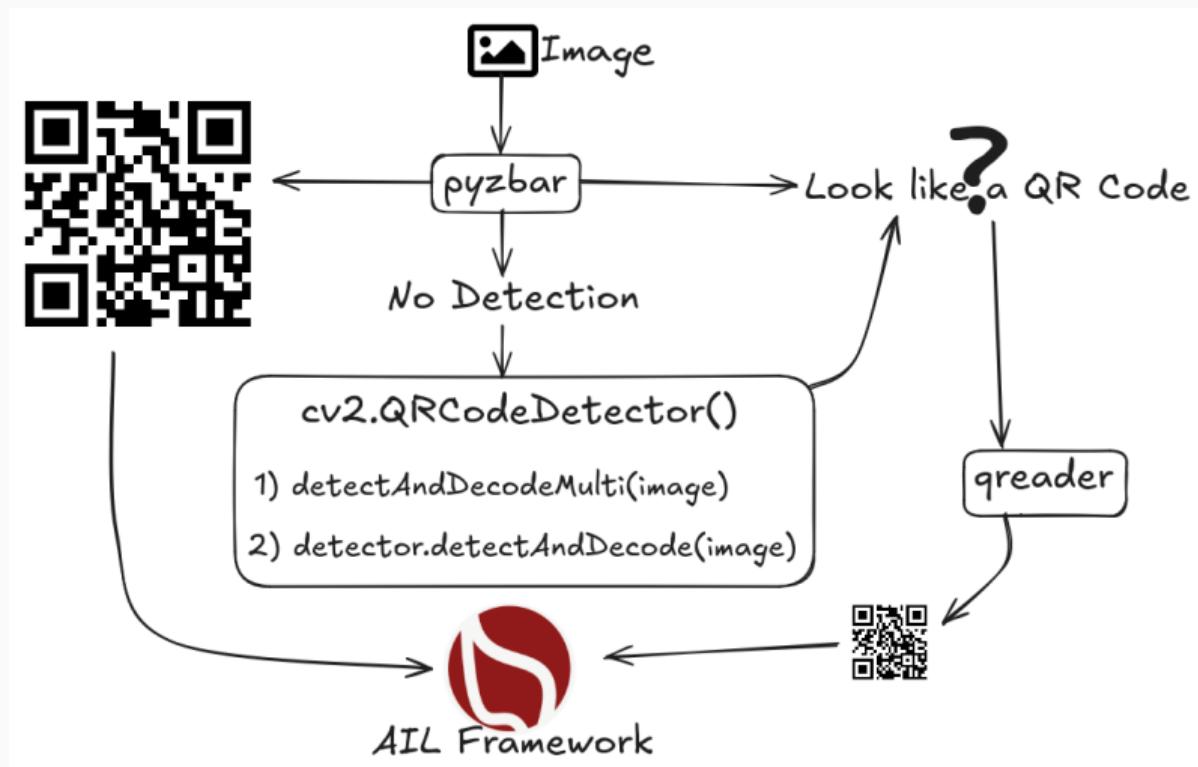
# OCR: Optical Character Recognition



- Threat actors are often verbose and frequently share extensive details in private channels.
- Many messages contain screenshots and images.
- Text detection and extraction are performed across 80+ languages using a CRNN (Convolutional Recurrent Neural Network).
- Enables keyword-based matching and detection.

TLP:CLEAR

# QR Code Extractor



# QR Codes



2022-12-16  
16:51:04

blue\_forever20



OCR

likeaboss

000201010211021643327300096494304155290090009649415312676076426760764000  
Mai62360520AP9J3I00000000000000000708447583166304F69D

Someone with union pay link or QR code

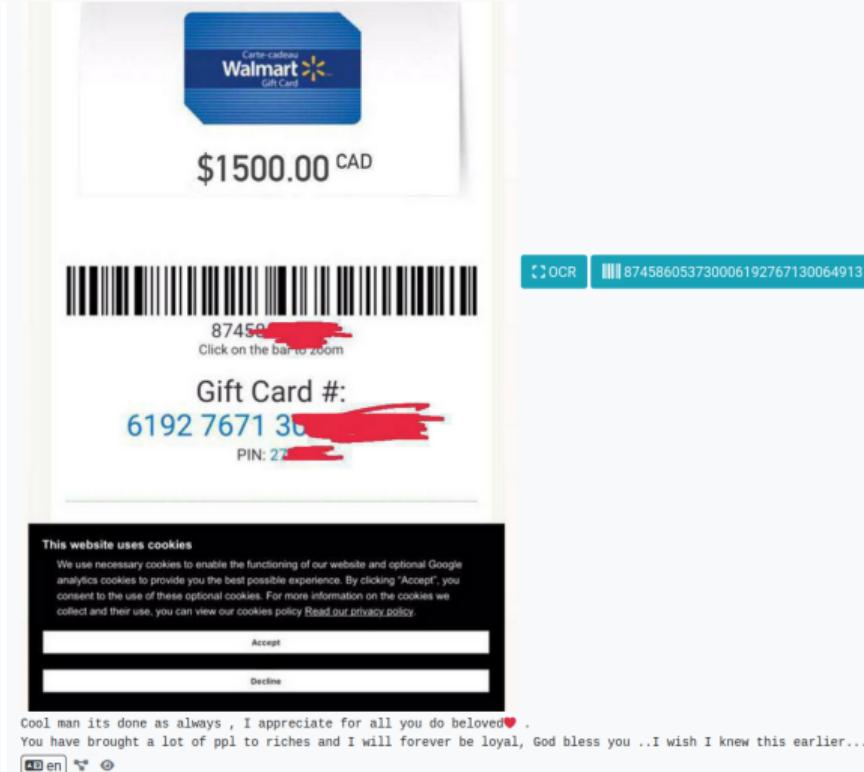
Hit me up

TLP:CLEAR

We can make money

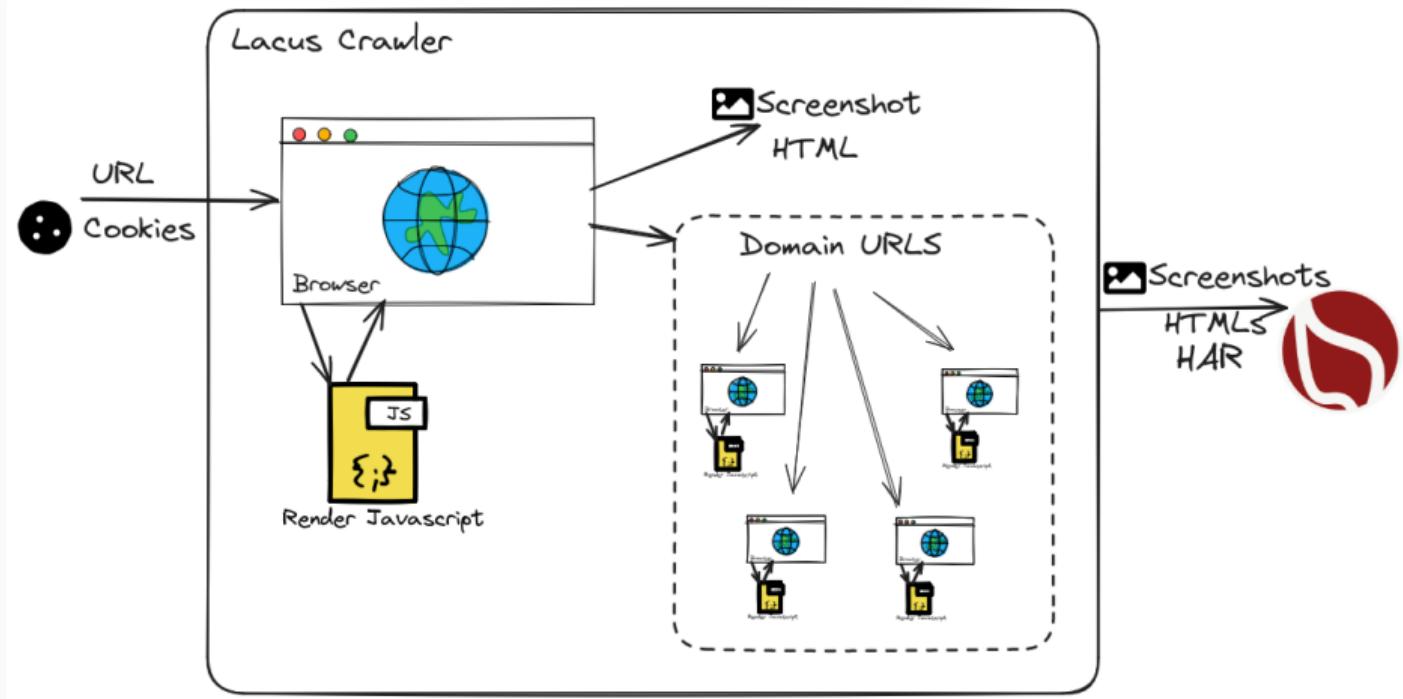
en

# Bar Codes



TLP:CLEAR

# Collection - Lacus Crawler



- Crawlers are used to navigate on regular website as well as .onion addresses (via automatic extraction of urls or manual submission)
- Lacus<sup>6</sup> ("scriptable" browser) is rendering the pages (including javascript) and produce screenshots (HAR archive too)
- How a domain is crawled by default:
  1. Fetch the first url
  2. Render javascript (webkit browser)
  3. Extract all urls
  4. Filter url: keep all url of this domain
  5. crawl next url (max depth = 1)

---

<sup>6</sup><https://github.com/ail-project/lacus>

- Lacus<sup>7</sup> is a capturing system using playwright, as a web service
- AIL utilizes Lacus for fetching and rendering domains.
  - Lacus can be installed and executed outside of AIL,
  - Enqueue what you want to capture,
  - Trigger the capture,
  - Get the capture result,

---

<sup>7</sup><https://github.com/ail-project/lacus>

# Crawler Settings - Lacus

## All Lacus Crawler

Lacus URL  Connected

[Edit](#)

## Crawlers

- TOR CRAWLER TEST OUTPUT: -

It works!

[ReRun Test](#)

Number of Concurrent Crawlers to Launch: **15**

[Edit](#)

# Crawler: Cookiejar

Use your cookies to login and bypass captcha

Edit Cookiejar

Description	Date	UUID	User
3thxemke2x7hcibu.onion	2020/03/31	90674deb-38fb-4eba-a661-18899ccb3841	admin@admin.test

[Edit Description](#) [Add Cookies](#)

[Edit](#) [Delete](#)

```
{ "domain": ".3thxemke2x7hcibu.onion", "name": "mybb[lastactive]", "path": "/forum/", "value": "1583829465" }
```

[Edit](#) [Delete](#)

```
{ "domain": ".3thxemke2x7hcibu.onion", "name": "loginattempts", "path": "/forum/", "value": "1" }
```

[Edit](#) [Delete](#)

```
{ "domain": ".3thxemke2x7hcibu.onion", "name": "sid", "path": "/forum/", "value": "047ab0cd97ff5bcc77edb6a" }
```

[Edit](#) [Delete](#)

```
{ "name": "remember_token", "value": "12|58cddd1511d74d341f23" }
```

[Edit](#) [Delete](#)

```
{ "domain": ".3thxemke2x7hcibu.onion", "name": "mybb[announcements]", "path": "/forum/", "value": "0" }
```

# Crawler: Cookiejar

3thxemke2x7hcibu.onion :

DOWN

First Seen	Last Check	Ports
2020/03/09	2020/03/30	[80]

infoleak:automatic-detection="onion" infoleak:automatic-detection="base64"

manual

Show Domain Correlations 139

Add to MISP Export

Decoded 1

Screenshot 138

Crawled Items Date: 2020/03/23 - 13:10:40 PORT: 80

Show 10 entries Search:

Crawled Dastes

Hide Full resolution

## Shere Khan

Welcome back, zuipori. You last visited: 03-20-2020, 01:35 PM Log Out

User CP View New Posts View Today's Posts Private Messages (Unread 2, Total 2)

You have 2 unread private messages. The most recent is from Jok3 titled KEY FOR PRIVATE SECTIONS

Shere Khan - Official Forum  
Private Messages

Menu User CP Home Messenger Compose Unread Sent Items Drafts Trash Can Tracking Edit Folders Your Profile Edit Profile Change Password Change Email Change Avatar Change Signature Edit Options Miscellaneous Group Memberships BuddyIgnore List Manage Attachments Send Drafts Subscribed Threads Forum Subscriptions View Profile

Inbox Enter Keywords Search PMs (Advanced Search) Message Title Sender Date/Time Sent [asc] KEY FOR PRIVATE SECTIONS Jok3 3 hours ago Verification Jok3 03-09-2020, 11:55 AM Move To Inbox or Delete the selected messages Jump to Folder: Inbox Get

Forum Team Contact Us Shere Khan - Hacking group Return to Top Lite (Archive) Mode Mark all forums read RSS Syndication Powered By MyBB, © 2002-2020 MyBB Group Current Time: 03-23-2020, 01:11 PM

<http://3thxemke2x7hcibu.onion/forum/private.php>

TLP:CLEAR

43/95

**Live demo!**

---

# Crawler: DDoS Booter

qy4n6ptiraa7mtfy73wcp6da2xrapmbanwfr5kei4zrq2va4uscvogid.onion :

First Seen	Last Check	Ports
2019/08/15	2019/10/06	[80]

[infoleak:automatic-detection="bitcoin-address"](#) [infoleak:automatic-detection="ethereum-address"](#)  
[infoleak:automatic-detection="onion"](#) [infoleak:automatic-detection="credit-card"](#) [ddos](#)

⊕

Last Origin: [crawled/2019/10/05/mqbyaj4ladgz5cd.onion](http://crawled/2019/10/05/mqbyaj4ladgz5cd.onion) [0aa31681-fa45-4fc3-8151-7a7c5ac7e906](#)

Show Domain Correlations 2

CRYPTOCURRENCIES 2

Hide Full resolution

HOME ABOUT PROOF PRICE PAYMENT

DDOSTECH

WICKR: DDOS TECHNOLOGY



Reviews

April 23, 2019

I turned to this service on the recommendation of my friend, ordered an attack for a whole week, the work was done with high quality and responsibility.

September 21, 2019

I found this site through YAHOO, immediately contacted this service, and I had a free attack for almost ten minutes.

We accept:

Accept payments cryptocurrency. Cryptocurrency transfers guarantee your our security transaction. We accept BTC, ETH, DASH, LTC, ETC, XMP ...

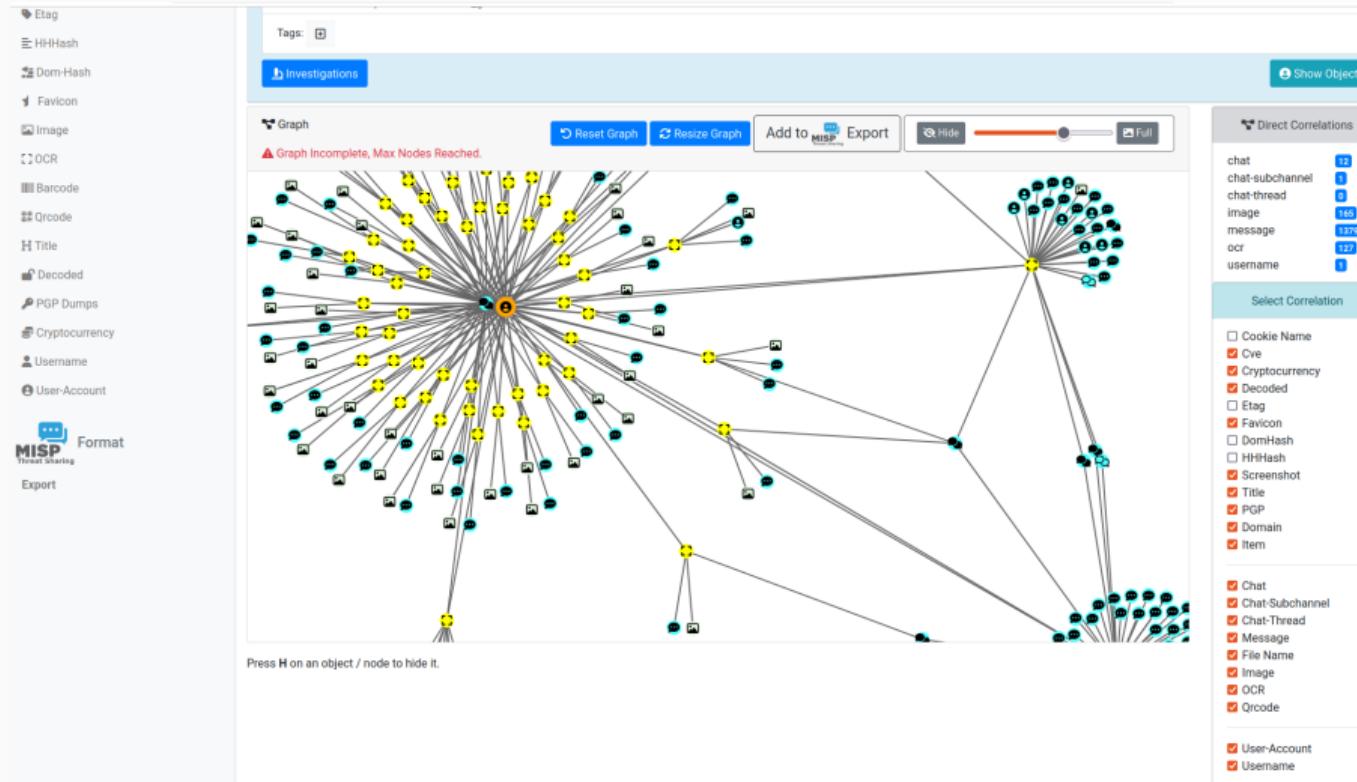


Wallets Addresses

TLP:CLEAR

44/95

# Correlations and relationship



TLP:CLEAR

45/95

# Investigations

### Tor Coin Mixer

UUID	9189d0e7c04c47a29f85666e9507e0a5	<a href="#">Delete</a>	<a href="#">Edit</a>	<a href="#">Export as Event</a>
Creator	admin@admin.test			
Tags	dark-web-topic="mixer"			
Date	2023-05-31			
Threat Level	medium			
Analysis	initial			
Info	Tor Coin Mixer			
# Objects	6			
Timestamp	2023-05-31 12:50:45			
Last change	2023-05-31 12:54:20			

### Objects

Show 10 entries Search:

Type	Id	Tags	
onion	jambler1y2zp8knhjbnj3mhfdajmyddqbxuf1voa32h5w4o6ux3cqrd.onion	[info:automatic-detections="index"] [info:automatic-detections="pgp-public-key-block"]	<a href="#">Delete</a>
onion	bitmonhf4cpncluhwffuskk23tvowswbef4lthree74oxjmz2yyqqd.onion	[info:automatic-detections="index"]	<a href="#">Delete</a>
key	0x0B280956F0E7CAF		<a href="#">Delete</a>
mail	support@jambler.io		<a href="#">Delete</a>
telegram	jambler		<a href="#">Delete</a>
name	Jambler.io		<a href="#">Delete</a>

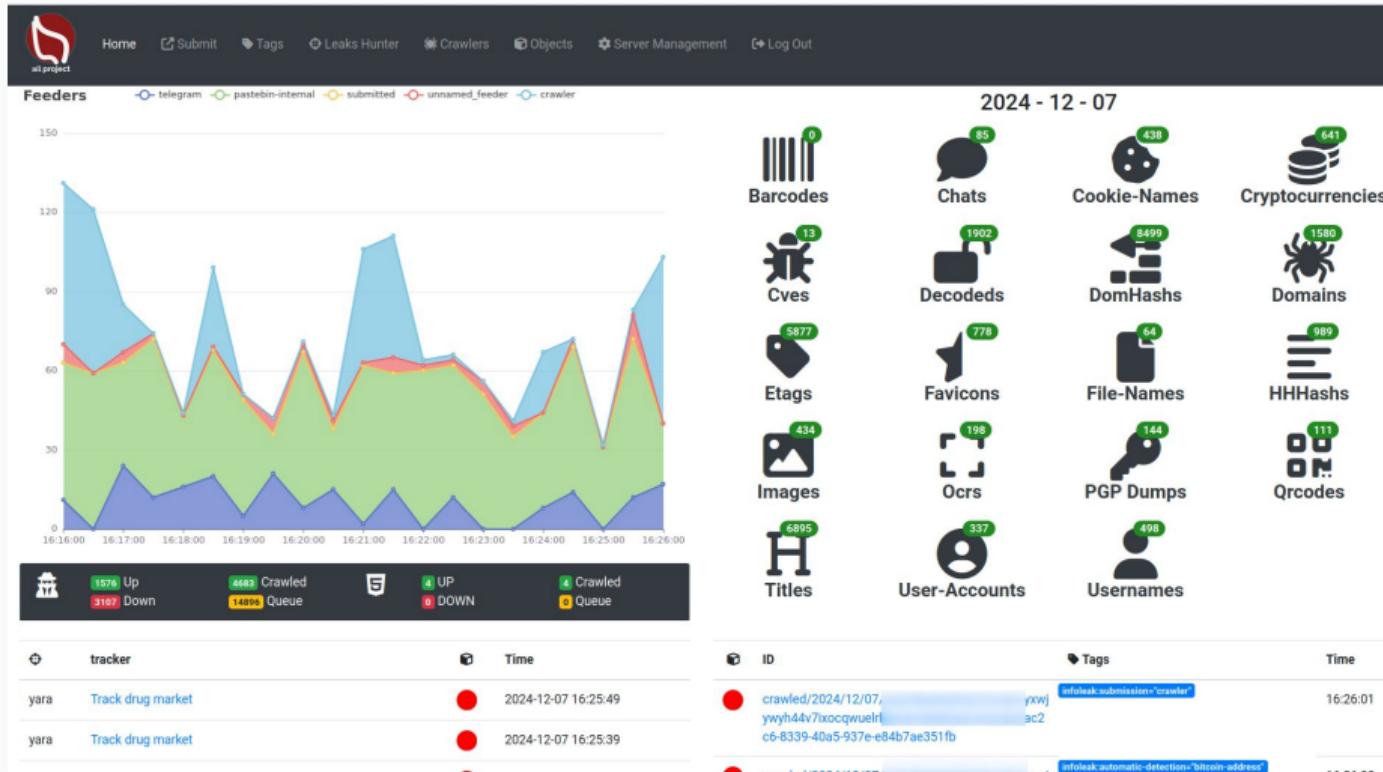
Showing 1 to 6 of 6 entries

TLP:CLEAR

Previous [1](#) Next

46/95

# Dashboard



# Example: Items Extracted

Date	Source	Size (Kb)	Number of lines	Max line length
2025/02/01	telegram	180.34	6033	27975

infoleak:automatic-detection="base64" infoleak:automatic-detection="phone-number" infoleak:automatic-detection="credit-card" infoleak:automatic-detection="credential" infoleak:automatic-detection="iban"

Add to Export Investigations

Extracted 3

Show 10 entries Search:

Type	ID	Extracted
tag	infoleak:automatic-detection="phone-number"	+97695152402 +972535242150 +919247146104 +212 608-950286 +212 629-117431 +212630864857 +84708794730 +96894427669 +44 7934 122985 +971501997173 +971-501027162 +971569883905 +971-585856200
tag	infoleak:automatic-detection="iban"	AE980260001015865178901 AE980260001015865178901
tag	infoleak:automatic-detection="credit-card"	373778604875195 4439137249559009

Showing 1 to 3 of 3 entries

**TLP:CLEAR**

Previous Next

48 / 95

# Example: Items Domain

Crawler

Last Origin:

- crawled/2025/02/03/drugaxeauuw4xt6eoqwztfkpa2ul5453q4ao4kqrtdhfoy6jmu5ryd.onionae52293f-f1da-4bd9-b83
- drugsxeauuw4xt6eoqwztfkpa2ul5453q4ao4kqrtdhfoy6jmu5ryd.onion

url <http://drugsxeauuw4xt6eoqwztfkpa2ul5453q4ao4kqrtdhfoy6jmu5ryd.onion/?product=chip-skimmer>

The screenshot shows a web browser window with the title "Anonymous Marketplace". The URL in the address bar is "http://drugsxeauuw4xt6eoqwztfkpa2ul5453q4ao4kqrtdhfoy6jmu5ryd.onion/?product=chip-skimmer". The page displays a product listing for "CHIP MSR CARD SKIMMER MCR 200 EMV MAG STRIPE READER WRITER". The price is \$149.00, and there are 28 customer reviews. The product image shows various electronic components, including a small grey box, several cables (USB and power), and some cards. Below the image, there are tabs for "DESCRIPTION" and "REVIEWS (28)".

## Description

CHIP MSR CARD SKIMMER MCR 200 EMV MAG STRIPE READER WRITER SOFTWARE MSR206

Original New Model of MCR-200 IC Chip Card Reader Writer Magnetic Stripe Card Reader Writer & EMV chip Encoder Track 1,2,3  
Comes with Reader/Writer Software v8.6 for Windows 95/98/Ms/2000/XP/Vista/Windows 7/Windows 8/Windows 8.1/Windows 10/MacOS/Linux (Only Ubuntu and Debian)

1. Magnetic Card Reader/Writer ICC206
2. Series designed to offer a card reading/writing
3. Solution for ISO 20121-6 formats

TLP:CLEAR

- Support PBOC2.0, EMV IC card
- Software provided for only magnetic card Reader or Writer. For EMV IC Card (chip card) Reader or Writer, customization needed. Needs to be reprogrammed using SDK to Read

49/95

# Example: 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:R0bluJ8XtB
```

```
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
```

```
Dom180:harkonnen
```

```
GaggedUK:a1k@chan
```

```
http://www.ariellaferrera.com/
```

TLP:CLEAR

# Example: Search by tags

Search Items by Tags :

2023-05-14      2023-05-27

[infoleak:automatic-detection="cve"](#) [infoleak:automatic-detection="bitcoin-address"](#)

[Search Items](#)

Show 10 entries      Search:

Date	Item	Action
2023/05/16	archive/gist.github.com/2023/05/16/Vazgen7788_c036ee7aad316d9038f2a3968abbcc5d.gz <a href="#">infoleak:automatic-detection="searchsploit-tool"</a> <a href="#">infoleak:automatic-detection="cve"</a> <a href="#">infoleak:automatic-detection="ethereum-address"</a> <a href="#">infoleak:automatic-detection="base64"</a> <a href="#">infoleak:automatic-detection="bitcoin-address"</a>	<a href="#">🔍</a>
2023/05/16	archive/gist.github.com/2023/05/16/vijay922_d35cd2f5c9abe682140379e35d5cd935.gz <a href="#">infoleak:automatic-detection="searchsploit-tool"</a> <a href="#">infoleak:automatic-detection="cve"</a> <a href="#">infoleak:automatic-detection="ethereum-address"</a> <a href="#">infoleak:automatic-detection="base64"</a> <a href="#">infoleak:automatic-detection="bitcoin-address"</a>	<a href="#">🔍</a>
2023/05/16	archive/gist.github.com/2023/05/16/DmitriyLewen_930515cde810283b7804950efafe3273.gz <a href="#">infoleak:automatic-detection="searchsploit-tool"</a> <a href="#">infoleak:automatic-detection="cve"</a> <a href="#">infoleak:automatic-detection="credential"</a> <a href="#">infoleak:automatic-detection="bitcoin-address"</a>	<a href="#">🔍</a>
2023/05/19	archive/gist.github.com/2023/05/19/GrahamcOfBorg_46422a069e8b942352a65f3121a769c5.gz <a href="#">infoleak:automatic-detection="cve"</a> <a href="#">infoleak:automatic-detection="credential"</a> <a href="#">infoleak:automatic-detection="bitcoin-address"</a>	<a href="#">🔍</a>
2023/05/26	archive/pastebin.com_pro/2023/05/26/5ewhAH10.gz <a href="#">infoleak:automatic-detection="ethereum-address"</a> <a href="#">infoleak:automatic-detection="cve"</a> <a href="#">infoleak:automatic-detection="bitcoin-address"</a>	<a href="#">🔍</a>

Showing 1 to 5 of 5 entries

Previous [1](#) Next

Items: 1-5 / 5

# MISP

---

- **Tagging** is a simple way to attach a classification to an event or attribute.
- **Classification must be globally used to be efficient.**
- Provide a set of already defined classifications modeling estimative language
- Taxonomies are implemented in a simple JSON format <sup>8</sup>.
- Can be easily cherry-picked or extended

---

<sup>8</sup><https://github.com/MISP/misp-taxonomies>

- **infoleak:** Information classified as being potential leak.
- **estimative-language:** Describe quality and credibility of underlying sources, data, and methodologies.
- **admiralty-scale:** Rank the reliability of a source and the credibility of an information
- **fpf<sup>9</sup>:** Evaluate the degree of identifiability of personal data and the types of pseudonymous data, de-identified data and anonymous data.

---

<sup>9</sup>Future of Privacy Forum

## Taxonomies useful in AIL

- **tor**: Describe Tor network infrastructure.
- **dark-web**: Criminal motivation on the dark web.
- **copine-scale<sup>10</sup>**: Categorise the severity of images of child sex abuse.

---

<sup>10</sup>Combating Paedophile Information Networks in Europe

## threat sharing and incident response platforms



**Goal:** submission to threat sharing and incident response platforms.

## threat sharing and incident response platforms



1. Use infoleak taxonomy<sup>11</sup>
2. Add your own tags
3. Export AIL objects to MISP core format
4. Download it or Create a MISP Event<sup>12</sup>

---

<sup>11</sup><https://www.misp-project.org/taxonomies.html>

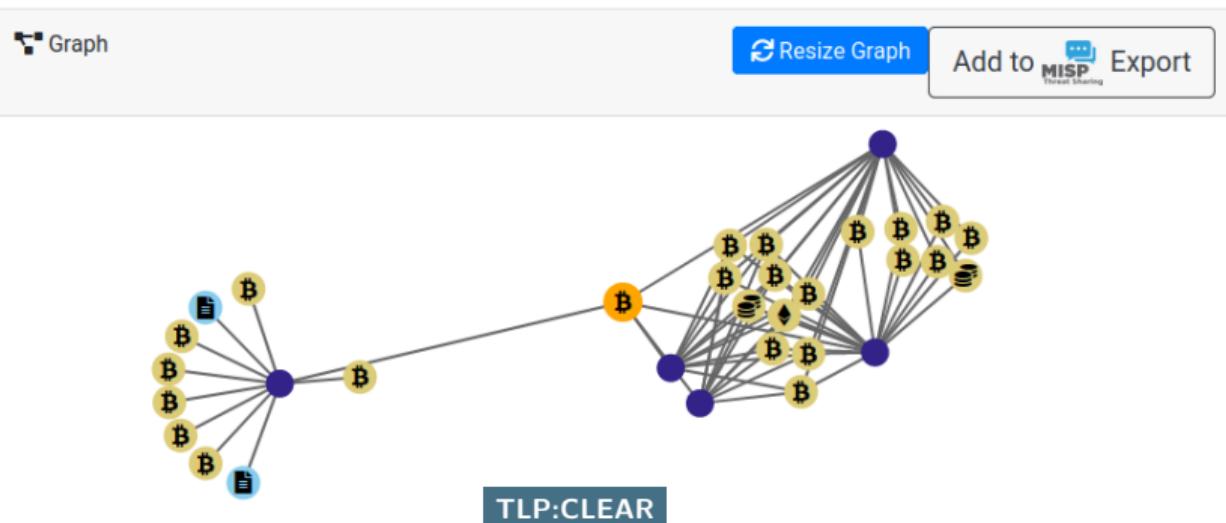
<sup>12</sup><https://www.misp-standard.org/rfc/misp-standard-core.txt>

# MISP Export

1Gt545E48EPsyTC8voKQDCFfpTkwiuXduw :

Object type	type	First seen	Last seen	Nb seen	
cryptocurrency	bitcoin	2020/01/17	2020/02/20	5	

Expand Bitcoin address



# MISP Export

nttfj36sp47cw2yecop572zjvjeazgazieunllouudplzqt2m  
5h465yd.onion :

First Seen	Last Check	Ports
2020/02/19	2020/02/19	['80]

infoleak:automatic-detection="onion"

[+]

Last Origin: [crawled/2020/02/19/dark.failc126d32a-3ed1-468f-ba24-f2e5956f4035](#)

Show Domain Correlations 4

Add to MISP Threat Sharing Export

Screenshot 4

TLP:CLEAR

Hide

 Empire Market

LOGIN REGISTER FORUMS VB

Login

LOGIN TO EMPIRE MARKET

Welcome to Empire Market! Please log in. Registrations are free and open to everyone.

Username

Password

What's the latest?

Login

Copyright © 2020 Empire Market

# MISP Export

 MISP Exporter

Select a list of objects to export

Object Type	Object ID	Lvl	
Object type...		0	
Object type...	1Gt545E48EPsyTC8voKQDCFfpTkwiuXduw	1	
Domain	nttfj36sp47cw2yecop57zjvjeazgazieunlouudplzqt2m5h465yd.onion	0	

JSON Export  Export to MISP Instance

Distribution:

Threat Level:

Analysis:

Event Info:

Publish Event



TLP:CLEAR

# Automatic MISP Export on tags

MISP Auto Event Creation Enabled



**MISP**  
Threat Sharing

× Disable Event Creation

The hive Auto Alert Creation Disabled



Enable Alert Creation

MISP Tags To Push : 3 / 89

Show 10 entries Search:

Enabled	Tag
<input checked="" type="checkbox"/>	infoleak:analyst-detection="aws-key"
<input checked="" type="checkbox"/>	infoleak:automatic-detection="credit-card"
<input checked="" type="checkbox"/>	test_custom
<input type="checkbox"/>	infoleak:analyst-detection="api-key"
<input type="checkbox"/>	infoleak:analyst-detection="base64"

The Hive Tags To Push : 4 / 89

Show 10 entries Search:

Enabled	Tag
<input type="checkbox"/>	infoleak:analyst-detection="api-key"
<input type="checkbox"/>	infoleak:analyst-detection="aws-key"
<input checked="" type="checkbox"/>	infoleak:analyst-detection="base64"
<input checked="" type="checkbox"/>	infoleak:analyst-detection="binary"
<input type="checkbox"/>	infoleak:analyst-detection="bitcoin-address"

TLP:CLEAR

60/95

# API

---

# ReST API Access

AIL exposes a ReST API which can be used to interact with the back-end<sup>13</sup>.

```
curl https://127.0.0.1:7000/api/v1/add/crawler/task  
--header "Authorization: iHc1_ChZxj1aXmiFiF1mkxxQkzawwriEaZpPqyTQj"  
-H "Content-Type: application/json"  
--data @input.json -X POST
```

---

<sup>13</sup><https://github.com/ail-project/ail-framework/blob/master/doc/README.md>

## Setting up the framework

---

## Installation Steps

```
git clone https://github.com/ail-project/ail-framework.git  
cd AIL-framework  
../installing_deps.sh
```

## Feeding the framework

---

# Feeding Data to AIL

There are different ways to feed data into AIL:

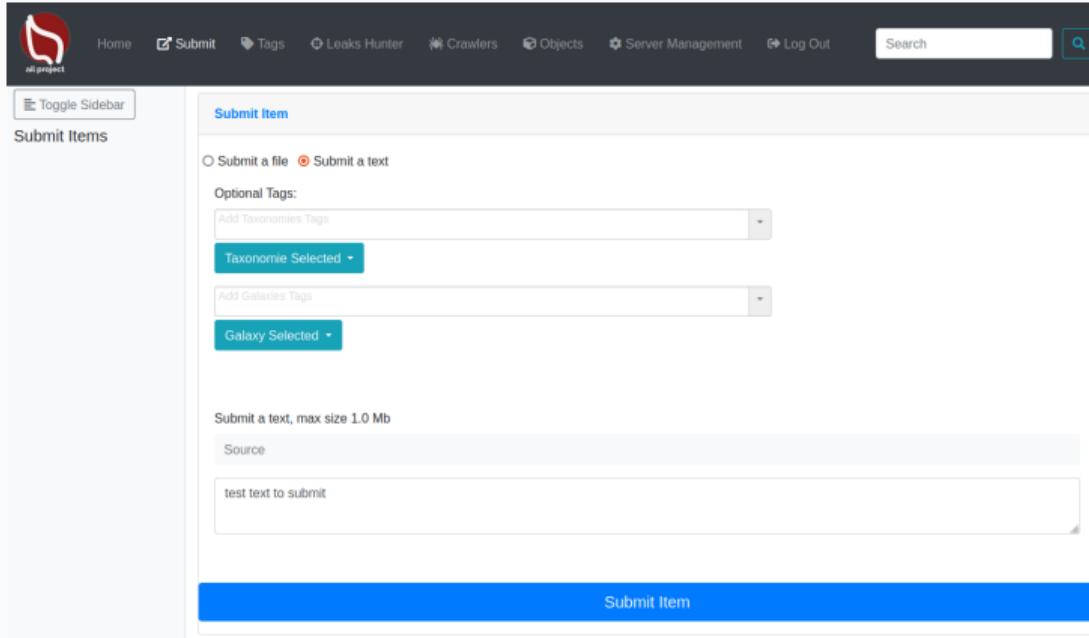
1. AIL Importers:
  - Dir / Files
  - ZMQ
  - *pystemon*
2. AIL Feeders (discord, telegram, ...)
3. Feed your own data using the API
4. Feed your own file/text using the UI (Submit section)



## /!\ Limitation:

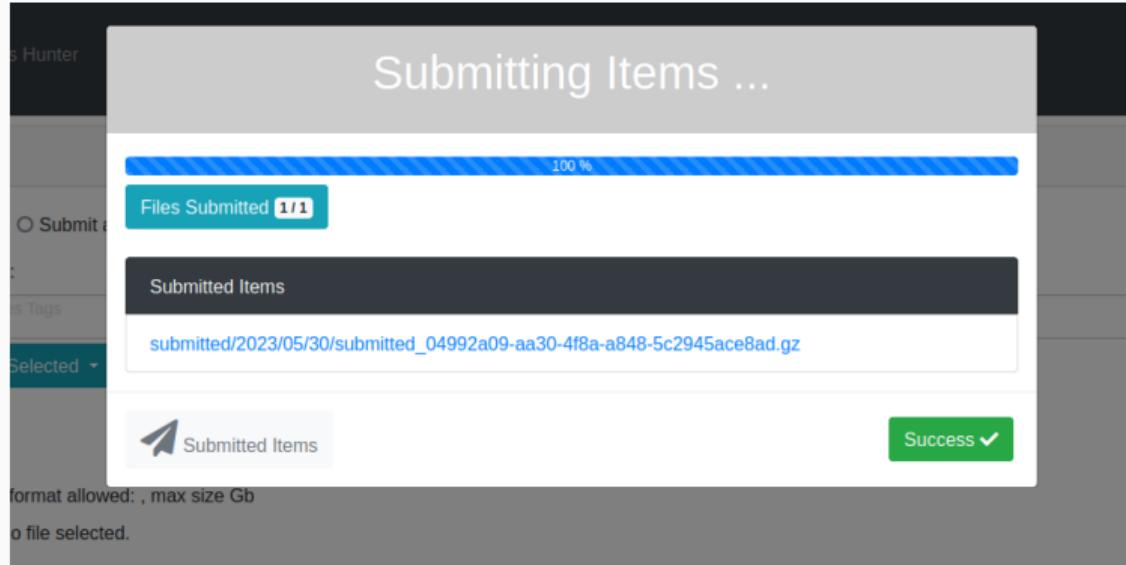
- Each file to be fed must be of a reasonable size:
  - ~ 3 Mb / file is already large
  - This is because some modules are doing regex matching (default timeout of 30 seconds)
  - If you want to feed a large file, better split it in multiple ones

# Via the UI (1)



The screenshot shows the 'Submit Item' page of the All Project UI. At the top, there is a navigation bar with links for Home, Submit, Tags, Leaks Hunter, Crawlers, Objects, Server Management, Log Out, and a search bar. On the left, there is a sidebar with a logo for 'all project' and buttons for 'Toggle Sidebar' and 'Submit Items'. The main content area has a title 'Submit Item' and two radio buttons: 'Submit a file' (unchecked) and 'Submit a text' (checked). Below these are sections for 'Optional Tags' with dropdown menus for 'Add Taxonomies Tags' and 'Taxonomy Selected', and 'Add Galaxies Tags' and 'Galaxy Selected'. There is also a note 'Submit a text, max size 1.0 Mb' and a 'Source' section containing the text 'test text to submit'. At the bottom is a large blue button labeled 'Submit Item'.

## Via the UI (2)



# API – Feeding AIL with Your Own Data

**Endpoint:** api/v1/import/item

## Example JSON Payload

```
{  
    "type": "text",  
    "tags": [  
        "infoleak:analyst-detection=\"private-key\""  
    ],  
    "text": "text to import"  
}
```

## AIL Feeders and Importers

- **More than 12 feeders** are available for AIL users to import data from external sources.
- External feeders can run anywhere and are completely independent of the AIL framework.
- Feeders can use their **own internal logic** and even generate custom JSON metadata.
- The resulting JSON is then pushed to the AIL API.

- `ail-feeder-gharchive`<sup>14</sup> is a generic tool to extract information from **GHArchive**, collect data, and feed AIL via the AIL ReST API.
- `ail-feeder-github-repo`<sup>15</sup> collects data from GitHub repositories and pushes it to AIL.
- This is a simple way to feed AIL with content from a set of **suspicious Git repositories** or to find leaks in existing or monitored repositories.

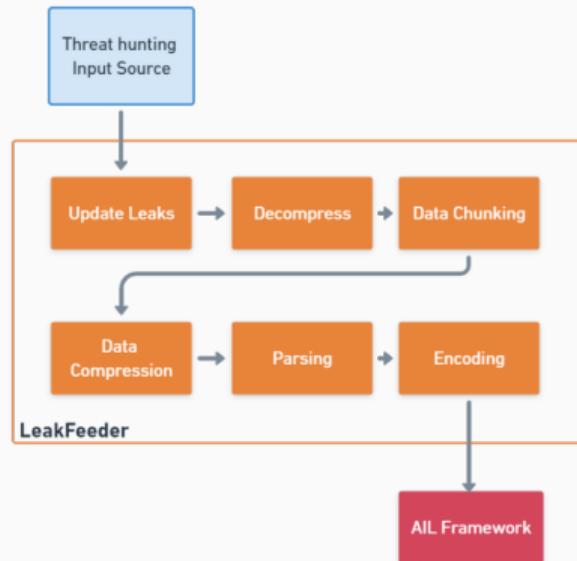
---

<sup>14</sup><https://github.com/ail-project/ail-feeder-gharchive>

<sup>15</sup><https://github.com/ail-project/ail-feeder-github-repo>

# AIL LeakFeeder

- ail-feeder-leak<sup>16</sup> automates the process of feeding large leaked files directly into AIL.



<sup>16</sup><https://github.com/ail-project/ail-feeder-leak>

- `ail-feeder-telegram`<sup>17</sup> is a **Telegram data feeder** for AIL.
- It requires a Telegram API ID and hash, which are linked to your Telegram phone number.

---

<sup>17</sup><https://github.com/ail-project/ail-feeder-telegram>

# Importers

---

- Importers are located in the `/bin/importer` directory.
- They are used to import various types of data into AIL.
- Adding new importers is straightforward and modular.
- Available importers include:
  - AIL Feeders
  - ZMQ
  - pystemon
  - File-based importers

# File Importer

- importer/FileImporter.py

## Import File:

### Import a Single File

```
..../AILENV/bin/activate  
cd tools/  
./file_dir_importer.py -f MY_FILE_PATH
```

## Import Directory:

### Import a Directory

```
..../AILENV/bin/activate  
cd tools/  
./file_dir_importer.py -d MY_DIR_PATH
```

TLP:CLEAR

## Feeding AIL with custom JSON



conti jabber leaks [anonfiles.com/VeP6K6K5xc/1\\_t...](https://anonfiles.com/VeP6K6K5xc/1_t...)

9:22 PM · 27 févr. 2022 · Twitter Web App

---

123 Retweets   23 Tweets cités   297 J'aime

```
{  
    "ts": "2020-09-08T00:28:49.471678",  
    "from": "ceram@q3mcco35auwcstmt.onion",  
    "to": "stern@q3mcco35auwcstmt.onion",  
    "body": "Проинструктируйте меня. Что делать?"  
}
```

## Feeding AIL with Conti leaks

---

- Conti jabber leaks are a good candidate for AIL analysis:
  - PGP keys
  - Bitcoin addresses, maybe others,
  - onion hidden services
- first we translated the files on english using deepl.com
- then we created a feeder to import json data in AIL
- Support added in AIL to correlate jabber usernames

## Feeding AIL with Conti Leaks – Python Feeder

### feeder.py

```
from pyail import PyAIL
#... imports
#... setup code
for content in sys.stdin:
    elm = json.loads(content)
    tmp = elm['body']
    tmpmt = {}
    tmpmt['jabber:to'] = elm['to']
    tmpmt['jabber:from'] = elm['from']
    tmpmt['jabber:ts'] = elm['ts']
    tmpmt['jabber:id'] = str(uuid.uuid4())
    pyail.feed_json_item(tmp, tmpmt, ailfeedertype, source_uuid)
```

## Feeding AIL with Conti Leaks – Running the Feeder

Run the feeder with Conti leaks

```
$ cat ~/conti/* | jq . -c | python ./feeder.py
```

# Feeding AIL with Conti leaks

- Use grep to limit the noise on an instance by only sending interesting bits:

## Filter for PGP keys

```
$ cat ~/conti/* | jq . -c | grep PGP | python ./feeder.py
```

## Filter for .onion URLs

```
$ cat ~/conti/* | jq . -c | grep "http://"
```

## Filter for Telegram links

```
$ cat ~/conti/* | jq . -c | grep "tg://"
```

## Filter for Bitcoin addresses

```
$ cat ~/conti/* | jq . -c | egrep --regexp="([13][a-km-zA-HJ-NP-Z1-9]{25,34})"
```

## Starting the framework

---

# Running Your Own Instance from Source

Accessing the environment and starting AII:

**Start AII from source:**

```
# Launch the system and the web interface  
cd bin/  
. ./LAUNCH -l
```

## Launch the updater:

```
cd bin/  
# git pull and launch all updates:  
. ./LAUNCH -u  
  
# PS:  
# The Updater is launched by default each time  
# you start the framework with  
# ./LAUNCH -l
```

## **AIL ecosystem - Challenges and design**

---

# AIL ecosystem: Technologies used

---

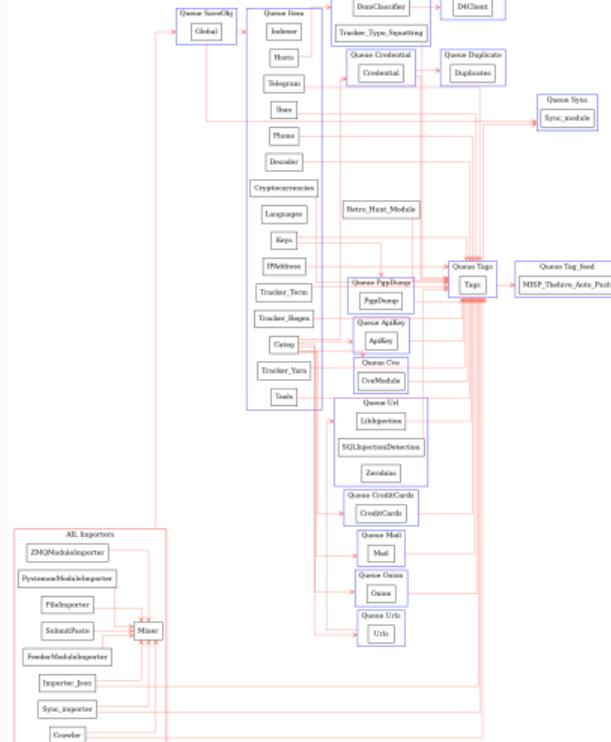
**Programming language:** Full python3

**Databases:** Redis and Kvrocks

**Server:** Flask

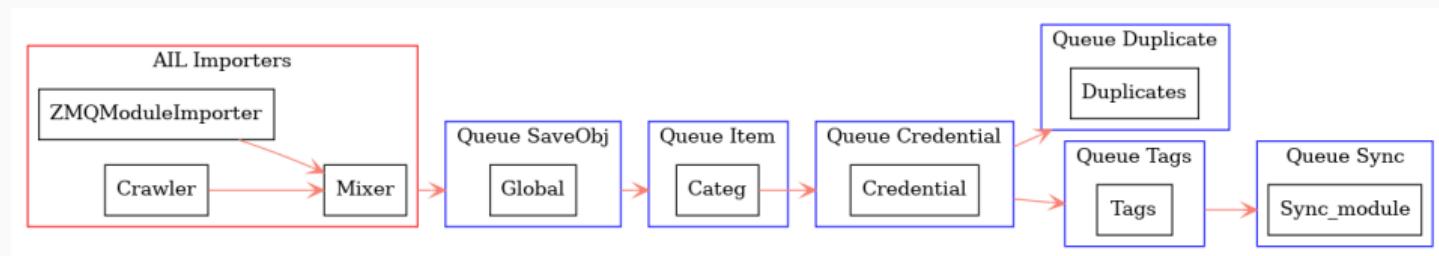
**Data message passing:** Redis Set

## AIL global architecture: Data streaming between module

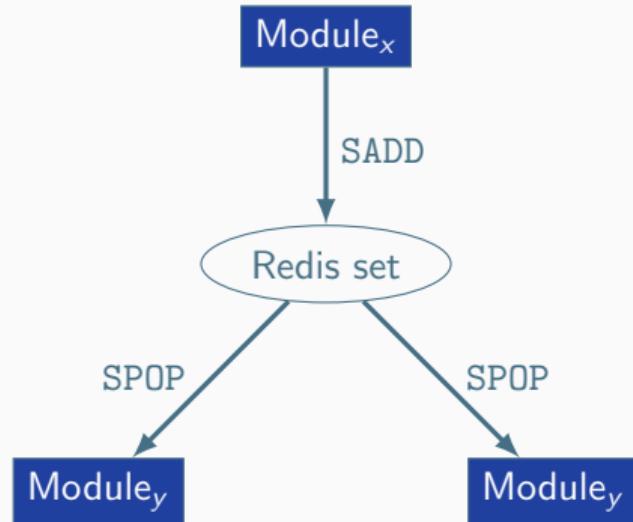


TLP:CLEAR

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



# Message consuming



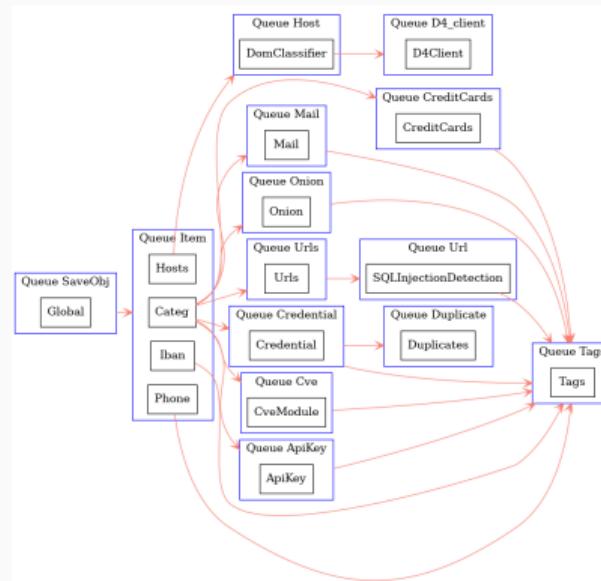
- No message lost nor double processing
- Multiprocessing!

## Creating new features

---

# Developing new features: Plug-in a module in the system

Choose where to put your module in the data flow:



Then, modify configs/modules.cfg accordingly

### Python Module Example:

```
from modules.abstract_module import AbstractModule

class NewModule(AbstractModule):

    def __init__(self):
        super().__init__()
        self.logger.info(f'Module {self.module_name} initialized')

    # Do something with the message from the queue
    def compute(self, message):
        # Process Message

# LAUNCH MODULE
if __name__ == '__main__':
    module = NewModule()
```

## Writing your own Importer - /bin/importer/

### Importer Example:

```
from importer.abstract_importer import AbstractImporter
from modules.abstract_module import AbstractModule

class MyNewImporter(AbstractImporter):

    def __init__(self):
        super().__init__()
        # super().__init__(queue=True)    # if it's a one-time run importer
        self.logger.info(f'Importer {self.name} initialized')

    def importer(self, my_var): # import function
        # Process my_var and get content to import
        content = GET_MY_CONTENT_TO_IMPORT
        # if content is not g TLP:CLEAR and/or not b64 encoded,
        # set gzipped and/or b64 to False
```

## Writing your own Importer - /bin/importer/

### Module Runner Example:

```
def get_message(self):
    return self.importer.importer()

def compute(self, message):
    self.add_message_to_queue(message)

if __name__ == '__main__':
    module = MyNewModuleImporter()
    module.run()

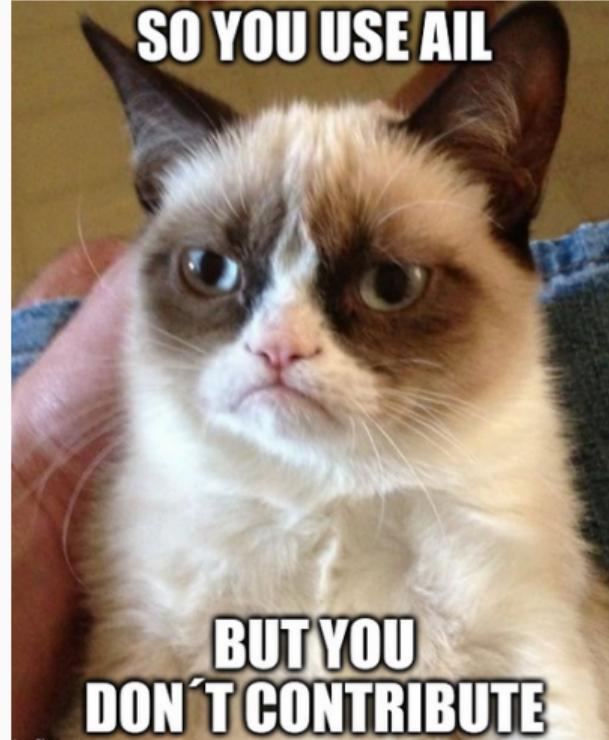
# if it's a one-time run:
# importer = MyImporter()
# importer.importer(my_var)
```

## Contribution rules

---

## How to contribute

---



## Glimpse of contributed features

---

- Docker
- Ansible
- Email alerting
- SQL injection detection
- Phone number detection

## 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!



- 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.

## Implementation Steps in AIL project

- **Gradual changes** in AIL to add required functionalities to support the objectives.
- **Time-memory trade-off** can be challenging to ensure a functional framework.
- Evaluation and integration of new modules in AIL based on time-memory comparisons.
- Semantic aspects are challenging due to the diverse data sources, unstructured data and languages seen.

# Ongoing Developments

- Text Geolocation
- Improved MISP Export
- Bloom filter filtering - PSS
- Analysis of relationships and activity between chats, including message forwards, replies, timelines, etc.
- Improved indexing relying on Solr, Lucene or other components
- Integration with FlowIntel

## Annexes

---

# Managing AIL: Old fashion way

## Access the script screen

```
screen -r Script
```

Table 1: 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